

Modelo de indicadores para el diseño de aplicaciones web, aplicado a los trabajos de grado de Ingeniería de Sistemas de la Universidad Popular del Cesar Seccional Aguachica.

KATERINE BELEÑO CASELLES

Ingeniera de Sistemas

Grupo de Investigación CICOM

Universidad de Pamplona

Trabajo para optar al título de Magister en Gestión De Proyectos Informáticos



Maestría en Gestión de Proyectos Informáticos
Facultad de Ingenierías y Arquitectura
Universidad de Pamplona
Pamplona, Diciembre de 2015

Modelo de indicadores para el diseño de aplicaciones web, aplicado a los trabajos de grado de Ingeniería de Sistemas de la Universidad Popular del Cesar Seccional Aguachica.

KATERINE BELEÑO CASELLES

Ingeniera de Sistemas
Universidad de Pamplona

Director

LUIS ALBERTO ESTEBAN VILLAMIZAR

Mg. En informática
Docente Facultad de Ingenierías y Arquitectura
Universidad de Pamplona

Maestría en Gestión de Proyectos Informáticos
Grupo de Investigación CICOM
Línea de Investigación en Ingeniería del software



UNIVERSIDAD DE PAMPLONA
PAMPLONA, Febrero 2016

AGRADECIMIENTOS

Gracias a Dios, por colocar en mí camino a las personas que se convirtieron en mi apoyo, compañía y orientación especialmente durante esta etapa de estudio, por iluminar mi mente para facilitar mi trabajo y mi formación profesional.

Gracias a mi madre, por su apoyo incondicional permanente, porque gracias a su esfuerzo desinteresado me convirtieron en una persona de servicio.

Gracias a mi esposo, por su apoyo constante, por hacer que el tiempo para dedicar a mis estudios fuese posible.

Gracias a los Docentes de la Universidad de Pamplona, quienes con sus conocimientos y orientaciones contribuyeron en mi formación profesional y alcanzar la meta deseada.

Gracias a mi director de trabajo de grado Luis Esteban Villamizar, quien con sus orientaciones permitió llevar a cabalidad el desarrollo del presente estudio, y que con su experiencia y aportes acertados permitieron generar un producto de calidad.

Gracias a todos mis compañeros de la Maestría en Gestión de Proyectos Informáticos por los momentos compartidos y hacer más agradable esta etapa de mi vida, con especial afecto a Darwin, JT, Yasser y Cesar,

A todos, eterna gratitud y sincero agradecimiento.

DEDICATORIA

*A mi madre: Ayda
A mi esposo: Luis Alfonso*

Resumen

Este documento presenta un modelo de métricas para los diseños de aplicaciones web desarrolladas como trabajo de grado en el programa de ingeniería de sistemas de la Universidad Popular del Cesar, Sede Aguchica. Dicho modelo surge de un estudio del estado de arte al respecto de métricas para diseño y está adaptado a las necesidades propias del contexto sobre el cual se desarrolló la investigación

De igual manera se presenta el proceso de validación del modelo propuesto mediante la valoración de algunos desarrollos de software web como trabajos de grado ya ejecutados.

Abstract

This paper presents a model metric designs for web applications developed as grade work in systems engineering program of the Popular University of Cesar, home Aguchica. This model comes from a study of the state of art about metrics to design and is adapted to the needs of the context in which the research was conducted

Similarly, the process of validating the model proposed by assessing some developments in web software and grade papers presented and executed.

Tabla de contenido

1	Introducción.....	1
1.1	Planteamiento del Problema y Justificación	2
1.2	Objetivo General.....	4
1.3	Objetivos Específicos	5
1.4	Acotaciones.....	5
1.5	Metodología de Investigación.....	5
1.5.1	Enfoque Cualitativo	5
1.5.2	Alcance descriptivo	6
1.5.3	Diseño del proceso de investigación	6
1.5.4	Métodos para la consecución del primer objetivo	7
1.5.5	Métodos para la consecución segundo objetivo	7
1.5.6	Métodos para la consecución tercer objetivo	8
1.5.7	Fuentes de Datos Consultadas	8
1.5.8	Instrumentos de captura de información	8
1.5.8.1	La técnica de revisión documental.....	8
1.5.8.2	La técnica de conversación asociada a la entrevista.	9
2	Marco Teórico	9
2.1	Calidad del software	10
2.1.1	Medición.....	11
2.1.2	Métrica.....	11
2.1.3	Unidad de medición.....	13
2.1.4	Indicador.....	13
2.2	Modelos y estándares de calidad del producto software.....	13
2.2.1	Modelo de McCall.....	15
2.2.2	Modelo Boehm (1978)	19
2.2.3	ModeloGild (1988).....	20
2.2.4	Modelo GQM (Goal – Question – Metric):.....	21
2.2.5	Modelo FURPS	21
2.2.6	Modelo DROMEY	23

2.2.7	Modelo SATC (Software Assurance Technology Center)	24
2.2.8	Metodología SQA	27
2.2.9	ISO 9126.....	28
2.2.9.1	Modelo de calidad interna y externa	30
2.2.9.2	Modelo de calidad en uso.....	31
2.2.10	ISO/IEC 25000: SQUARE	31
2.3	Diseño	34
2.4	Aplicaciones web	35
2.4.1	Calidad en aplicaciones web:	36
2.4.2	Desarrollo basado en componentes	37
2.4.3	Seguridad	38
2.4.4	Diseño de aplicaciones web.....	38
2.4.4.1	Metodologías de diseño usadas en Ingeniería Web	39
2.5	Modelos de calidad para las aplicaciones web	42
2.5.1	W3C : (1999).....	42
2.5.2	Nielsen (2002):	42
2.5.3	IvoryMelody (2001)	43
2.5.4	SUE: (1997).....	44
2.5.5	WebQeM: (Olsina 2001)	46
2.5.6	Zuh Y Gauch: (2000).....	46
2.5.7	Otros modelos para la web:	47
2.6	Estado del arte de Modelo de Indicadores para Aplicaciones Web en fase de Diseño.....	55
2.7	Análisis comparativo de los modelos de calidad del producto	57
2.8	Tipo de proyectos sobre el cual se realiza el aporte.....	60
3	Modelo de Indicadores para la fase de diseño de aplicaciones web.....	62
3.1	Artefactos a medir en la fase de diseño de aplicaciones web	67
3.1.1	Modelo Estructural	67
3.1.2	Modelo Funcional.....	69
3.1.3	Modelo de Interfaz de usuario.	70
3.2	Propuesta atributos, métricas e indicadores y medición del modelo.	72
3.2.1	Métricas del diseño estructural:	74

3.2.2	Métricas del diseño funcional.....	76
3.2.3	Medición del diseño de Interfaz de Usuario.....	78
3.3	Indicadores del modelo propuesto	80
3.4	Validación del modelo	86
3.4.1	Caso Estudio Proyecto 1:.....	87
3.4.1.1	Medición:	88
3.4.2	Caso Estudio Proyecto 2:.....	91
3.4.2.1	Medición:	91
3.4.3	Caso Estudio Proyecto 3:.....	96
3.4.3.1	Medición:	96
4	Conclusiones y Trabajos futuros	102
4.1	Conclusiones	102
4.2	Trabajos futuros	104
5	Bibliografía.....	105
5.1	Referencias Bibliográficas	105
5.2	Referencias web	106
6	Anexos.....	110

Lista de Figuras

Figura 1. Tipos de entidades de medición del software. Fuente: Medición y Estimación de software. Piattini (2003)	15
Figura2. Modelo de Boehm divide la calidad en tres características. Fuente: (OLSINA, 2002).....	20
Figura 3. Metodología GQM. Fuente:Medición y Estimación de software. Piattini (2003)	21
Figura 4. Jerarquía de la Metodología SAQE. Tomado de Scalone (2006).	27
Figura 5. Mapeo de las áreas de Calidad y Factores de Calidad de la Metodología SQAE. Tomado de Scalone Fernanda (2006).	28
Figura 6.Pirámide del diseño. Rogeer Pressman (2010)	35
Figura 7. Árbol de requisitos de calidad (Olsina,1999).....	37
Figura 8.Fuente:Pressman(2010).....	39
Figura 9. Descripción Jerárquica del Modelo Quint2. Fuente Fernanda Scalone (2006)	48
Figura 10. Modelo WQM. Fuente: Piattini (2008).....	54
Figura 11. Estructura general de los modelos de calidad. Fuente: Los autores.....	63
Figura 12. Modelo de Calidad para aplicaciones web en fase de diseño. Área-Factor- Características.....	66
Figura 13. Ejemplo de modelo relacional. Fuente: Los autores.	68
Figura 14. Ejemplo de cómo se representa una funcionalidad de forma gráfica. Navegabilidad.....	69
Figura 15. Gráfico del diseño funcional completo para una aplicación web.	70
Figura 16. Ejemplo de diseño de interfaz gráfica abstracta para el modelo de aplicaciones web de la Universidad Popular del Cesar Seccional Aguachica. Fuente los autores.	71

Figura 17. Interfaz gráfica abstracta. Caso ejemplo. Fuente los autores.	72
Figura 18. Modelo de calidad interna del producto para aplicaciones web en fase diseño. Características y atributos. Fuente: Los autores.	82
Figura 19. Modelo Estructural Caso Estudio Proyecto1.. Fuente: Proyecto de grado. Pallares (2011).	88
Figura 20. Modelo Estructural Caso Estudio Proyecto2. Fuente: Proyecto de grado. Barrios (2012).	92
Figura 21. Diseño Navegacional presentado Caso estudio Proyecto2. Proyecto de grado Barrios (2012).	93
Figura 22. Modelo Estructural Caso Estudio Proyecto3. Fuente: Proyecto de grado .León (2013).	97

Lista de Tablas

Tabla 1. Relación entre los factores de calidad y las métricas de McCall. Fuente: Los autores.....	19
Tabla 2. Características o criterios asociados a los factores o categorías de calidad. Tomado de Solano (2013)	24
Tabla 3. Metas, atributos y métricas según el Modelo SATC. Tomado de Scalone Fernanda (2006).....	26
Tabla 4. Modelo para la calidad externa e interna. Tomado de Piattini (2009).	30
Tabla 5. Modelo de Calidad del Producto Norma ISO/IEC 25010. Adaptado de ISO/IEC 25010 (2011).	33
Tabla 6. Relación ISO/IEC 9126 y SQUARE. Tomado de Scalone(2006)	34
Tabla 7. Fases de la Metodología UWE y los principales artefactos. Fuente: Los autores.	41
Tabla 8 Comparación de los diferentes modelos revisados Parte 1. Fuente los autores	58
Tabla 9 Comparación de los diferentes modelos revisados Parte 2. Fuente los autores	58
Tabla 10 Características que más persisten en los modelos de calidad del producto. Fuente los autores	59
Tabla 11 Plantilla de Definición de GQM. Pitanni 2009, Calidad de Sistemas Informáticos.....	65
Tabla 12 Plantilla de Definición para modelo de indicadores para la fase de diseño de aplicaciones web. Fuente los autores.....	65
Tabla 13. Artefactos de diseño para una aplicación web para la Universidad Popular del Cesar. Fuente los autores.....	67
Tabla 14. Presencia de las características de Usabilidad y Mantenibilidad en los modelos de calidad del producto estudiados. Fuente: Los autores.....	73

Tabla 15. Características de Usabilidad y Mantenibilidad de ISO 9126 con sus atributos. Fuente: Calidad de Sistemas Informáticos de Piattini (2009)..	73
Tabla 16. Atributos para la característica de Usabilidad para el modelo de indicadores de las Aplicaciones Web en fase diseño de la Universidad Popular del Cesar Seccional Aguachica. Fuente los autores.	74
Tabla 17. Atributos para la característica de Usabilidad para el modelo de indicadores de las Aplicaciones Web en fase diseño de la Universidad Popular del Cesar Seccional Aguachica. Fuente los autores.	74
Tabla 18. Atributos y métricas para la medición de mantenibilidad del modelo estructural.	75
Tabla 19. Atributos y métricas de la medición de usabilidad del modelo estructural	76
Tabla 20. Atributos y métricas para la medición del modelo funcional	77
Tabla 21. Atributos y métricas para la medición de la Usabilidad del modelo funcional.	78
Tabla 22. Métricas, indicadores para medir la mantenibilidad del modelo Interfaz gráfica. Fuente los autores.	79
Tabla 23. Atributos y métricas para la medición de la Usabilidad del modelo de interfaz gráfica.	79
Tabla 24. Rangos de aceptabilidad del modelo de indicadores para aplicaciones web en la fase de diseño. Fuente: Los autores.	81
Tabla 25. Ejemplo de agregación de pesos a los atributos y características para el cálculo final del factor de calidad interna del producto. Fuente: Los autores.	82
Tabla 26. Atributos, métricas y tipo de indicador usabilidad del modelo estructural.	83
Tabla 27. Atributos, métricas y tipo de indicador mantenibilidad del modelo estructural.	83
Tabla 28. Atributos, métricas y tipo de indicador usabilidad del modelo funcional.	84

Tabla 29. Atributos, métricas y tipo de indicador mantenibilidad del modelo funcional.	84
Tabla 30. Atributos, métricas y tipo de indicador usabilidad del modelo de interfaz gráfica.	85
Tabla 31. Atributos, métricas y tipo de indicador mantenibilidad del modelo de interfaz gráfica.	85
Tabla 32. Evaluación de la calidad interna Modelo Estructural. Caso estudio: Proyecto 1.	89
Tabla 33. Evaluación de la calidad interna Modelo Funcional. Caso estudio: Proyecto 1.	90
Tabla 34. Evaluación de la calidad interna Modelo Interfaz Gráfica. Caso estudio: Proyecto 1.	91
Tabla 35. Evaluación de la calidad interna Modelo Estructural. Caso estudio: Proyecto 2.	94
Tabla 36. Evaluación de la calidad interna Modelo Funcional. Caso estudio: Proyecto 2.	95
Tabla 37. Evaluación de la calidad interna Modelo Interfaz Gráfica. Caso estudio: Proyecto 2.	96
Tabla 38. Evaluación de la calidad interna Modelo Estructural. Caso estudio: Proyecto 3.	98
Tabla 39. Evaluación de la calidad interna Modelo Funcional. Caso estudio: Proyecto 3.	99
Tabla 40. Evaluación de la calidad interna Modelo Interfaz Gráfica. Caso estudio: Proyecto 3.	100
Tabla 41. Resumen de evaluación de los tres proyectos aplicando el modelo de indicadores para aplicaciones web en fase de diseño.	100

Tabla 42. Resumen de evaluación de usabilidad de los modelos estructural, funcional e interfaz gráfica en los tres proyectos.	100
Tabla 43. Tabla 44. Resumen de evaluación de mantenibilidad de los modelos estructural, funcional e interfaz gráfica en los tres proyectos.	101

1 Introducción

Es importante para cualquier desarrollador de software conocer el estado de su producto en etapas tempranas de desarrollo y qué mejor que sea en el diseño, fase donde afirma Pressman (2010) se establece la calidad del software.

Las aplicaciones web poco a poco han ido posicionándose en el mercado y van creciendo los desarrollos en esta área, de tal forma que ha surgido lo que podría ser una nueva disciplina llamada ingeniería web o Iweb, que es la que se encarga de todos los temas concernientes al desarrollo de aplicaciones para la web. Este proyecto tiene como objeto de estudio lo concerniente al diseño de aplicaciones para la web y específicamente en su medición.

En el primer capítulo se dan a conocer los aspectos que representan la investigación, el porqué de la misma, las acotaciones y todo lo concerniente al diseño de la investigación, con los métodos que llevaron a que poco a poco se fuese recopilando la información necesaria para desarrollar esta propuesta.

En el segundo capítulo se encuentran temas concernientes a la calidad del software, la medición del mismo, la conceptualización de diseño, de aplicación web y la historia de cómo han venido apareciendo diferentes propuestas de modelos de calidad para el producto software hasta encontrar los modelos de calidad para las aplicaciones web específicamente. Se presenta cada modelo y las intenciones de medición que en su momento generaron la necesidad de su creación, sus factores, características y atributos asociados. Se presenta igualmente el estado del arte en forma resumida a través de una figura que permite ver cómo se hicieron aportes a través del tiempo.

En el capítulo tres, ya conocidos los diferentes modelos de calidad para la web, se plantea en primer lugar la propuesta de cuáles son los entregables de diseño para las aplicaciones web de la Universidad Popular del Cesar en el Programa de Ingeniería de Sistemas así como también las plantillas que resumen información relevante sobre cada entregable, este aporte pretende enriquecer los manuales que son parte importante para la mantenibilidad del software característica ésta de importancia pues en el mantenimiento es donde se generan los mayores gastos en cuanto al recurso

Financiero (Swebok_Guide 2004), es en esta fase es donde más inversión se requiere, y el hecho de tener éstas plantillas aseguran las posibilidades de fácil entendimiento¹ del software en la fase de diseño.

También se encuentran en éste capítulo las métricas seleccionadas con sus indicadores para realizar la medición a los artefactos de diseño propuestos, cada artefacto es medido a través de un ejemplo que sirve para entender el modelo y su aplicación, se genera una herramienta que ayuda a automatizar parte del proceso de medición de los artefactos y se desarrolla en Excel. En éste capítulo se hace una propuesta muy novedosa respecto a medir todas las métricas a través de un único indicador.

Por último se realiza la validación del modelo de indicadores para aplicaciones web en la fase de diseño en tres proyectos terminados que se catalogan como aplicaciones web.

Al final se presentan unas conclusiones y unas recomendaciones para trabajos futuros ya que en la medición del software lo que queda es camino por recorrer pues esta área aún está muy joven y se presentan diferencias en las posiciones de muchos investigadores quienes aún no confían del todo en las algunas de las propuestas de medición realizadas hasta ahora, para los proyectos del Programa de Ingeniería de Sistemas de la Universidad Popular de Aguachica este es uno de más acercamientos que se vendrán a futuro con la medición de software.

1.1 Planteamiento del Problema y Justificación

El desarrollo de software ha venido cobrando importancia y hoy por hoy es una de las actividades representativas en el sector de servicios. Cada día la miniaturización en los elementos electrónicos que se usan, van generando nuevas necesidades de aplicaciones para el usuario final y esto genera que las exigencias sean cada vez mayores en cuanto a su uso y calidad. Sin embargo, estudios que ha venido desarrollando el StandishGroup² como son los informes “CHAOS” **Piattini (2009)**, la situación que se venía presentando en los fallos de software desarrollados en el año 2001 se debían a casos de atrasos del tiempo estimado, trayendo como consecuencia que muchos de los proyectos no finalizaran a tiempo, algunos no se terminaran, que terminaran con menos de las funcionalidades previstas o con fallos, Piattini (2009). La situación no ha mejorado en

¹ Esta definición está relacionada con el atributo Entendibilidad del modelo WQM

² Grupo éste que trabaja en retomar información de proyectos de aplicación de TI y que genera datos importantes de lo que muchas empresas vienen realizando

los informes posteriores que han seguido generando y por ello se discute mucho acerca de la calidad del software y las posibles soluciones para mejorar éste tema en particular.

En lo referente al desarrollo de software en el ámbito web, se puede afirmar y muchos teóricos como Reifer (2000) confirman, que el desarrollo de aplicaciones para sistemas web tiene diversas características que lo diferencian de un desarrollo de software tradicional. Este tipo de aplicaciones se han vuelto cada vez más complejas en cuanto a estructura, funcionalidad e interfaz. Un estudio realizado sobre los proyectos web, publicados por “Cutter Consortium (2000)”, describe los siguientes problemas: *“el 84% de los proyectos entregados no corresponden a las necesidades del negocio; el 79% de las veces ocurren retrasos en la entrega; el 63% de las veces los proyectos exceden el presupuesto; el 53% de las veces las aplicaciones web entregadas no proporcionan la funcionalidad requerida; el 52% de las veces las aplicaciones web entregadas tienen baja calidad”*, Piattini (2009). Todas estas fallas que se vienen presentando en este tipo de sistemas generan descontento y desconfianza en los sistemas de información web, por lo tanto el proceso se debe gestionar y dirigir de manera rigurosa y cuantitativa. Por ello: *“La utilización de principios de medición para evaluar el desarrollo web proporciona una retroalimentación que ayudará a entender, controlar, mejorar y predecir éstos productos y su proceso de desarrollo”*.

Piattini (2009) manifiesta que el tiempo, el coste y la calidad son variables del negocio a tener en cuenta en el desarrollo de software y cada desarrollador debe entender la economía y poder determinar el equilibrio óptimo del funcionamiento y el riesgo para su negocio y mercados si no implementa una manera de medir para poder controlar lo que está realizando.

Mantener controlado el producto que se está desarrollando en cada una de las fases es lo ideal en las empresas dedicadas a la producción de software. Sin embargo, la fase de desarrollo del software donde se aplica la ingeniería en toda su dimensión es la fase de diseño. El diseño es una fase esencial, la exigencia de calidad en ésta redundará en el mejoramiento del desarrollo del código y en la disminución del esfuerzo realizado en la fase de mantenimiento y no debemos olvidar que los costos más altos se realizan en esta fase, por ello una mejora del diseño conlleva una disminución de costes de mantenimiento.

Por otro lado, en el ámbito académico de los programas de las Instituciones de Educación Superior que trabajan en torno al desarrollo de software en Colombia, es recomendable tener en cuenta de igual forma el hecho de medir para reconocer el estado de los productos generados por ello que se busca aplicar en un número considerable de proyectos de grado donde desarrollen aplicaciones para la web éste modelo de indicadores. En el caso de la Universidad Popular del Cesar, específicamente en el programa de Ingeniería de Sistemas se viene desarrollando proyectos de grados tipificados en aplicaciones para la web y a los cuales es deseable aplicar un modelo de indicadores en la fase de diseño que ayude a mejorar los índices de calidad de los mismos.

El aporte de éstos resultados enriquecerá en la línea de ingeniería del software a investigadores y desarrolladores de aplicaciones para la web, pues el modelo será un punto de referencia en el ámbito de medición de la calidad de éste tipo de software que viene aumentando en exigencias por parte de los usuarios finales a un ritmo muy acelerado, por lo que medir un producto en la etapa de diseño y permitir conocer su grado de calidad dará la confianza de seguir por buen camino el ciclo de vida del mismo. La etapa de diseño es una de las primeras fases del ciclo técnico de la ingeniería del software y como manifiesta Pressman (2010) la importancia de esta fase se puede describir en una palabra – calidad.

Toda lo anteriormente expuesto hace que se formule la siguiente pregunta:

¿Cómo se puede evaluar la calidad de un diseño de software para la web, aplicable a los proyectos de grado del programa de ingeniería de sistemas de la Universidad Popular del Cesar Seccional Aguachica?

1.2 Objetivo General

Definir un modelo de indicadores para el diseño de aplicaciones web, aplicado a los trabajos de grado de ingeniería de sistemas de la Universidad Popular del Cesar Seccional Aguachica.

1.3 Objetivos Específicos

Establecer un estado del arte sobre los modelos de calidad para la fase de diseño de software para la web, basados en métricas o indicadores del producto.

Determinar los indicadores para evaluar la calidad del diseño de software para la web, adaptados al contexto y tipos de desarrollos realizados en el Programa de Ingeniería de Sistemas de la Universidad Popular del Cesar.

Validar el modelo de indicadores propuestos mediante la ejemplificación de su aplicación en proyectos desarrollados al interior del programa de ingeniería de sistemas.

1.4 Acotaciones

Las mediciones se pudieron haber realizado para el proyecto, el proceso y el producto. Sin embargo, éste trabajo se centró en los aspectos relativos al producto, sin excluir casos donde se haga necesaria la gestión en la fase de diseño. Teniendo en cuenta lo antes mencionado, no se tuvo en cuenta la calidad en uso, puesto que la investigación está más direccionada a la calidad interna del producto que a la externa y a la de uso que se menciona en la ISO 9126.

1.5 Metodología de Investigación

En esta sección se mencionará toda la información relativa los métodos que se aplicaron para lograr alcanzar los objetivos propuestos. Las herramientas y la forma como se abordó la investigación, la forma como se buscarán los resultados a través de todo un marco metodológico.

1.5.1 Enfoque Cualitativo

Hernández, et al., (2003), afirman que el enfoque cualitativo es el que “utiliza recolección de datos sin medición numérica para descubrir o afinar preguntas de investigación y puede o no probar hipótesis en su proceso de interpretación”. Este estudio se plantea desde un enfoque cualitativo pues soporta sus resultados en evidencias empíricas como la observación directa realizada sobre los proyectos de grado sobre los cuales se aplicó el modelo, brindando la posibilidad de pasar de la casuística, de lo informal a conocimientos más elaborados que permitan generalizaciones, aunque no necesariamente universales, pero si con la posibilidad de aplicarse en contextos

similares al del objeto de estudio que en este caso son las aplicaciones web en fase de diseño.

Se logra desarrollar un proceso inductivo al formular posturas teóricas como el modelo propuesto para los diseños de aplicaciones web que incluye la propuesta de un indicador para medir todos los atributos propuestos bajo un accionar de explorar y describir algo del objeto de estudio que son los diseños de las aplicaciones web para luego generalizarlo a los proyectos de grado catalogados como tal de la Universidad Popular del Cesar Seccional Aguachica. Para ello se requiere conocer cuál es la percepción de diseños de calidad para las aplicaciones web y así generar la propuesta.

La mayor parte de datos recolectados no serán de medición numérica sino más bien de documentación recolectada para ser analizada y con ellos: realizar la propuesta, aplicarla a dos casos de la Universidad y luego construir un marco de referencia basado en los datos de tipo empírico obtenidos y analizados.

1.5.2 Alcance descriptivo

Se pretende describir con este proyecto los factores, las características, los atributos y por último las métricas que hacen que una aplicación web sea considerada o no un producto de calidad. Y posteriormente plantear un modelo de calidad desde las necesidades de la Institución para posteriormente aplicarlo en algunos de los proyectos de grado desarrollados o en fase de desarrollo. Hernández, et al., (2003), define la investigación descriptiva como el tipo de investigación que “busca especificar propiedades, características y rasgos importantes de cualquier fenómeno que se analice”. Y ese es el caso de esta investigación.

1.5.3 Diseño del proceso de investigación

El diseño metodológico de esta investigación propone diversos métodos y técnicas de corte inductivo como la observación, la clasificación, estudio de hechos como investigaciones particulares de medición, la derivación inductiva y la contrastación, sin embargo se adapta a las circunstancias del contexto durante el desarrollo del estudio, porque así lo permiten los enfoques cualitativos, claro está sin perder el rigor que un trabajo de éste tipo requiere.

La metodología se explica a continuación.

1.5.4 Métodos para la consecución del primer objetivo

Para el logro de este objetivo se realizará un recorrido histórico por los diferentes modelos de calidad del producto de software a nivel general hasta llegar a los modelos de calidad del producto definidos para aplicaciones web, se describirán sus factores, características hasta llegar a los atributos pues las métricas harían demasiado extenso este objetivo y el fin es conocer qué existe y cómo se aplica cada modelo según el tipo de software para luego realizar una comparación entre éstos modelos y verificar cuáles son las características que más están presentes en todos. De igual forma se describirá lo que se considera a nivel de estándares como calidad, diseño y aplicación web para luego tener una certeza de los artefactos de cada aplicación web en fase de diseño sobre los que debería aplicarse la medición, concepto éste del cual también se hará una claridad.

Una vez expuestos todos los conceptos y aplicado el método histórico para los modelos de calidad del producto y realizada la comparación entre los diferentes modelos se pasará a selección de los artefactos a medir para plantear el modelo de indicadores en la fase de diseño de aplicaciones web.

1.5.5 Métodos para la consecución segundo objetivo

Vistos los diferentes modelos de calidad del producto y los fines de su medición, se inicia la propuesta del modelo sobre el planteamiento de la necesidad de medición de los proyectos de grado enmarcados en aplicaciones web en su fase de diseño de la Universidad Popular del Cesar Seccional Aguachica. Se expresa una pregunta alrededor de éste objetivo de medición y luego se plantean las características relevantes que se desean medir.

Paso seguido se deben seleccionar los artefactos a medir en la fase de diseño para delimitar qué medirá el modelo. Una vez se seleccionan éstos artefactos se proponen plantillas para su análisis, esto basado en las características que ya han sido escogidas para la medición.

El análisis desarrollado para diseñar las plantillas ayudará a seleccionar los atributos y las métricas de cada atributo por el método de abstracción científica.

Una revisión bibliográfica ayudará de igual forma a abstraer los indicadores que se conocen en el medio y sus posibles formas de medición. De aquí se seguirá analizando las mejores formas de plantear los indicadores y los rangos que algunos miembros de la

comunidad científica proponen (pues algunos son de tipo empírico y otros de tipo teórico) como una medición que se pueda considerar aceptable en el rango de la calidad que se pretende obtener con este modelo de diseño y de medición.

1.5.6 Métodos para la consecución tercer objetivo

Para validar el nivel de funcionalidad del modelo se pretende aplica en tres proyectos de grado en fase de diseño que configuren como aplicaciones web. La revisión realizada a los proyectos de la Institución se hizo mediante métodos de revisión documental.

Una vez seleccionados los tres proyectos, mediante revisión documental del listado de proyectos de grado que existen en la coordinación de investigación y proyectos se seleccionan tres que se tomarán para la validación. Se verifican los resultados y se hace un análisis de los mismos.

1.5.7 Fuentes de Datos Consultadas

Se realizó una triangulación de datos de las fuentes consultadas como artículos científicos, libros sobre el tema de calidad, experiencia del docente que dirige los proyectos de grado de la Universidad y enlaces web de interés, así como las recomendaciones y conclusiones de varios trabajos de grado que son aplicaciones web. Esto permitió hacer las posibles confrontaciones y abstracciones para plantear el modelo de indicadores.

1.5.8 Instrumentos de captura de información

Se utilizaron varias técnicas para la captura de datos, esto de alguna forma lo permitía la diversidad metodológica que se propuso. Las técnicas son las siguientes:

1.5.8.1 La técnica de revisión documental.

Para la aplicación de esta técnica se utilizó el instrumento del Anexo A, la intención es revisar si los proyectos de grado terminados y que se encuentran en la biblioteca entregaron artefactos susceptibles de medir para el modelo.

Para la selección del material que sirvió como fuente bibliográfica no se utilizó ningún tipo de herramienta, sólo la observación directa del temade cada libro o artículo realizando una lectura al prefacio y al resumen de los mismos previamente con esto se podía saber si el documento era importante para la investigación.

Para la selección del proyecto en fase de desarrollo se realizó observación directa sobre el cronograma entregado en el formato de anteproyecto a la oficina de investigación y proyectos de la Institución.

1.5.8.2 La técnica de conversación asociada a la entrevista.

Para la aplicación de esta técnica no se utiliza ningún formato específico, lo único que se pretende es conocer los artefactos de diseño que recomienda el docente que dirige los proyectos de grado configurados como aplicaciones web en la Universidad Popular del Cesar Seccional Aguachica y qué consideración tiene sobre lo que es un buen diseños para los proyectos.

2 Marco Teórico

El marco teórico se desarrolla en primer lugar definiendo la calidad de software de acuerdo a ciertos autores y estándares, pues este trabajotiene como objeto de estudio la calidad de los diseños de las aplicaciones Web. Para conocer la calidad de un producto de software se debe establecer una línea base sobre los factores que se quieren medir para luego seleccionar las características, sus atributos y las medidas que se toman gracias a las métricas de software. Por ello dentro en calidad de software se definen términos como medición, métrica e indicador.

La siguiente sección corresponde a los Modelos de calidad y estándares del producto, existen diverso tipos de modelos y estándares que se han venido desarrollando a través del tiempo sobre el tema de calidad en el software, se hace una descripción de cada uno de éstos modelos enfocados al producto, pues además de medir la calidad del producto, de igual forma se puede medir el proceso y el proyecto. Al pretender medir la calidad en la fase de diseño se ha delimitado esta investigación al ámbito del producto y los modelos esbozados describen los más relevantes en orden cronológico de aparición. Muchos de éstos modelos más adelante se convirtieron en la base para otros que tuviesen relación con modelos para aplicaciones web, dentro de éstos el que más ha sido usado como base para otros es la norma 9126.

Se continúa en éste marco teórico con la definición de lo que es diseño en el proceso del desarrollo del software, esta fase donde se aplica en todas sus dimensiones el componente ingenieril y en donde muchos autores como Pressman (2010), afirma se define y se aplica la calidad en el software.

Luego se considera importante definir lo que es una aplicación web para tener claro cuál es el ámbito sobre el cual se propondrá el modelo de indicadores en la fase de diseño; esta delimitación hace que se enfoque aún más la medición a un ámbito de software.

Una vez teniendo claridad de lo que es una aplicación web se define lo que se puede considerar calidad en éstas aplicaciones mencionando algunos modelos³ que se han desarrollado en éste tipo de productos de software, para pasar al diseño de aplicaciones web donde se tienen claro cuáles son los componentes que hacen parte del diseño en aplicaciones web, que luego se convertirán en los objetos sobre los cuales se determinará la calidad en la fase de diseño.

Por último se realiza una descripción detallada de los modelos antes mencionados sobre calidad del producto web y algunas investigaciones que se han venido desarrollando a través del tiempo. Es bueno tener claridad que no se encuentran hasta la fecha modelos de calidad para el producto específicamente enfocados a medir la fase de diseño ni a nivel de aplicaciones tradicionales ni a nivel web.

Se finaliza con un resumen en forma gráfica del estado del arte de investigaciones desarrolladas para medir artefactos para la web.

2.1 Calidad del software

La Real Academia de la Lengua Española, define calidad como “la propiedad o conjunto de propiedades inherentes a algo, que permiten juzgar su valor”⁴. Estas propiedades permiten conocer en qué estado están las características de ése algo una vez sea medido. En el caso del software estas propiedades son características intrínsecas al software y que también pueden y deben medirse.

Pressman (2010) define la calidad del software como “concordancia del software producido con los requisitos específicamente establecidos, con los estándares de desarrollo expresamente fijados y con los requisitos implícitos, no establecidos formalmente, que desea el usuario”. Según la IEEE, Std. 610-1990, Diccionario de estándar de términos de computación (IEEE, 1990), “es el grado con el que un sistema, componente o proceso cumple los requerimientos especificados y las necesidades o expectativas del cliente o usuario”

³ WQM, WebQem, Quint2, entre otros.

⁴ <http://dle.rae.es/?id=6nVpk8Pl6nXVL1Z>

Entre las instituciones a nivel internacional que primero se preocuparon por el tema de la calidad en el software está el SEI⁵ quien veía necesario el establecimiento de métricas para los contratistas de la rama. Como respuesta al tema de la estandarización de procesos en el desarrollo de software la ISO crea su apartado de calidad para los mismos con la ISO9001:1994 que fue una respuesta al modelo CMM⁶ del SEI.

Con respecto a la medición del software es bueno tener claro los siguientes términos:

2.1.1 Medición

Al hablar de calidad del software debemos pensar en la medición y es bueno tener claro que una de las razones principales del incremento masivo en el interés en la medición del mismo ha sido la percepción de que las métricas son necesarias para la mejora de la calidad del proceso (Fenton, 2001).

Fenton y Pfleeger (1997) definen la medición como: “el proceso de asignar números o símbolos a los atributos de las entidades del mundo real de forma que se puedan describir de acuerdo a unas reglas claramente definidas”.

La métrica tomada como una forma de medición del software y cuya definición está dada en la norma técnica ISO/IEC 14598-1, como el método de medición y la escala de medición, será la definición que se tomará de la misma en el presente trabajo.

La unidad de medición es “una cantidad particular, definida y adoptada por convención, con la que se puede comparar otras cantidades de la misma clase para expresar sus magnitudes respecto a esa cantidad particular”.

2.1.2 Métrica

Según Fenton (2001) “la métrica es la correspondencia de un dominio empírico (mundo real) a un mundo formal (matemático). La medida incluye al valor numérico o nominal asignado al atributo de un ente por medio de dicha correspondencia”.

El objetivo principal de una métrica es valorar y estimar, se habla de valorar en el sentido de magnitudes tales como la calidad, la fiabilidad y la productividad. Y en cuanto a la estimación se tiene en cuenta características tales como el tiempo y el

⁵ (Software Engineering Institute), creado por el departamento de defensa de los Estados Unidos

⁶ (CapabilityMaturityModel), modelo propuesto por el SEI que determina la capacidad de de los procesos software de una organización (Piattini, 2009).

esfuerzo. Los indicadores son medidas derivadas de otras medidas utilizando un modelo de análisis como forma de medir, ésta será la definición formal de indicadores que ha sido tomada del libro de Piattini (2009).

La escala en las métricas se conceptúa como un conjunto de valores con propiedades definidas (ISO 14598-1), las escalas se clasifican así: nominal, ordinal, intervalo, proporción, absoluta; cada una define la forma como se realizan las transformaciones admisibles para ellas, el tipo de operaciones matemáticas y los análisis estadísticos que pueden aplicarse sobre el conjunto de valores de la escala.

A continuación se presenta un resumen de cada uno de los tipos de escalas admisibles con sus posibles valores, estadística significativa y ejemplos:

El tipo de escala indica la naturaleza de la relación entre los valores de la escala. Según Piattini(2009) se destacan estos cinco tipos principales de escalas:

Escala Nominal. Esta escala sitúa las entidades en clases o categorías asignando al atributo un nombre⁷.

Escala Ordinal. Los atributos son ordenados en rangos, sin embargo, la distancia entre ellos no es significativa. Un ejemplo de este tipo de escalas en una encuesta pueden ser: “0” totalmente en desacuerdo, “1” en desacuerdo, “2” de acuerdo, 3 “totalmente de acuerdo”. La distancia entre un rango y otro no es necesariamente igual; es decir la distancia entre 0-1 no necesariamente es la misma que entre 2-3.

Escala Intervalo. Es igual que la escala ordinal, sólo que cuando medimos los atributos a través de esta escala la distancia si tiene sentido.

Escala Ratio. Esta escala tiene en cuenta el orden, el tamaño de los intervalos y también los ratios entre las entidades. El punto fijo de referencia es el cero y se debe incrementar en pasos iguales. Es la escala más útil en la medición de software.

Escala absoluta. Que es la más restrictiva de todas y es utilizada únicamente cuando sólo hay una forma posible de medir un atributo: contado.

⁷Éstas se identifican únicamente mediante un número o un nombre. Son el tipo de escala más básica.

A continuación se presenta el resumen de propiedades de las escalas según criterios definidos por Maxwell (2000).

Tipo de Escala	¿Hay un orden significativo?	¿Las mismas distancias entre valores tienen significado?	¿El cálculo de ratios tiene sentido?
Nominal	No	No	No
Ordinal	Sí	No	No
Intervalo	Sí	Aproximadamente	No
Quasi-Intervalo	Sí	Sí	No
Ratio	Sí	Sí	Sí
Absoluta	Sí	Sí	Sí

2.1.3 Unidad de medición.

La unidad de medición es una cantidad particular, definida y adoptada por convención, con la que se puede comparar otras cantidades de la misma clase para expresar sus magnitudes respecto a esa cantidad particular, (Piattini, 2008).

En lo referente a las formas de medir, se define como la secuencia lógica de operaciones descritas de forma genérica, usadas para realizar mediciones de un atributo respecto de una escala específica. También se encuentran medidas derivadas que se calculan a partir de las medidas base, (Piattini, 2008).

Al final la medición se trata de obtener información que satisfaga una necesidad detectada por una empresa en la que se lleva a cabo el software. Luego de tener clara las necesidades se establecen los atributos objetos de medición y se definen las métricas necesarias (Piattini, 2008).

Las métricas no pueden interpretar por sí solas un concepto medible, es de allí que viene el concepto de indicadores.

2.1.4 Indicador

El método de cálculo y la escala definidos, además del modelo y criterios de decisión con el fin de proveer una evaluación o estimación de un concepto medible con respecto a una necesidad de información. Piattini (2003).

2.2 Modelos y estándares de calidad del producto software

Para realizar la medición del software y así conocer el estado de la calidad del mismo tanto los diferentes representantes de la industria como investigadores de diferentes

latitudes han creado diferentes modelos o estándares para brindar un marco que pueda ser una guía para las empresas y los desarrolladores de este producto como lo es el software.

Algunos autores han propuesto que la solución al problema de evaluar la calidad del software, pasa por la descomposición de los conceptos genéricos de calidad más sencillos de medir y evaluar, dándole el nombre de **Modelo de Calidad** a este tipo de descomposición. Cataldi (2000).

La ISO 9126-1(ISO/IEC 9126) define el modelo de calidad como: “El conjunto de características y las relaciones entre las mismas, que proveen la base para especificar requerimientos de calidad y evaluar calidad”.

Muchos de estos modelos están enfocados hacia éstos tres aspectos importantes sobre los cuales se deben desarrollar las mediciones, los cuales se mencionan a continuación (Piattini, 2003):

- Sobre el proceso de software
- Sobre el proyecto de software
- Sobre el producto de software

Esto indica que se encuentran métricas para el proceso, métricas para el proyecto y para el producto. La relación entre ellas se muestra en la siguiente figura:

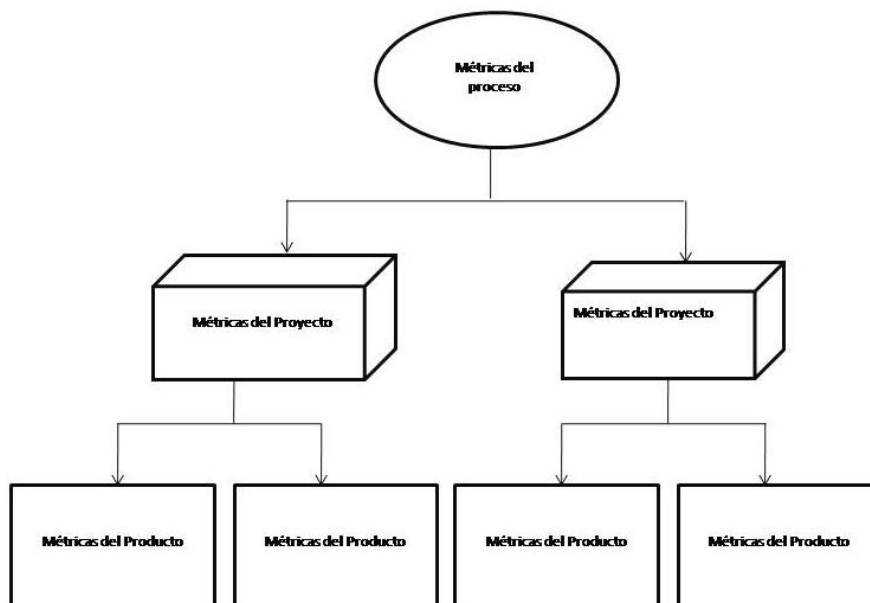


Figura 1. Tipos de entidades de medición del software. Fuente: Medición y Estimación de software. Piattini (2003)

Los esfuerzos de la industria se han enfocado en su gran mayoría en la aplicación de modelos de calidad del proceso y en menor medida a los modelos de producto. Chrissis, Konrad, & Shrum (2007).

Algunos expertos en el tema plantean que cumplir con un modelo de calidad de procesos aporta a la calidad del producto pero no la garantiza. Kitchenham & Pfleeger (1996). Los procesos estandarizados, también llamados institucionalizados, no necesariamente llevan a obtener productos de calidad haciendo necesario que las evaluaciones de calidad deban basarse en evidencias extraídas directamente de los atributos del producto. Tom & Alan (2008).

El tema central de ésta investigación son las métricas relativas al producto de software, específicamente al producto del diseño de aplicaciones web. En primera instancia se presentará lo relacionado con las métricas relativas al producto, para luego hacer énfasis en las de aplicaciones web en el diseño en el caso específico del tema tratado a través de todo este documento se centrará en los modelos o estándares de calidad del producto software.

2.2.1 Modelo de McCall

Este es uno de los modelos clásicos para evaluar la calidad del software, fue desarrollado por MacCall (1977) y de él se derivan otros como el de Boehm (Scalone, 2006); éste modelo está compuesto por 11 características o factores de calidad las cuales se organizan en tres categorías.

Categorías:

Operación del producto

Revisión del producto

Transición del producto

Características de Operación del producto:

1) Corrección: Es el grado en que un programa satisface sus especificaciones y consigue los objetivos pedidos por el cliente. La pregunta asociada es: ¿Hace lo que quiero?

- 2) **Fiabilidad:** Es el grado en que se puede esperar que un programa lleve a cabo sus funciones esperadas con la precisión requerida. La pregunta asociada es: ¿Lo hace de forma fiable todo el tiempo?
- 3) **Eficiencia:** Es la cantidad de recursos de computadoras y de código requeridos por un programa para llevar a cabo sus funciones. La pregunta asociada es: ¿Se ejecutará en mi hardware lo mejor que pueda?
- 4) **Integridad:** Consiste en la capacidad de un sistema para resistir ataques contra su seguridad. La pregunta asociada es: ¿Es seguro?
- 5) **Facilidad de uso:** Es un intento de cuantificar “lo amigable que puede ser con el usuario”. La pregunta asociada es: ¿Es fácil de usar?

Características Revisión del producto (Capacidad para soportar cambios):

- 6) **Facilidad de mantenimiento:** Es el esfuerzo requerido para localizar y arreglar un error en un programa. La pregunta asociada es: ¿Puedo corregirlo?
- 7) **Flexibilidad:** Es el esfuerzo requerido para modificar un programa que ya está en funcionamiento. La pregunta asociada es: ¿Puedo cambiarlo?
- 8) **Facilidad de prueba:** Es el esfuerzo requerido para probar un programa de forma que se asegure que realiza su función requerida. La pregunta asociada es: ¿Puedo probarlo?

Características Transición del producto (Adaptabilidad de nuevos entornos):

- 9) **Transportabilidad:** Es el esfuerzo requerido para transferir el programa desde un hardware y/o un entorno de sistema de software a otro. La pregunta asociada es: ¿Podré usarlo en otra máquina?
- 10) **Reusabilidad:** Es el grado en que un programa (o partes de este) se puede reusar en otras aplicaciones, en relación al empaquetamiento y alcance de las funciones que realiza el programa. La pregunta asociada es: ¿Podré reusar alguna parte del software?
- 11) **Interoperabilidad:** Es el esfuerzo requerido para acoplar un sistema con otro. La pregunta asociada es: ¿Podré hacerlo interactuar con otro sistema?

Estas 11 características a su vez se descomponen en métricas.

En el esquema de puntuación se usan las siguientes **métricas de calidad** del software:

Facilidad de Auditoria: Es la facilidad con que se puede comprobar la conformidad con los estándares.

Exactitud: Consiste en la precisión de los cálculos y el control.

Estandarización de Comunicaciones: Es el grado de empleo de estándares de interfaces, protocolos y anchos de banda.

Completitud: Es el grado con que se ha logrado la implementación total de una función.

Concisión: Consiste en lo compacto que es el programa en términos de líneas de código.

Consistencia: Es el uso de un diseño uniforme y de técnicas de documentación a lo largo del proyecto de desarrollo de software.

Estandarización en los datos: Consiste en el uso de estructuras de datos y de tipo estándar a lo largo de todo el programa.

Tolerancia de Errores: Es el daño que se produce cuando el programa encuentra un error.

Eficiencia en la Ejecución: Es el rendimiento en tiempo de ejecución de un programa.

Capacidad de expansión: Es el grado en que se puede ampliar el diseño arquitectónico de datos o procedural.

Generalidad: Consiste en la amplitud de aplicación potencial de los componentes del programa.

Independencia del Hardware: Es el grado en que el software es independiente del hardware en que opera.

Instrumentación: Es el grado en que el programa muestra su propio funcionamiento e identifica errores que aparecen.

Modularidad: Consiste en la independencia funcional de los componentes del programa.

Facilidad de Operación: Es la facilidad de operación de un programa.

Seguridad: Consiste en la disponibilidad de mecanismos que controlen o protejan los programas o datos.

Auto-Documentación: Es el grado en que el código fuente proporciona documentación significativa.

Simplicidad: Es el grado de facilidad con que se puede entender un programa.

Independencia del sistema de software: Es el grado de independencia del programa respecto a las características del lenguaje de programación no estándar, características del sistema operativo y otras restricciones del entorno.

Trazabilidad: Consiste en la capacidad de seguir una representación del diseño o componentes reales del programa hasta los requisitos.

Formación: Es el grado en que ayuda el software a manejar el sistema a los nuevos usuarios.

La relación entre los factores de calidad del software y las métricas se muestra en la siguiente tabla:

Métrica de la Calidad del Software	Factor de la Calidad										
	1	2	3	4	5	6	7	8	9	10	11
Facilidad de auditoria				X			X				
Exactitud		X									
Estandarización de comunicación										X	
Compleción	X										
Complejidad		X				X	X				
Concisión			X		X	X					
Consistencia	X	X			X	X					
Estandarización de datos										X	
Tolerancia a errores		X									
Eficiencia de ejecución			X								

Capacidad de expansión						X					
Generalidad						X		X	X	X	
Independencia del hardware								X	X		
Instrumentación				X	X		X				
Modularidad		X			X	X	X	X	X	X	
Operatividad			X								X
Seguridad				X							
Auto documentación					X	X	X	X	X		
Simplicidad		X			X	X	X				
Independencia del sistema								X	X		
Trazabilidad	X										
Facilidad de formación											X

Tabla 1. Relación entre los factores de calidad y las métricas de McCall. Fuente: Los autores.

2.2.2 Modelo Boehm (1978)

Se caracteriza por centrarse en la calidad del producto final, agrega algunas características al modelo de McCall.

El modelo de Boehm tiene como finalidad que a través de la calidad del software, el mismo: (1) realice lo que desea el usuario, (2) utilice recursos informáticos de manera correcta y eficiente, (3) sea fácil de utilizar y aprender; y (4) sea bien diseñado, codificado, probado y mantenido. Este modelo es similar al de McCall ya que presenta una jerarquía de características, está basado en un amplio rango de características e incorpora 19 criterios que incluyen características de performance del hardware.

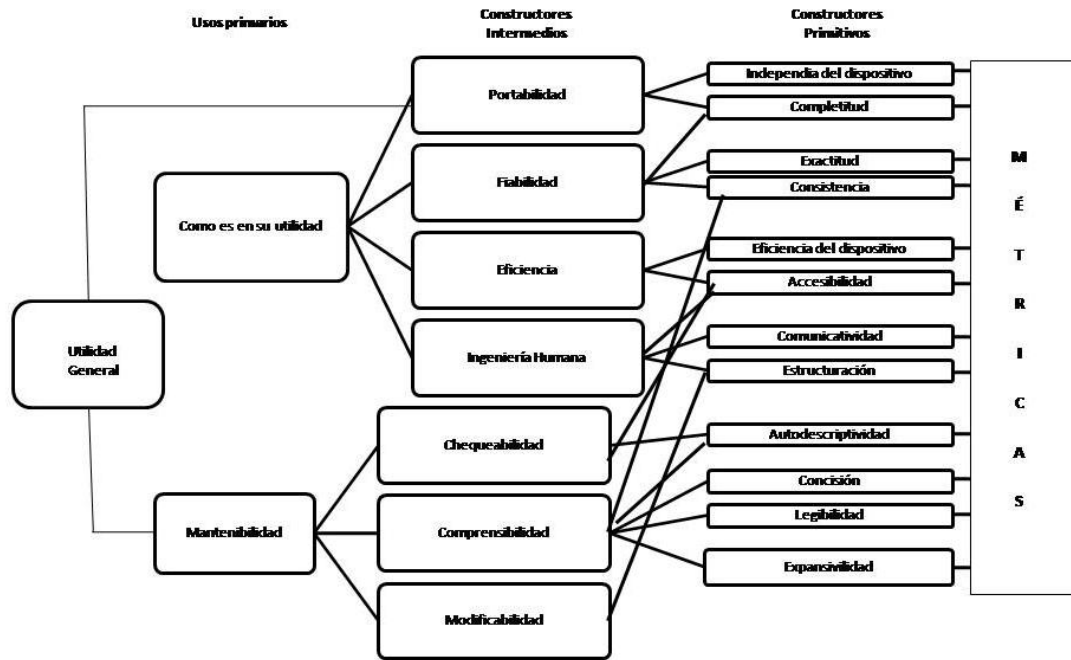


Figura2. Modelo de Boehm divide la calidad en tres características. Fuente: (OLSINA, 2002)

2.2.3 Modelo Gilb (1988)

El modelo de Gilb plantea la creación de una especificación de requisitos de calidad para cada proyecto que deben escribir conjuntamente el usuario y el analista. Es un modelo que permite determinar una lista de características que definen la calidad de la aplicación.

Puede ser de 2 tipos: (1) Originales y (2) de modelos tradicionales.

Las características se pueden medir mediante varias sub-características o métricas detalladas. Para cada una de ellas, se deben especificar los siguientes conceptos:

1. Nombre y definición de la característica.
2. Escala o unidades de medición.
3. Recopilación de datos o prueba.
4. Valor previsto.
5. Valor óptimo.
6. Valor en el sistema actual.

7. Comentarios.

2.2.4 Modelo GQM (Goal – Question – Metric):

Este modelo planteado por Basili y Weiss (1984) tiene la particularidad que las mediciones se hacen sobre objetivos que se trazan comenzando el desarrollo del proyecto. “GQM define un objetivo, refina éste objetivo en preguntas y define métricas que intentan dar información para responder a éstas preguntas”.

En el siguiente gráfico se muestran las fases del método GQM que son en su orden la planificación, la definición, la recopilación de datos y la interpretación.

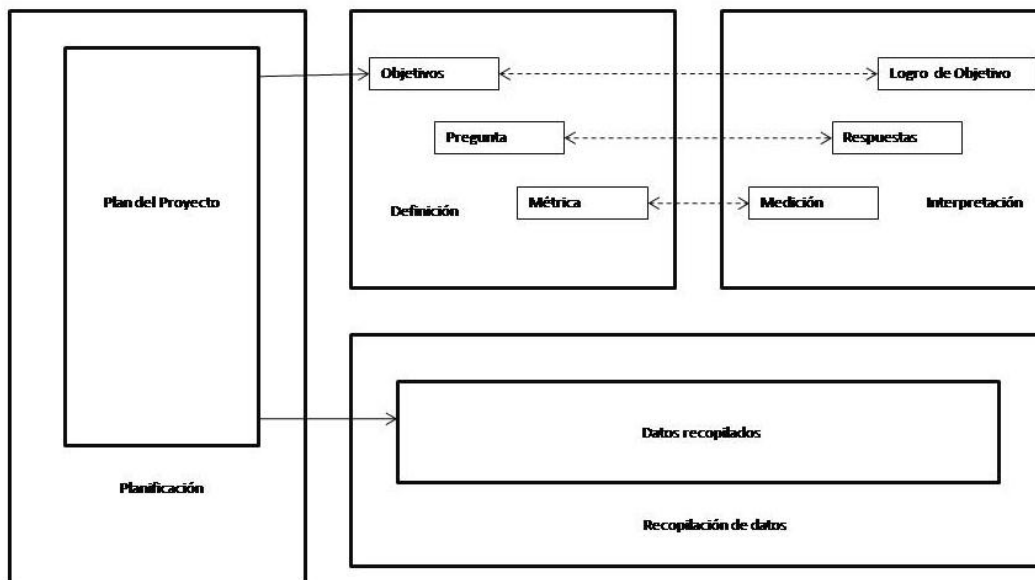


Figura 3. Metodología GQM. Fuente: Medición y Estimación de software. Piattini (2003)

Este modelo permite elegir las métricas que se relacionen a las metas más importantes de los problemas más urgentes.

2.2.5 Modelo FURPS

Modelo de calidad propuesto por Robert Grady y Hewlett Packard Co (HP) en 1987. Esta propuesta contempla, por un lado 5 características de las cuales se deriva su nombre (Funcionalidad, Facilidad de Uso, Confiabilidad, Performance y Facilidad de Soporte), y por otro, que los requisitos se clasifiquen en dos categorías: requisitos funcionales (F), que son los que especifican funciones que el sistema debe ser capaz de

realizar sin tener en cuenta las restricciones físicas; y requerimientos no funcionales (URPS), que puntualizan atributos del sistema o del medioambiente del sistema.

MODELO	CATEGORÍA	CARACTERÍSTICA
FURPS	Requerimientos Funcionales (F)	Funcionalidad
	Requerimientos No Funcionales (URPS)	Usabilidad
		Confiabilidad
		Desempeño
	Facilidad de soporte	

Funcionalidad - Los requisitos de funcionalidad deben incluir: Conjunto de Características, Capacidades, Seguridad.

Facilidad de Uso - Deben incluir sub-categorías tales como: Factores humanos, Estéticos, Consistencia en la Interfaz de Usuario, Ayuda en línea, Asistentes, Documentación del usuario, Material de capacitación.

Confiabilidad - Se considera requisitos de confiabilidad: Frecuencia y severidad de fallas, Recuperación a fallos, Tiempo entre fallos.

Performance (Rendimiento)- Un requisito de rendimiento impone condiciones a los requisitos funcionales. Por ejemplo a una acción dada, se pueden especificar los siguientes parámetros de rendimiento: Velocidad, Eficiencia, Disponibilidad, Tiempo de Respuesta, Tiempo de Recuperación, Utilización de Recursos.

Soporte - Los requisitos de soporte pueden incluir: Requisitos de instalación, Requisitos de Configuración, Requisitos de Adaptabilidad, Requisitos de Compatibilidad.

Según Scalone (2006), FURPS se aplica realizando los siguientes pasos:

- (1) Asignación de prioridades.
- (2) Definición de los atributos de calidad que pueden ser medidos.

Para controlar la calidad en el proceso de fabricación de su hardware, HP contempla un concepto global bajo las siglas FURPS: Funcionalidad, Facilidad de uso, Confiabilidad (Reliability), Prestaciones (Performance) y Servicio. Se trata de variables sobre las que se incide durante todo el ciclo de vida del producto compuesto por diversas fases. La primera es la concepción del producto a partir de análisis del mercado, después se

marcan los objetivos y se define el proyecto, apartado que se mezcla inevitablemente con la fase de diseño real del producto e investigación. Posteriormente, se procede a traducir los diseños en piezas o prototipos reales en el laboratorio. El Manufacturing Release es la fabricación real a partir de los prototipos. Una vez fabricado, la siguiente fase en la vida del producto es su comercialización hasta que se convierta en algo obsoleto y se deje de fabricar como última etapa del ciclo.

2.2.6 Modelo DROMEY

El modelo DROMEY presentado por Robert Dromey (1995), aparece como una propuesta que permite construir y utilizar un modelo de calidad práctico para evaluar etapas de: determinación de los requerimientos, diseño e implementación.

Este modelo plantea la calidad del producto por medio de la definición de sub-características además que organiza tres modelos para cada etapa del proceso de desarrollo:

Modelo de requerimientos.

Modelo de diseño.

Modelo de implementación.

El modelo de calidad utiliza las características de: Eficiencia, Confiabilidad, Facilidad de Mantenimiento, Portabilidad, Facilidad de Uso y Funcionalidad. Y unas categorías que implican propiedades de calidad que son: Correctitud, internas, contextuales, y descriptivas, (Solano, 2013).

La relación existente entre estas categorías y las características se presentan en el siguiente cuadro.

Factor	Criterio
Correctitud	Funcionalidad Confiabilidad
Internas	Mantenibilidad Eficiencia Confiabilidad
Contextuales	Mantenibilidad Reusabilidad Portabilidad Confiabilidad
Descriptivas	Mantenibilidad Reusabilidad

	Portabilidad Usabilidad
--	----------------------------

Tabla 2. Características o criterios asociados a los factores o categorías de calidad. Tomado de Solano (2013)

Los pasos para la aplicación del modelo de Dromey son los siguientes, Scalone (2006):

Seleccionar el conjunto de atributos que se necesitan evaluar.

Realizar una lista de todos los componentes o módulos del sistema.

Identificar las propiedades de calidad de cada componente.

Determinar cómo afecta cada propiedad en los atributos de calidad.

Evaluar el modelo de calidad.

2.2.7 Modelo SATC (Software Assurance Technology Center)

Los modelos de calidad mencionados anteriormente no entran en el detalle de las métricas necesarias para el área de ingeniería de requisitos exceptuando el Dromey, al ser modelos que proporcionan un entorno de trabajo y unas directrices para crear el modelo de calidad de software adecuado a nuestras necesidades. En 1996 el [SATC \(Software Assurance Technology Center\) de la NASA](#) estableció su propio modelo de calidad de software, siguiendo el modelo ISO 9126.

En el año 1996 cuando el SATC decidió las métricas que iban a formar parte de su modelo de calidad para requisitos, tomó como criterio de decisión que las mediciones a realizar tuvieran un coste razonable en su ejecución, es decir, que la cantidad de tiempo y dinero invertido en obtener los datos no fuera desorbitado. Es por esa razón que el número de métricas no es grande y se dejaron fuera del modelo otra serie de factores que habrían ayudado a conseguir una mayor precisión en la calidad de las especificaciones de requisitos software de la NASA.

Este modelo utiliza un amplio rango de medidas o métricas y; tiene objetivos, atributos y métricas asociadas a los procesos de desarrollo y al software propiamente dicho. Este modelo define un conjunto de metas u objetivos relacionados al producto de software y atributos del proceso que permiten realizar indicaciones de la probabilidad de éxito de los objetivos. Un conjunto de métricas es seleccionado o desarrollado, las que medirán los atributos seleccionados, Scalone (2006).

Stac posee cuatro metas u objetivos mencionados a continuación, éste material se ha tomado exactamente del proyecto de Fernanda Scalone (20006):

1- Calidad de los Requerimientos: El objetivo de esta meta es que los documentos de requerimientos estén completos, no ambiguos y entendibles.

Esta meta tiene los siguientes atributos:

Ambigüedad: Requerimientos con múltiples significados.

Integridad:Item a ser especificados.

Facilidad de entender: Documento legible.

Trazabilidad: Trazabilidad de los requerimientos generales respectodelCódigo y de las pruebas.

2- Calidad del Producto: Un objetivo importante de un proyecto de desarrollo de software es desarrollar código y documentación que se correspondan con los requerimientos del proyecto. Esta meta u objetivo tiene los siguientes atributos:

Estructura / Arquitectura: La evaluación de un módulo para identificar posibles errores e indicar problemas potenciales en la *facilidad de uso y facilidad de mantenimiento*.⁸

Reutilización: Utilizar el software en diferentes contextos o aplicaciones.

Facilidad de mantenimiento: Es el esfuerzo requerido para localizar y corregir un error enun programa.

Documentación: Tener la adecuada documentación del código interno y la documentación externa.

3- Efectividad de la implementación: El objetivo de la efectividad de la implementación es maximizar la efectividad de los recursos dentro de las actividades programadas en el proyecto. Los atributos de este objetivo son:

Uso del recurso: El uso del recurso relacionado a la etapa apropiada del proyecto.

Cumplimiento de los porcentajes: Avances realizados en los ítems.

⁸ Son las características que más adelante serán seleccionadas en el modelo a proponer.

4- Efectividad de la prueba: Los objetivos de la prueba de efectividad es ubicar y reparar las fallas del software. El atributo es la corrección. Una vez generado el código, se realizan pruebas de unidades, pruebas finales y pruebas de aceptación.

La relación entre las metas, los atributos y las métricas se definen en la siguiente tabla:

Metas	Atributos	Métricas
Calidad de los Requerimientos	Ambigüedad Integridad Facilidad de entender Volatilidad del Requerimiento Trazabilidad	Nro de frases claras Nro de frases opcionales Nro de TBDs/TBAs Estructura del documento Cantidad de cambios / cantidad de requerimientos Etapa del ciclo de vida cuando se realiza un cambio Nro de requerimientos del software que no se ajustan a los requerimientos del sistema Nro de requerimientos del software que no se ajustan al código y a las pruebas
Calidad del Producto	Estructura / Arquitectura Facilidad de Mantenimiento Reusabilidad Documentación interna Documentación externa	Complejidad lógica Uso del GOTO Tamaño Correlación de complejidad / tamaño Correlación de complejidad / tamaño Porcentaje de comentarios Índice legible
Efectividad de la implementación	Uso de los recursos Porcentaje de terminación	Horas staff dedicadas a las actividades del ciclo de vida Tareas terminadas Tareas terminadas planificadas
Efectividad de la prueba	Corrección	Errores y criticidad Tiempo de encuentro de errores Tiempo de errores fijos Ubicación del código de falla

Tabla 3. Metas, atributos y métricas según el Modelo SATC. Tomado de Scalone Fernanda (2006)..

2.2.8 Metodología SQAE

Esta metodología desarrollada por el MITRE Corporation cuyo objetivo es producir un sistema de evaluación que satisfaga el objetivo de producir resultados confiables en todas las etapas del ciclo de vida del software.

Scalone (2006) dice: “Es una metodología que permite cuantificar los riesgos asociados al software”. SQAE se crea basada en modelos como McCall, Boehm y Dromey.

La jerarquía de este modelo se presenta de la siguiente manera:

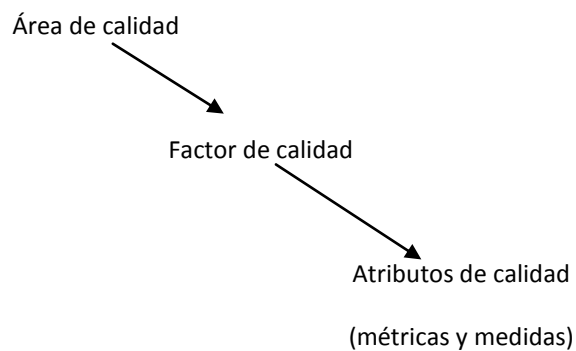


Figura 4. Jerarquía de la Metodología SAQE. Tomado de Scalone (2006).

Se definen cuatro áreas de calidad en la SQAE: Mantenibilidad, Evolutibilidad, Portabilidad, Descriptibilidad. Cada una con sus propios factores de calidad que proveen una estructura para medir la calidad de un sistema, éstas son: consistencia, independencia, modularidad, documentación, auto-descripción, control de fallos, simplicidad del diseño.

Las áreas de calidad se usan para definir los conceptos de riesgos del ciclo de vida y se expresan como la suma de varios factores que abarcan aspectos del concepto a medir. Estas áreas son definidas por medio de atributos y porcentajes usados en el proceso de Evaluación, Scalone (2006).

El uso de porcentajes en los factores de calidad deriva de las áreas de calidad. Por cada factor de calidad se tiene definido un mapeo entre el factor de calidad y una o más áreas de calidad. Cada factor de calidad está definido por un conjunto de atributos, los cuales

contemplan distintas facetas del factor de calidad en cuestión como lo indica la figura a continuación:

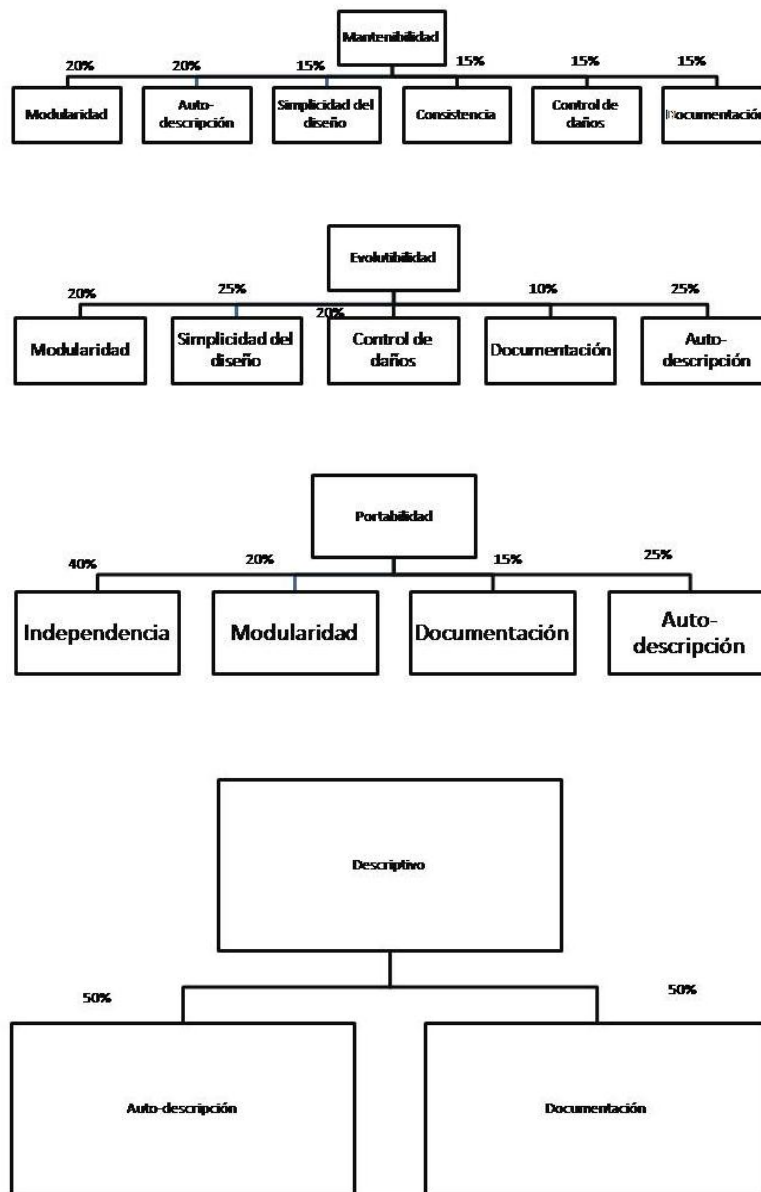


Figura 5. Mapeo de las áreas de Calidad y Factores de Calidad de la Metodología SQAE. Tomado de Scalone Fernanda (2006).

2.2.9 ISO 9126

Se presenta la primera versión en 1991, luego es presentada la ISO 9126:1 en el 2001 quedando como actualmente se conoce, la ISO 25000 luego es otra mejora a este modelo y que se presentará más adelante en este mismo capítulo. La ISO 9126 se

descompone en dos partes: Modelo para la calidad Externa e Interna y Modelo para calidad en uso.

La calidad externa evalúa que el software satisfaga las necesidades del usuario teniendo en cuenta las condiciones especificadas. Esta calidad es medible en el comportamiento del producto. La calidad interna evalúa el total de atributos que un software debe satisfacer teniendo en cuenta condiciones especificadas. Esta calidad es medible a partir de las características intrínsecas, Scalone (2006).

La ISO 9126 descompone la calidad en seis factores que se mencionan a continuación:

Funcionalidad.

Confiabilidad.

Usabilidad.

Eficiencia.

Capacidad de mantenimiento.

Portabilidad.

El modelo se desglosa de la siguiente manera:

ISO 9126-1: Modelo de Calidad.

ISO 9126-2: Métricas Externas.

ISO 9126-3: Métricas Internas.

ISO 9126-4: Calidad en Uso.

Según ISO/IEC 9126-1 (2001), algunos atributos pueden contribuir a más de una sub-característica y una característica puede ser influenciada por más de un atributo. Se puede presentar también que los niveles de ciertos atributos internos influyan en los niveles de algunos atributos externos. Por ejemplo, la característica de confiabilidad puede ser medida externamente mediante la observación del número de fallas en un período determinado de tiempo de ejecución durante una prueba del software, e internamente mediante la inspección de las especificaciones detalladas y el código

fuente para evaluar el nivel de tolerancia a fallos; se dice que los atributos internos son indicadores de los atributos externos.

2.2.9.1 Modelo de calidad interna y externa

La calidad externa e interna posee seis características las cuales a su vez presentan sub-características presentadas en la siguiente tabla:

Calidad externa e interna	Funcionalidad	Adecuación Exactitud Interoperabilidad Seguridad de acceso Cumplimiento de la Funcionalidad
	Confiabilidad	Madurez Tolerancia a fallos Capacidad de recuperación Cumplimiento de la fiabilidad
	Usabilidad	Capacidad de ser entendido Capacidad de ser aprendido Capacidad de ser operado Capacidad de atracción Cumplimiento de usabilidad
	Eficiencia	Comportamiento temporal Utilización de recursos Cumplimiento de la eficiencia
	Capacidad de mantenimiento	Capacidad de ser analizado Capacidad de ser cambiado Capacidad de ser probado Cumplimiento de la mantenibilidad
	Portabilidad	Adaptabilidad Instalabilidad Coexistencia Capacidad de ser reemplazado Cumplimiento de portabilidad

Tabla 4. Modelo para la calidad externa e interna. Tomado de Piattini (2009).

Las siguientes definiciones son tomadas de Piattini (2009):

Funcionalidad

Capacidad del producto de software para proporcionar funciones que satisfacen necesidades declaradas e implícitas cuando se usa bajo condiciones especificadas.

Fiabilidad

Capacidad del producto software para mantener un nivel especificado de prestaciones cuando se usa bajo condiciones especificadas.

Usabilidad

Capacidad del producto software para ser entendido, aprendido, usado y ser atractivo para el usuario, cuando se usa bajo condiciones especificadas.

Eficiencia

Capacidad del producto software para proporcionar prestaciones apropiadas, relativas a la cantidad de recursos usados, bajo condiciones determinadas.

Mantenibilidad

Capacidad del producto software para ser modificado. Las modificaciones podrían incluir correcciones, mejoras o adaptación del software a cambios en el entorno, y requisitos y especificaciones funcionales.

Portabilidad

Capacidad del producto software para ser transferido de un entorno a otro.

2.2.9.2 Modelo de calidad en uso

Esta parte presenta el concepto de calidad en uso proponiendo un modelo de calidad en uso que permite visibilizar la interrelación entre el usuario y el producto desde la óptica de la eficiencia y la satisfacción. En ISO/IEC 9126-1 (2001), se define la calidad de uso como la capacidad que tiene un producto software para facilitar que usuarios específicos alcancen metas específicas con eficacia, productividad, seguridad y satisfacción en contextos específico de uso.

Las características de la calidad en uso son las siguientes:

Efectividad, Productividad, Seguridad de uso, Satisfacción. Este modelo no es de interés para objetivos de este proyecto por tal razón no se ahondará en él.

2.2.10 ISO/IEC 25000: SQUARE

Presentada en el 2005 la ISO/IEC se basa en una combinación de la ISO 9126 y en ISO 14598 (Evaluación del software). Este conjunto de normas deriva su nombre de Software Quality Requirements and Evaluation y tiene como objetivo fundamental

guiar el desarrollo de los productos de software con la especificación y evaluación de requisitos de calidad.

La ISO/IEC 25000 se centra más en el producto de software, es una mejora respecto a la 9126 presentando algunos cambios en las características y sub características de calidad interna/externa y de calidad en uso de la misma.

La ISO 25000 se divide en:

ISO/IEC 2500n – División de Gestión de Calidad, Piattini (2009). Las normas que forman este apartado definen todos los modelos, términos y definiciones comunes referenciados por todas las otras normas de la serie SQUARE.

ISO/IEC 2501n – División de Modelo de Calidad, Piattini (2009). La norma de este apartado presenta un modelo de calidad detallada incluyendo características para calidad interna, externa y en uso.

ISO/IEC 2502n – División de Medición de Calidad, Piattini (2009). Estas normas incluyen un modelo de referencia de la medición de la calidad del producto, definiciones de medidas de calidad (interna, externa y en uso) y guías prácticas para su aplicación.

ISO/IEC 2503n – División de Requisitos de Calidad, Piattini (2009). Estas normas ayudan a especificar requisitos de calidad que pueden ser utilizados en el proceso de elicitación de requisitos de calidad del producto software a desarrollar o como entrada del proceso de evaluación.

ISO/IEC 2504n – División de Evaluación de Calidad, Piattini (2009). Este apartado incluye normas que proporcionan requisitos, recomendaciones y guías para la evaluación de productos software.

De estas divisiones ampliaremos la que tiene relación con el modelo de calidad el 2501n, se establecerán las características y sub-características que se desarrollan en la ISO 25010 que presenta algunas modificaciones respecto a la 9126.

	Característica	Sub-característica
	Funcionalidad	Compleitud Corrección Idoneidad
	Rendimiento	Comport. En el tiempo

Calidad externa e interna		Utilización de recursos
	Compatibilidad	Coexistencia Interoperabilidad
	Usabilidad	Inteligibilidad Aprendizaje Operabilidad Protección a errores de usuario Atractividad Accesibilidad
	Confiabilidad	Madurez Disponibilidad Tolerancia a fallos Capacidad de recuperación
	Seguridad	Confidencialidad Integridad No repudio Autenticidad Responsabilidad
	Capacidad de mantenimiento	Modularidad Reusabilidad Capacidad de ser analizado Capacidad de ser cambiado Capacidad de ser probado
	Portabilidad	Adaptabilidad Instalabilidad Intercambiabilidad

Tabla 5. Modelo de Calidad del Producto Norma ISO/IEC 25010. Adaptado de ISO/IEC 25010 (2011).

Una de las nuevas características la compatibilidad, hace referencia a la posibilidad de intercambio de información entre sistemas o componentes. La otra característica, seguridad, es incorporada para indicar el grado de disponibilidad y protección de la información o datos frente a accesos no autorizados.

El modelo de calidad en uso presenta respecto a la 9126 una característica más, es decir presenta cinco características. La nueva característica denominada contexto de uso, indica el grado de uso con efectividad, eficiencia, seguridad y con satisfacción en el contexto especificado.

Como se mencionó en el apartado de la ISO 9126, la calidad en uso está fuera de los alcances de este proyecto, por tanto no se ampliará más sobre éste.

La relación entre ISO/IEC 9126 y SQUARE se determina en la siguiente tabla:

ISO/IEC	SQUARE
9126-1: QualityModel	25000: Guide to SQUARE 25010: Quality Model and Guide

9126-2: Externalmetrics	25020: Measurement referente model and guide 25023: Measurement of external quality
9126-3: Internalmetrics	25020: Measurement referente model and guide 25022: Measurement of internal quality
9126-4: Quality in use metrics	25020: Measurement referente model and guide 25024: Measurement of quality in use

Tabla 6.Relación ISO/IEC 9126 y SQUARE. Tomado de Scalone(2006)

2.3 Diseño

El diseño una de las etapas del desarrollo de software definida en la norma [IEEE610.12-90] como “el proceso para definir la arquitectura, los componentes, los interfaces, y otras características de un sistema o un componente” y “el resultado de este proceso.” es de especial importancia pues es de destacar que gracias a un buen diseño se podrá generar un código más eficiente y un proceso de mantenimiento donde haya que emplear menos esfuerzo.

Para Pressman (2010) “el diseño del software es un proceso iterativo mediante el cual los requisitos se traducen en un «plano» para construir el software. Inicialmente, el plano representa una visión holística del software. Esto es, el diseño se representa a un nivel alto de abstracción -un nivel que puede rastrearse directamente hasta conseguir el objetivo del sistema específico y según unos requisitos más detallados de comportamiento, funcionales y de datos-. A medida que ocurren las iteraciones del diseño, el refinamiento subsiguiente conduce a representaciones de diseño a niveles de abstracción mucho más bajos”

Cada uno de los elementos resultantes del modelo de requerimientos representa una información útil para la construcción de los cuatro modelos de diseño (Pressman 2010), éstos son el diseño de datos o clases, de la arquitectura, de la interfaz y de los componentes.

La siguiente figura tomada del libro Ingeniería del Software un enfoque práctico de Pressman (2010) nos muestra los productos del modelo de diseño.



Figura 6. Pirámide del diseño. Rogeer Pressman (2010)

El diseño de datos o clases que define las estructuras de datos que se requieren para implementar el software, el diseño de la arquitectura que define la relación entre los elementos principales de la estructura del software, el diseño de la interfaz que describe la forma en la que el software se comunica con los sistemas que interactúan con él y los humanos que lo utilizan, el diseño a nivel de componentes que transforma los elementos estructurales de la arquitectura en una descripción de sus componentes en cuanto a procedimiento son los cuatro productos del diseño que serían las entidades a tener en cuenta en el proceso de medición.

Por último cabe destacar esta frase textual de Pressman (2010). “La importancia del diseño de software se resume en una palabra: calidad. El diseño es el sitio en el que se introduce la calidad en la ingeniería del software” y es una de las razones que motivan este estudio para las aplicaciones web que tanto auge tienen actualmente y que son motivo de muchos desarrollos en los proyectos de grado del Programa de Ingeniería de Sistemas y los cuales son desarrollados para la misma Institución o para empresas de la zona que así lo han requerido.

2.4 Aplicaciones web

Una aplicación es un software que es utilizado accediendo a un servidor web, a través de internet o de una intranet. Las aplicaciones web son populares debido a la practicidad del navegador web, que actualmente está disponible en diferente tipo de equipos de escritorio, notebooks, celulares, tablet, etc. Además de la facilidad para la actualización

y mantenimiento sin realizar instalaciones ni distribución del software en miles de potenciales usuarios.

2.4.1 Calidad en aplicaciones web:

Todas las personas que hayan navegado alguna vez por la Web o hayan utilizado una intranet de una empresa pueden opinar sobre lo que hace una «buena» WebApp. Los puntos de vista individuales varían enormemente. Algunos usuarios disfrutan con gráficos llamativos, en cambio otros solo quieren un texto sencillo. Algunos exigen información copiosa, otros desean una presentación abreviada. En efecto, la percepción de «lo bueno» por parte del usuario (y como consecuencia, la aceptación o no aceptación resultante de la WebApp) podría ser más importante que cualquier discusión técnica sobre la calidad de la WebApp.

En realidad, todas las características generales de la calidad del software se aplican también a las WebApps. Sin embargo, **las características más relevantes -usabilidad, fiabilidad, eficiencia y capacidad de mantenimiento-** proporcionan una base útil para evaluar la calidad de los sistemas basados en Web. Olsina (1999) y sus colaboradores han preparado un «árbol de requisitos de calidad» que identifica un conjunto de atributos que conduce a WebApps de alta calidad. La Figura resume su trabajo.

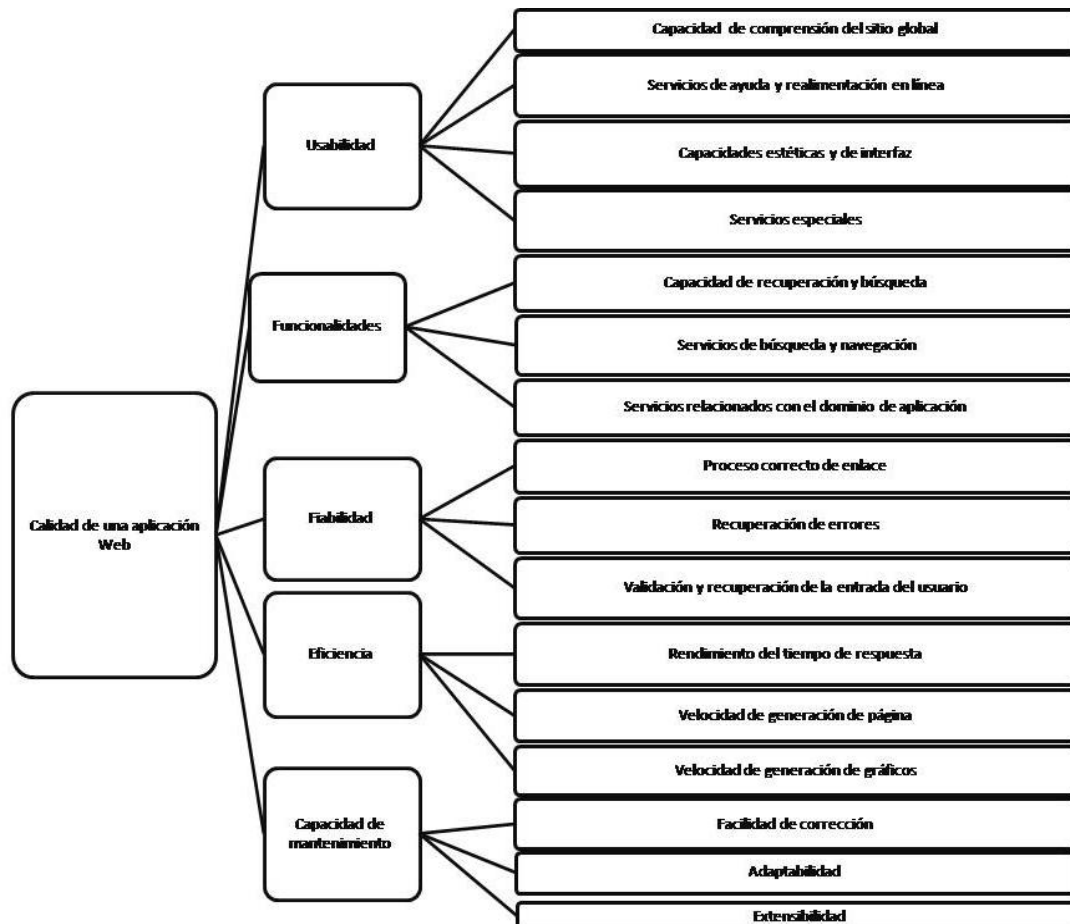


Figura 7. Árbol de requisitos de calidad (Olsina, 1999).

El diseño y la implementación de sistemas basados en Web incorporan tres tecnologías importantes: el desarrollo basado en componentes, la seguridad y los estándares de Internet. Un ingeniero Web deberá estar familiarizado con las tres para construir WebApps de alta calidad.

2.4.2 Desarrollo basado en componentes

Las tecnologías de componentes han evolucionado en gran parte gracias al crecimiento explosivo de los sistemas y aplicaciones basados en Web. Los ingenieros Web disponen de tres estándares importantes para la infraestructura: CORBA, COMDCOM y JavaBeans. Estos estándares (acompañados por los componentes pre-construidos, herramientas y técnicas) proporcionan una infraestructura que permite a los que diseñan emplear y personalizar componentes de terceras partes permitiéndoles así comunicarse unos con otros y con servicios a nivel de sistemas.

2.4.3 Seguridad

Si en una red reside una WebApp, ésta está abierta a un acceso sin autorización. En algunos casos, ha sido el personal interno el que ha intentado acceder sin autorización. En otros casos, intrusos (hackers) pueden intentar acceder por deporte, por sacar provecho o con intenciones más maliciosas. Mediante la infraestructura de red se proporciona una variedad de medidas de seguridad, tales como encriptación, cortafuegos y otras. Un estudio amplio de este tema queda fuera del ámbito de este trabajo.

2.4.4 Diseño de aplicaciones web

El diseño de una aplicación web lo resume Pressman (2010) de la siguiente manera: “La creación de un diseño eficaz requerirá por lo general de un contenido diversificado de aptitudes. En ocasiones, para proyectos pequeños, un desarrollador único necesitará tener varias habilidades. Para los proyectos grandes, es aconsejable o factible confiar en la experiencia de especialistas: ingenieros web, diseñadores gráficos, desarrolladores de contenido, programadores, especialistas de bases de datos, arquitectos de la información, ingenieros de redes, expertos en seguridad y probadores. Depender de estas distintas aptitudes permite la creación de modelos cuya calidad puede evaluarse a fin de mejorar su contenido y su código antes de que se generen contenido y código, de que se realicen pruebas y de que se involucre un gran número de usuarios. Si el análisis reside en donde se establece la calidad de la aplicación web, entonces el diseño está donde la calidad está en verdad incrustada”.

Para describir el diseño de aplicaciones web lo primero que se establecerá son los artefactos del diseño que se generan: diseño de los componentes, de la arquitectura, de la navegación, del contenido, estético y de la interfaz (Pressman, 2010). En este orden el diseño de componentes estaría más cerca la tecnología mientras el diseño de la interfaz estaría más cerca al usuario.

La pirámide de diseño de la Webapp que luego representará una acción del diseño para el diseñador se presenta a continuación:

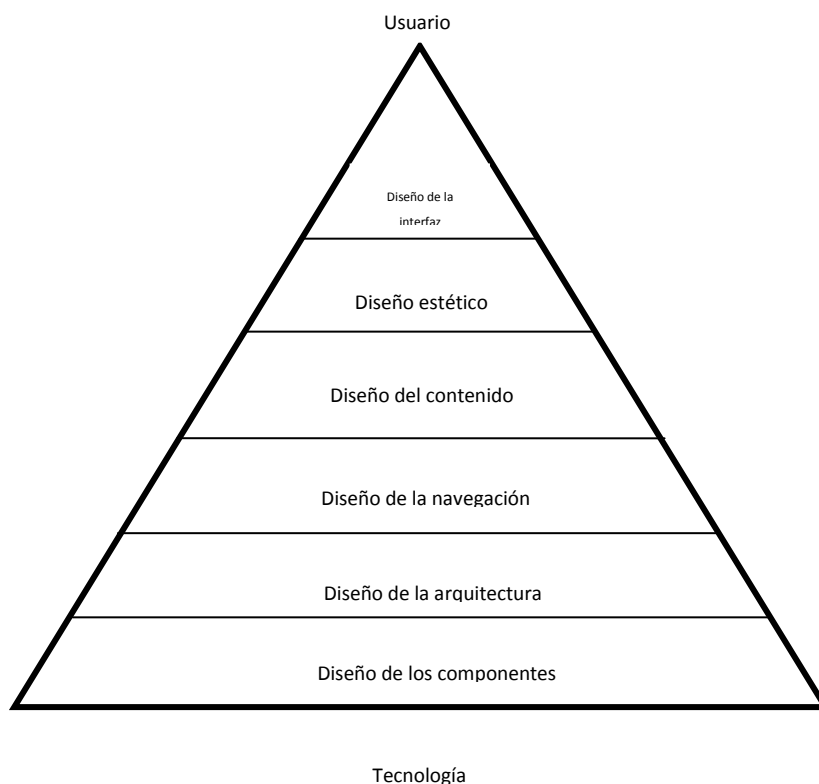


Figura 8. Fuente: Pressman (2010).

En este caso tendremos como usuario que recibe elementos del diseño al programador o al equipo de implementación cuando se trate de un proyecto de grandes dimensiones.

Olsina (2006) presentan un árbol de requerimientos de calidad, donde identifica los siguientes atributos técnicos: usabilidad, funcionalidad, confiabilidad, eficiencia y susceptibilidad de recibir mantenimiento.

2.4.4.1 Metodologías de diseño usadas en Ingeniería Web

La Ingeniería Web es una disciplina emergente que nace de la necesidad de realizar estudios más específicos sobre el desarrollo de aplicaciones para la Web, pues es claro que éste tipo de aplicaciones tiene unas nuevas implicaciones que no se pueden asumir con las metodologías y con muchas de las herramientas con las que se desarrollan las aplicaciones convencionales. Debido a que no es el objetivo de este proyecto el estudio de las metodologías, se mencionaran las más conocidas y se tendrá en cuenta en el

momento de analizar el modelo a proponer, pues muchas de éstas metodologías en sus fases indican los entregables que se dan en cada una de ellas y que serían los llamados a medir en el modelo que se propondrá.

Entre los métodos de la Iweb encontramos: Hypertext Design Model (HDM), Relationship Management Methodology (RMM), Object Oriented Hypermedia Design Method (OOHDM), Método OO-H, Object Oriented Web Solution (OOWS), UML Based Web Engineering (UWE), Del Valle(2009).

El principio que subyace en todas ellas es que una aplicación debe desarrollarse partiendo de su descripción precisa en forma de un esquema conceptual. En la mayoría de las veces así se den con nombres diferentes los métodos coinciden en las siguientes etapas:

- Diseño conceptual.
- Diseño navegacional.
- Diseño de la presentación o diseño de la interfaz.
- Implementación.

Como se puede ver con éstos tres diseños se realiza la implementación de una aplicación web en la mayoría de las metodologías.

El método de Ingeniería web basada en UML plantea los diseños anteriormente descritos. Incluye una fase de análisis de requisitos cuyo principal entregable es el diagrama de casos de uso, el diseño conceptual, que es representado principalmente por un diagrama de clases en UML, la construcción de este modelo conceptual se debe llevar a cabo de acuerdo con los casos de uso que se definen en la especificación de requisitos. El diseño navegacional que entrega como artefacto principal el modelo navegacional, el diseño de presentación que otros autores le denominan el de interacción que entrega como artefactos principales los historyboarding o visualización de escenarios web y el diagrama de secuencias. Estos modelos son los necesarios para lograr representar de una manera satisfactoria los elementos arquitectónicos significativos para una aplicación web (Ocaña, 2015).

Los entregables de cada fase se pueden organizar así:

Fase de Metodología UWE	Artefacto UML
Análisis de requisitos	Diagrama de casos de uso
Diseño conceptual	Diagramas de objetos y de clases

	Diagramas de componentes
Diseño Navegacional	Modelo Navegacional
Diseño de Presentación o Fase de Interacción	Diagrama de visualización de escenarios o Storyboarding Diagrama de secuencias
Implementación	No se encuentran

Tabla 7. Fases de la Metodología UWE y los principales artefactos. Fuente: Los autores.

Otro paradigma nuevo en el ámbito del desarrollo es el denominado MDD o metodología de desarrollo de software dirigida por modelos. Esta desarrolla los diseños en un mayor nivel de abstracción. Las motivaciones principales del MDD son la independencia de los productores de software a través de estandarizaciones y la portabilidad de los sistemas de software.

MDD separa el diseño del sistema de la arquitectura de las tecnologías, para que puedan ser modificados independientemente. Para lograr esto, se asigna a los modelos un rol central y activo bajo el cual se derivan modelos que van desde los más abstractos a los concretos, este proceso se realiza a través de transformaciones sucesivas e iteraciones.

Según Molina (2011) algunas ventajas de MDD son:

Proporciona una elevación del nivel de abstracción debido a que se estudian los niveles de conceptos cercanos al dominio del problema.

El código fuente final se genera a partir de especificaciones de alto nivel (automatización) proporcionando una mayor productividad, mayor calidad, favorece el mantenimiento, **los usuarios no expertos puede escribir códigos con más facilidad.**

Esta última ventaja es bueno considerarla pues los desarrolladores de los proyectos de la Universidad Popular del Cesar Seccional Aguachica son estudiantes del programa de Ingeniería de Sistemas que no son expertos en el desarrollo de aplicaciones. Además que el nivel de abstracción en los diseño también aportaría mucho a la sencillez del desarrollo. Faltaría verificar cómo es la calidad de este tipo de diseño y para eso debemos medirlo.

2.5 Modelos de calidad para las aplicaciones web

Al igual que los proyectos de desarrollo de software tradicional para los sistemas web existen las mediciones para el proceso y para el producto, en el caso particular de este estudio se mencionarán los avances en el tema de sistemas web referentes al producto.

2.5.1 W3C : (1999)

Grupo encargado de desarrollar protocolos y directrices que aseguren el crecimiento de la Web a largo plazo. Los estándares del W3C definen las partes claves que hacen que la World Wide Web funcione.

El objetivo del W3C es guiar la Web hacia su máximo potencial a través del desarrollo de protocolos y pautas que aseguren el crecimiento futuro de la Web, promoviendo la visión de Web Única.

2.5.2 Nielsen (2002):

En el año 2002 Nielsen y Tahir presentan una guía para el análisis de páginas de inicio de sitios web, en forma de 26 variables generales, que a su vez se dividen en 113 directrices específicas,(Noriega, 2009).

Tabla: Parámetros generales para medir la Usabilidad de un sitio Web

1. Tipo Artilugios de la interfaz.
2. Tipo de gestión de las emergencias relacionadas con el sitio.
3. Cantidad de tiempo de la recarga y actualización de la página.
4. Grado de precisión de la finalidad del sitio hacia el usuario.
5. Claridad de los títulos de ventana.
6. Claridad en el nombre del dominio.
7. Tipo de bienvenida y colocación.
8. Estructura de la información acerca de la empresa.
9. Grado de definición del área de navegación.
10. Grado de facilidad de las herramientas Búsqueda dentro del sitio.

11. Tipo de herramientas y accesos directos a tareas relacionadas con el sitio.
12. Claridad en la redacción de contenido.
13. Claridad en la estructura de la noticia.
14. Tipo de utilización de las fechas y horas en las noticias.
15. Grado de visualización de cotizaciones bursátiles y números.
16. Tipo de formato para la recopilación de datos del cliente.
17. Grado de fomento de una comunidad relacionada con el sitio.
18. Facilidad de acceso a archivos y contenido ya aparecido.
19. Grado de utilidad de los vínculos.
20. Cantidad de ventanas emergentes.
21. Cantidad de anuncios. Tipo de anuncios. Grado de rechazo al anuncio de acuerdo a un tipo.
22. Nivel de complejidad del diseño gráfico presentado al usuario.
23. Imágenes y animación.
24. Formato de los ejemplos utilizados.
25. Grado de exposición de los permisos y distinciones recibidos.
26. Tipo de personalización que se ofrece al usuario.

Esta metodología parte de la idea de que cuando se diseña un sitio web es necesario seguir ciertos parámetros mínimos para que quien acceda a ellos pueda tener una experiencia agradable, es decir, que encuentre lo que está buscando, que sepa utilizarlo, que comprenda cómo llevar a cabo acciones como comprar, que perciba las posibilidades de participación, entre otras.

2.5.3 Ivory Melody (2001)

“En su investigación sobre la medición de la usabilidad, Ivory pretende establecer modelos estadísticos que definan los perfiles de páginas web buenas y malas desde el

punto de vista de la usabilidad. Para ello comienza por realizar un estudio profundo de las diferentes pautas y guías de usabilidad existentes incluyendo los trabajos de Nielsen, Spool, Roosenfield o Moreville y a partir de ella establece una lista con los aspectos de una página web que resultan determinantes para que una sea usable. Entre otros aspectos se encuentran, por ejemplo, el número de enlaces, imágenes y palabras que tenga la página, el número de bloques de texto y sus cambios de posición, el número de colores utilizados y el uso de los mismos, el formato de los encabezados y la agrupación de los enlaces y los elementos interactivos. Con todo esto, establece las métricas que va a utilizar para evaluar las páginas y desarrolla un crawler y una herramienta para efectuar las medidas.

Para poder establecer con qué medidas se identifican las páginas buenas, Ivory efectúa las medidas sobre el conjunto de páginas web evaluadas para los WebbyAwards en el año 2000 y hace una comparación de sus resultados con las puntuaciones que recibieron las páginas en los premios” (Alonso, 2013).

Estos resultados muestran un fundamento empírico para las guías de diseño web y también sugieren qué métricas son importantes para la evaluación de la calidad de estos productos.

2.5.4 SUE: (1997)

A mediados de 1993 Paolini y Gaezotto, emplearon métodos y heurísticas de evaluación cualitativa en el campo de Hipermedia. Paolini y Garzotto avanzan en el proyecto SUE (Systematic Usability Evaluation) en el Laboratorio de Hipermedia del Politécnico de Milán. En uno de los primeros artículos de sus investigaciones describen un método de evaluación orientado al diseño. Básicamente, se podría describir al mismo como una evaluación heurística, cualitativa, centrada en el análisis del artefacto de hipermedia a partir de un modelo de diseño apropiado (como HDM) empleando, además, criterios generales de usabilidad, aunque no métricas cuantificables. Los autores identifican cinco dimensiones para evaluar aplicaciones de Hypermedia, a saber:

- Contenido, relacionado a los objetos de información incluidos en la aplicación
- Estructura, la organización del contenido

- Presentación, cómo se muestra a los lectores el contenido y las funciones de la aplicación
- Dinámica, cómo los usuarios se mueven e interactúan con los elementos y componentes de información
- Interacción, cómo se usan los componentes dinámicos

Argumentan que para un análisis sistemático de la aplicación de hipermedia, se necesita un modelo de diseño con el cual se pueda describir a la misma (por ese tiempo además del modelo HDM, otros modelos y métodos como OOHDM, RMM ya eran reconocidos en la comunidad de Hipermedia). Una vez descrita la aplicación en base a un modelo, los autores le aplican varios criterios de evaluación o heurísticas, como por ejemplo:

- Riqueza, que representa la abundancia de ítems de información y el modo de alcanzarlos.
- Facilidad, que evalúa la accesibilidad a la información y cuán fácil de operar y comprender son las operaciones
- Consistencia, que podría resumirse en (según los autores) elementos conceptualmente similares (dentro de la aplicación) son tratados de modo similar y elementos conceptualmente diferentes son tratados diferentemente.
- Auto-evidencia, representa cuán bien los usuarios se arman la idea mental del propósito de lo que se les está siendo presentado
- Predictibilidad, expresa cuán bien los usuarios se anticipan al resultado de una operación
- Legibilidad, expresa un sentimiento de la validez de toda la aplicación.
- Reuso, considera el uso de objetos y operaciones en diferentes contextos y propósitos.

Sin embargo, este enfoque es sólo conveniente cuando el problema de evaluación es más bien simple e intuitivo. En el caso en que se desee expresar a la aplicación con mayor cantidad de factores y atributos de más bajo nivel se vuelve difícil una evaluación justificable y precisa, en donde se dificulta identificar menores diferencias entre valores similares de atributos de sistemas comparativos. Además, el proyecto

SUE, avanzó en líneas de investigación a partir de estas bases; no obstante, hasta el presente no se ha convertido en una metodología reconocida para evaluar sitios.

Sin embargo, este enfoque no proporciona métricas para medir los atributos identificados y se aplica a evaluaciones simples e intuitivas. En el caso que se desee evaluar un producto complejo y con gran cantidad de factores y atributos, se hace difícil una evaluación precisa. Piattini(2003).

2.5.5 WebQeM: (Olsina 2001)

La propuesta, denominada Metodología de Evaluación de Calidad de SitiosWeb (o, en inglés, Web-site Quality Evaluation Method, o, metodología Web QeM), pretende realizar un aporte ingenieril al proponer un enfoque sistemático, disciplinado y cuantitativo que se adecue a la evaluación, comparación y análisis de calidad de sistemas de información centrados en la Web (más o menos complejos).

El principal objetivo de Web QEM es evaluar y determinar el nivel de cumplimiento de los siguientes factores de calidad descritos en el estándar ISO 9126 (ISO/IEC, 2001): Usabilidad, Funcionalidad, Confiabilidad y Eficiencia. (Piattini, 2003).

2.5.6 Zuh Y Gauch: (2000)

Estos investigadores demuestran que la incorporación de métrica de calidad puede mejorar la eficiencia de búsqueda en ambientes web centralizados y distribuidos. La mayoría de los sistemas de la recuperación de datos en la web utilizan algoritmos basados en términos estadísticos donde no se tiene en cuenta la calidad de la información. Como consecuencia muchos documentos recuperados no tienen calidad. El factor crítico en la evaluación de los documentos (páginas web) es la selección del criterio de calidad. En este trabajo se han seleccionado seis métricas: (a) actual: indica la última actualización de la página web; (b) disponibilidad: el número de enlaces rotos contenidos en la página; (c) tasa de información: proporción de información útil contenida en una página web de un dado tamaño; (d) autoridad: la organización que produjo la página web según el Yahoo Internet Life (YIL); (e) popularidad: cuántas páginas web han referenciado una página; (f) cohesividad: el grado en el cual el contenido de la página se centra en un asunto, (Zuh, 2000).

2.5.7 Otros modelos para la web:

El modelo de calidad QUINT2 (Niessink, 2002) también presenta una ampliación de la norma ISO 9126, pensada para valorar la calidad de arquitecturas software, (Scalone 2006).

El modelo Quint2 (QUality in InformationTechnology), desarrollado inicialmente en Van Zeist et al.,1996, es una ampliación de la norma 9126, un súper-conjunto del conjunto de características y sub-características de la ISO 9126. Quint2, (modelo extendido de la ISO de la calidad del software) amplía el estándar con nuevas características, apropiadas para productos web: funcionalidad, fiabilidad, usabilidad, eficiencia, portabilidad y mantenibilidad, y proporciona métricas para medir la calidad. Quint2 define un modelo que cubre las perspectivas de calidad del usuario y del desarrollador para una aplicación web. La perspectiva del usuario enfoca calidades externas de un producto de software ejecutable, y la perspectiva del desarrollador se dirige hacia las calidades internas del producto durante su desarrollo y mantenimiento. Quint2 permite a los usuarios y desarrolladores hablar de distintos tipos de calidad y priorizar distintos tipos de calidad.

El estándar original de la ISO 9126 ha sido ampliado en Quint2 proporcionando características adicionales de la calidad y asociando indicadores que se calculan para cada sub-característica. Estos indicadores permiten que estimemos la calidad de los productos software con exactitud y un grado aceptable de confianza. Los indicadores están asociados a las sub-características de calidad, pero Quint2 no especifica cómo se deben combinar los indicadores para obtener una estimación integrada de una característica de la calidad. La sensibilidad de una característica de la calidad se calcula como la sensibilidad media de sus sub-características, y ésta se determina como la sensibilidad media de los indicadores correspondientes. La sensibilidad de un indicador es 1 si el indicador es mejorado significativamente o 0 de otra manera, (Scalone, 2006).

Quint2 es un modelo jerárquico que fija seis atributos de calidad principales y 32 sub- atributos. Se muestra esta estructura a continuación en la figura 9.

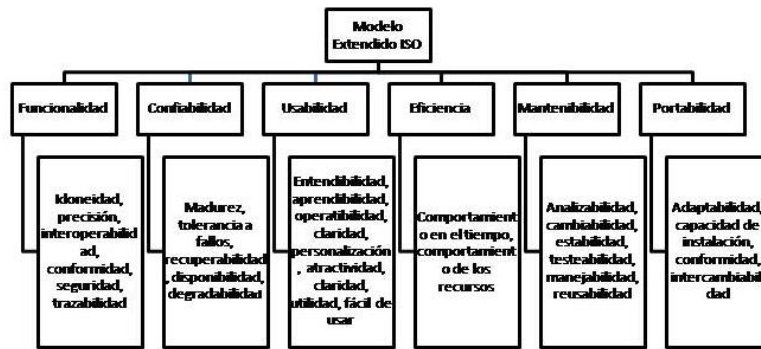


Figura 9. Descripción Jerárquica del Modelo Quint2. Fuente Fernanda Scalone (2006)

Las seis características del modelo Quint2 y las 32 sub-características con sus indicadores correspondientes se describen a continuación:

Funcionalidad: Un conjunto de atributos que tienen que ver con la existencia de un conjunto de funciones y de sus características especificadas. Las funciones son aquellas que satisfacen necesidades indicadas o implicadas.

La funcionalidad trata sobre todo la presencia real de la funcionalidad deseada en el producto de software, exactitud de la implementación, interoperabilidad con los sistemas especificados, etcétera.

Sub-características: conveniencia, exactitud, interoperabilidad, conformidad, seguridad y trazabilidad.

Conveniencia: Atributo del software que tiene que ver con la presencia y la adecuación de un conjunto de funciones para tareas especificadas. La conveniencia es una sub-característica de calidad que considera si las funcionalidades deseadas están presentes en el producto de software. Se determina, sobre todo, por la efectividad de las etapas de la ingeniería de requisitos, cuando un usuario y un desarrollador del software tienen que comunicarse para determinar las funcionalidades deseadas de un producto de software. Debe existir la precisión de la comunicación y la comprensión inequívoca del dominio de la aplicación y del problema que se tratará. Esto tendrá un efecto positivo en el indicador del porcentaje de cobertura (coverage ratio), reduciendo el número de requisitos omitidos o mal entendidos.

Exactitud: Atributo del software que tienen que ver con la corrección de los resultados requeridos del producto. La sub-característica de exactitud estima la corrección de las

funcionalidades y otros resultados requeridos (manuales de usuario etc.) que provienen del producto de software.

Interoperabilidad: Atributo del software que tienen que ver con su capacidad de actuar recíprocamente con sistemas especificados. La interoperabilidad caracteriza la capacidad de un producto de software de actuar recíprocamente con sistemas especificados.

Cumplimiento: Atributo del software que indica la facilidad con que un producto de software se adhiere a estándares y normas.

Seguridad: Atributo del software que tiene que ver con su capacidad para prevenir el acceso no autorizados, accidental o deliberado, a programas o datos.

Trazabilidad: Atributo del software que indica la cantidad de esfuerzo necesitado para verificar la corrección del proceso de datos sobre puntos requeridos.

Fiabilidad: Un conjunto de atributos que refieren la capacidad del software para mantener su nivel de funcionamiento bajo condiciones indicadas por un período de tiempo indicado.

Sub-características: madurez, tolerancia a fallos, recuperabilidad, disponibilidad, y degradabilidad.

Madurez: Atributo del software que tiene que ver con la frecuencia de fracaso por fallos en el software.

Tolerancia a Fallos: Atributos del software que refiere su capacidad de mantener un nivel especificado de funcionamiento en los casos de fallo del software o de las infracciones de su interfaz especificado.

Recuperabilidad: La sub-característica de recuperabilidad tiene que ver con la capacidad de reestablecer su nivel de funcionamiento y recuperar los datos directamente afectados en caso de un fallo y durante el tiempo y el esfuerzo necesario para ello.

Disponibilidad: Atributo del software que refiere la cantidad de tiempo que el producto está disponible para el usuario en el tiempo en que es necesario.

Degradabilidad: Atributo del software que tiene que ver el esfuerzo necesitado para reestablecer la funcionalidad esencial después de una interrupción.

Usabilidad: Un conjunto de atributos que refieren el esfuerzo necesario para el uso de una aplicación, así como la evaluación individual de tal uso, por un determinado conjunto de usuarios.

Debido a la noción sumamente informal de usabilidad, se hace difícil capturar y expresar conceptos relacionados con la usabilidad y el conocimiento. Por otra parte, la disponibilidad de un dominio de la teoría mejora la comunicación entre el usuario y desarrolladores de aplicación y permite el empleo constante de conceptos familiares a los usuarios.

Sub-características: Entendibilidad, facilidad de aprendizaje, operabilidad, explicitidad, personalización, atractividad, claridad, utilidad y facilidad de uso (user-friendliness).

Entendibilidad: Atributo del software que tiene que ver con el esfuerzo de los usuarios para reconocer el concepto lógico del producto software y su aplicabilidad, está relacionado con el reconocimiento de conceptos y su aplicabilidad. La claridad en los conceptos beneficia significativamente la disponibilidad de un dominio de la teoría que transfiere conceptos y nociones familiares a los usuarios por todas las etapas de desarrollo de software. Esto asegura que la terminología obtenida de los usuarios durante la ingeniería de requerimientos es conservada durante todo el diseño y pasos de implementación.

Aprendibilidad: Sub-característica que refleja el esfuerzo de los usuarios para aprender un producto software (por ejemplo, control, entrada, salida). Mejorando la entendibilidad y claridad de los conceptos empleados en la aplicación, se hace mucho más fácil aprender cómo controlar la aplicación y qué tipo de entrada y salida produce o requiere. Esto mejora el esfuerzo requerido para aprender una operación.

Operabilidad: Sub-característica de calidad que indica el esfuerzo de los usuarios para el control de operación y la operación.

Explicitidad: Atributo del software que tiene que ver con el estado del producto software (barras de progresión, etc.).

Personalización: Sub-característica de calidad que permite al software ser personalizado por el usuario para reducir el esfuerzo requerido para su uso, e incrementar el nivel de satisfacción con el software.

Atractividad: Atributo del software que tiene que ver con la satisfacción de los deseos y preferencias del usuario, a través de servicios, comportamiento y presentación más allá de la demanda real.

Claridad: Sub-característica de la calidad del software que tienen que ver con la claridad de hacer al usuario consciente de las funciones que el producto software puede realizar.

Utilidad: (Helpfulness). Atributo del software que tienen que indica la disponibilidad de instrucciones para el usuario sobre cómo interactuar con él.

Facilidad de uso: (User-friendliness). Sub-característica de calidad que indica la satisfacción de los usuarios.

Eficacia: Un conjunto de atributos que refieren la relación entre el nivel de funcionamiento del software y la cantidad de recursos usados, bajo unas condiciones indicadas. Sub-características: Comportamiento del tiempo, comportamiento de los recursos. (*Time behaviour, resource behaviour*).

Comportamiento del tiempo: Atributo del software que se refiere a los tiempos de respuesta y de procesamiento, y a las tasas de rendimiento en la realización de su función.

Comportamiento de los recursos:(Resource behaviour). Sub-característica que indica la cantidad de recursos usados y de la duración de ese uso.

Mantenibilidad: Un conjunto de atributos que refieren el esfuerzo necesitado para realizar modificaciones especificadas.

Sub-características: Analizabilidad, cambiabilidad, estabilidad, facilidad de prueba, flexibilidad y reusabilidad.

Analizabilidad: Atributo del software que refiere el esfuerzo necesitado para el diagnóstico de deficiencias o de causas de fallos, o para la identificación de partes a ser modificadas.

Cambiabilidad: Atributo del software que refiere el esfuerzo necesitado para una modificación, eliminación de fallo o para un cambio en el entorno.

Estabilidad: Sub-característica de la calidad del software que indica el riesgo del efecto inesperado de modificaciones.

Facilidad de prueba: Atributo del software que refiere el esfuerzo necesitado para validar el software (modificado).

Flexibilidad: Atributo del software que refiere el esfuerzo necesitado para restablecer su estado de ejecución.

Reusabilidad: Atributo del software que refiere su potencial para la reutilización completa o parcial en otro producto de software.

Portabilidad: Un conjunto de atributos que refieren la capacidad del software de ser transferido a partir de un entorno a otro. Sub-características: adaptabilidad, installability, conformidad, reemplazabilidad.

Adaptabilidad: Atributo del software que tienen que ver con la oportunidad para su adaptación a entornos diferentes especificados sin aplicar otras acciones o medios que aquellos proveídos por esta razón para el software en cuestión.

Instalabilidad: Atributo de software que refiere el esfuerzo necesitado para instalar el software en un entorno especificado.

Conformidad: Atributo del software que hacen el software adherirse a normas o convenciones que se relacionan con la portabilidad.

Reemplazabilidad: Atributo del software que tienen que ver con la oportunidad y el esfuerzo de su utilización en lugar de otro software especificado en el entorno de este software.

Todas las definiciones anteriores del modelo Quint2 fueron tomadas de (Esparza, 2006), se le dio especial énfasis a éste modelo porque se basa en la ISO 9126 y está enfocado hacia la arquitectura del software.

El modelo de calidad propuesto por **Franch and Carvalho (2003)** presenta una adaptación de la ISO9126 para correo electrónico, (Esparza, 2006).

En **Zo and Ramamurthy (2002)** los autores presentan un modelo para valorar y seleccionar los sitios Web de comercio electrónico en un entorno B2C (Business-to-Consumer), (Esparza, 2006).

En **Webb and Webb (2002)** se presentan los factores de calidad del sitio Web que son importantes para los consumidores, (Esparza, 2006).

En **Parasuraman et al (1998)** se describe el modelo **SERVQUAL** el cual contiene 5 dimensiones y 22 ítems para medir los diferentes elementos de la calidad de un servicio en general. La idea de este modelo es que puede ser adaptado a diferentes entornos en función de los servicios ofrecidos por cada uno de ellos adaptando las dimensiones descritas en el modelo original.

En el caso de **WQM (Web QualityModel)** la gran presencia de tecnología Web y la gran información asociada a esta tecnología hace imprescindible que los diseños se realicen bajo unos mínimos criterios de calidad, hasta ahora prácticamente inexistentes. Las aplicaciones web desarrolladas sin criterios de calidad tendrán un pobre rendimiento y causarán fallos, por lo que es necesario que los sistemas web sean gestionados y dirigidos de forma rigurosa y cualitativa.

WQM está caracterizado por tres elementos: (1) La característica de calidad (basada en Quint2 y en la ISO 9126), (2) El proceso del ciclo de vida (basado en la ISO12207) y (3) Características (contenido, presentación y navegación).

Los 3 factores más utilizados para caracterizar un sitio web son: (1) Contenido, (2) Presentación y (3) Navegación.

Este modelo por tanto es tridimensional y considera las siguientes dimensiones ortogonales:

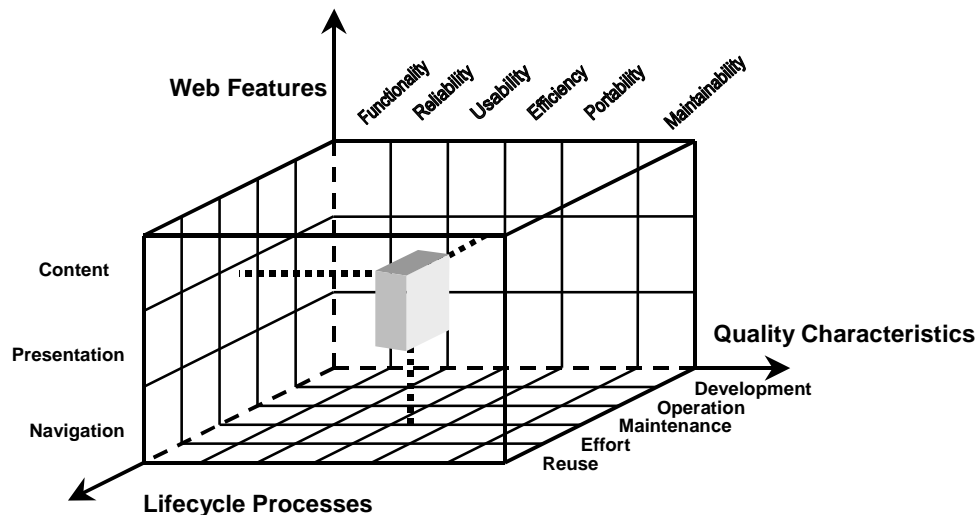


Figura 10. Modelo WQM. Fuente: Piattini (2008)

En total son 385 métricas organizadas en este modelo de acuerdo a las tres dimensiones mencionadas: componentes del sitio web (contenido, presentación, navegación), características de calidad (funcionalidad, fiabilidad, usabilidad, eficiencia, portabilidad, mantenibilidad) y procesos del ciclo de vida (desarrollo, explotación, mantenimiento, esfuerzo y reutilización). Hay más de 60 trabajos entre 1992 y 2004 de revisión de éste modelo (Piattini, 2008) y se puede decir que es uno de los más reconocidos entre los estudiosos de la calidad de software para web.

PQM (Portal Quality Model): tiene como objetivo definir un modelo de calidad para portales, denominado PQM, para lo que se ha utilizado el método GQM. El modelo consta de 6 dimensiones: tangibles, confiabilidad, capacidad de respuesta, aseguramiento, empatía y calidad de los datos. Este modelo está en la etapa de definición y debe ser considerado como una primera aproximación y no como un modelo cerrado y definitivo.

Teniendo en cuenta que la calidad de un portal es difícil tanto de definir como de medir, el modelo debe ser considerado como una primera propuesta de un marco conceptual para medir la calidad de un portal, sabiendo que la calidad del portal la podemos definir como el grado con el que el portal facilita servicios e información relevante al usuario.

2.6 Estado del arte de Modelo de Indicadores para Aplicaciones Web en fase de Diseño

Hasta ahora se han venido esbozando modelos de calidad para el software y algunos modelos de calidad para aplicaciones web a nivel general, el marco teórico se plantea así debido a que no se encontró en la literatura un modelo específico sólo para la fase de diseño; de igual forma que no se encontró un modelo de calidad para aplicaciones web en la fase de diseño. Alguno de los modelos que se mencionarán como el Quint2 hacen referencia como una aproximación a esta investigación a uno de los componentes a tener en cuenta en el diseño de aplicaciones web como lo es la arquitectura del software.

Tenemos como primer hito en el estado del arte de las aplicaciones para web que **Tim Berners-Lee** en octubre de 1994 crea el **World Wide Web Consortium** cuyo objetivo principal es desarrollar protocolos y directrices para la WWW, el siguiente hito alrededor de calidad para aplicaciones web es el generado por el modelo **SUE** en 1997, **Paollini y Gaezotto** emplearon métodos y heurísticas de evaluación cualitativa en el campo de la Hipermedia, luego **Parasuraman et al (1998)** describe el modelo **SERVQUAL** el cual contiene 5 dimensiones y 22 ítems para medir los diferentes elementos de la calidad de un servicio en general.

En el año 2000 **ZUH-GAUCH** demuestran que la incorporación de métricas de calidad pueden mejorar la eficiencia en la búsqueda en ambientes web centralizados y distribuidos. En 2001 **IvoryMelody** establece un **modelo estadístico** que definen qué páginas web son consideradas como buenas y malas esto desde el punto de vista de la usabilidad; en este mismo año **Olsina-Rossi** presenta una **metodología de evaluación de calidad de sitios web**, con ella pretendió realizar un aporte ingenieril que se adecuara a la evaluación, comparación y análisis de calidad de sistemas de información centrados en web.

El año 2002 es el año de mayor productividad para los aportes sobre modelos alrededor de la calidad de aplicaciones web entre ellos se mencionan: **Zo and Ramamurthy** que presentan un **modelo para valorar y seleccionar sitios web de comercio electrónico** en un entorno B2C (Business to Consumer); **Webb and Webb** presentan los factores de calidad del sitio web que son importantes para el consumidor. **Covella y Olsina** hacen una especificación de **atributos de calidad para los sitios con funcionalidad E-Learning**; un aporte alrededor del tema de diseño es propuesto por **Niessink** con el

modelo **QUINT2** que pretende ampliar la norma 9126 pesándola para valorar la calidad de la arquitectura del software.

Franch y Carvallo adaptan la norma 9126 al correo electrónico, en el año 2003. **Calero y Piattini** en el 2005 definen una **clasificación de métricas web** utilizando el modelo de calidad web, para luego en el 2006 presentar la definición de un **modelo de calidad de datos para portales web**. En este mismo año **Sefahh, Donyae, Kline y Padda** desarrollan la **medición de usabilidad y métricas** un modelo consolidado.

En el 2007 las investigaciones o propuestas más importantes se dan con **Moraga, Calero, Piattini y Garzás** con un **modelo de evaluación de calidad de portales**: tomado de una experiencia real; en Argentina desarrollan **Córdoba, Cachero, Calero, Género y Marhuendo** un **modelo de calidad para portales bancarios**.

Así como se estipulan los modelos de calidad para las aplicaciones web también se fueron definiendo con el tiempo modelos de evaluación de las aplicaciones que se destacan en este apartado porque es necesario tenerlos presentes debido a que se pretende luego de organizar el modelo de indicadores llegar hasta la medición de los mismos en los proyectos de grado que se clasifiquen como aplicaciones web del Programa de Ingeniería de Sistemas de la Universidad Popular del Cesar Seccional Aguachica.

Entre los modelos de medición se destacan el **modelo WebQual** en el 2001 cuyo objetivo es la **evaluación de la calidad de sitios web de subastas**. En 2002 Olsina desarrolla un **método cuantitativo para la evaluación de la calidad de aplicaciones y sitios web**.

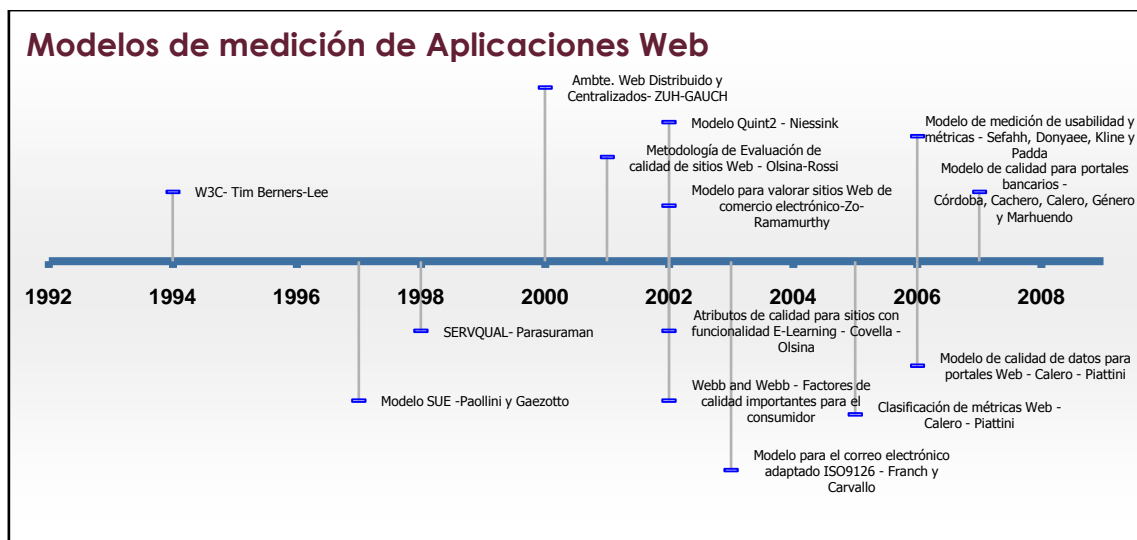
El **modelo 2QCV3Q** es desarrollado en **Mich, Franch y Gaio** y socializado en **2003** y su objetivo es la **evaluación y diseño de calidad de sitios web**. Por último en **2012** se plantea en Argentina un **índice de evaluación web** para medir, evaluar y analizar el grado de cumplimiento de características y criterios de los sitios web bancarios, que operan en la República Argentina, para ser considerado de calidad.

Como se ha podido analizar no se ha mencionado específicamente un modelo que se enfoque a la calidad en la fase de diseño de las aplicaciones web, por ello este primer planteamiento abre el espacio para que para ésta fase como en la de análisis se planteen modelos específicos que aseguren que la calidad se ha propuesto y evaluado, para así

evitar los gastos y pérdidas de tiempo que se generan en el mantenimiento de cualquier software.

A futuro por lo visto en esta descripción histórica de los estudios alrededor de la calidad en las aplicaciones web es que se van delimitando los estudios a casos particulares de calidad en portales bancarios, portales de bibliotecas, en el correo electrónico, en comercio electrónico y así se van enfocando los modelos y las investigaciones objeto de estudio más particulares, reduciendo el ámbito de los modelos y haciéndolos más específicos.

A continuación se ilustra de forma gráfica en una línea de tiempo los aportes a los modelos de calidad en aplicaciones web:



2.7 Análisis comparativo de los modelos de calidad del producto

Realizando un resumen de los modelos vistos y organizando los factores y las características observamos lo siguiente:

McCall		Boehm	FURPS		DROMEY	
FACTORES	CARACTERÍSTICAS	CARACTERÍSTICAS	FACTORES	CARACTERÍSTICAS	FACTORES	CARACTERÍSTICAS
Operación del	Corrección	Portabilidad	Requerimientos	Funcionalidad	Correctitud	Funcionalidad
	Fiabilidad					Confiabilidad

producto	Eficiencia	Fiabilidad	Funcionales (F)		Internas	Mantenibilidad	
	Integridad			Usabilidad		Eficiencia	
	Facilidad de uso			Confiabilidad		Confiabilidad	
Revisión del producto	Facilidad de mantenimiento	Eficiencia	Requerimientos No Funcionales (URPS)	Confiabilidad	Contextuales	Mantenibilidad	
	Flexibilidad					Reusabilidad	
	Facilidad de prueba	Ingeniería Humana				Desempeño	Portabilidad
Transición del producto	Transportabilidad			Facilidad de soporte			Confiabilidad
	Reusabilidad						Mantenibilidad
	Interoperabilidad	Chequeabilidad				Descriptivas	Reusabilidad
		Comprensibilidad		Portabilidad			
		Modificabilidad		Usabilidad			

Tabla 8 Comparación de los diferentes modelos revisados Parte 1. Fuente los autores

SATC		ISO 9126		SQUAE	WQM	
FACTORES	CARACTERISTICAS	FACTORES	CARACTERISTICAS	FACTORES	FACTORES	CARACTERISTICAS
Calidad de los Requerimientos	<i>Ambigüedad</i>	CALIDAD INTERNA Y EXTERNA	Funcionalidad	Mantenibilidad	CALIDAD INTERNA Y EXTERNA	Funcionalidad
	<i>Integridad</i>		Confiabilidad	Evolutibilidad		Confiabilidad
	<i>Facilidad de entender</i>		Usabilidad	Autodescripción		Usabilidad
	<i>Trazabilidad</i>		Eficiencia	Portabilidad		Eficiencia
Calidad del Producto	<i>Estructura / Arquitectura</i>		Capacidad de mantenimiento			Capacidad de mantenimiento
	<i>Reutilización</i>					
	<i>Facilidad de mantenimiento</i>		Portabilidad			Portabilidad
	<i>Documentación</i>					
	<i>Uso del recurso</i>					
	<i>Cumplimiento de los porcentajes</i>					
Efectividad de la prueba						

Tabla 9 Comparación de los diferentes modelos revisados Parte 2. Fuente los autores

Se han tomado los modelos más representativos y de los que se encuentran mayor información en la literatura sobre los estudios y validaciones realizadas de los mismos.

En el ámbito de aplicaciones web se seleccionó el WQM por ser el más amplio y no estar delimitado para un solo tipo de aplicaciones web.

Una vez comparados los principales modelos de calidad para el producto se analizan las características que más se presentan en la mayoría de ellos:

Característica	McCall	Boehm	FURPS	DROMEY	SATC	ISO 9126	SQUAE 25000	WQM	Persistencia
Corrección	X								1
Fiabilidad	x	X	X	XXX		X	X	X	7
Eficiencia	X	X		X		X		X	5
Integridad	X								1
Facilidad de uso	X								1
Facilidad de mantenimiento	X			XXX	X	X	X	X	6
Flexibilidad	X								1
Facilidad de prueba	X								1
Transportabilidad	X	X	X	XX		X	X	X	7
Reusabilidad	X			XX	X				3
Interoperabilidad	X								1
Ingeniería Humana		X							1
Chequeabilidad		X							1
Comprensibilidad		X							1
Modificabilidad		X							1
Funcionalidad			X	X		X	X	X	5
Usabilidad			X	X		X	X	X	5
Desempeño			X						1
Ambigüedad					X				1
Integridad					X				1
Facilidad de entender					X				1
Trazabilidad					X				1
Estructura / Arquitectura					X				1
Documentación					X				1
Uso del recurso					X				1
Cumplimiento de los porcentajes					X				1
Rendimiento							X		1
Compatibilidad							X		1
Seguridad							X		1
Características presentes	11	7	5	13	10	6	8	6	

Tabla 10 Características que más persisten en los modelos de calidad del producto. Fuente los autores

De la anterior tabla podemos inferir que las características que más persisten en los modelos estudiados en el orden de mayor número de veces que aparecen: **fiabilidad y transportabilidad** siete veces, **facilidad de mantenimiento** seis veces y por último eficiencia, funcionalidad y usabilidad cinco veces. Siendo la fiabilidad y la facilidad de mantenimiento las características que se repiten mayor número de veces en un modelo en concreto, hallando éste número de repitencias tres, en el modelo Dromey, modelo

éste que aparece como una propuesta que permite construir y utilizar un modelo de calidad práctico para evaluar etapas de: determinación de los requerimientos, diseño e implementación.

2.8 Tipo de proyectos sobre el cual se realiza el aporte

Como se viene desarrollando desde el marco teórico los proyectos sobre los cuáles se desarrollará el modelo de indicadores son los que clasifican como aplicaciones web, las aplicaciones web tienen unas connotaciones diferentes a los programas tradicionales y por eso estudiosos del tema sobre desarrollo de software consideran que los modelos de calidad varían y debido a ello encontramos los diversos modelos mencionados anteriormente.

En el caso de la presente propuesta se pretende aplicar o validar el modelo de indicadores propuesto en los proyectos de grado del programa de Ingeniería de Sistemas de la Universidad Popular del Cesar Seccional Aguachica que estén en éste dominio de aplicaciones específicamente en la fase de diseño lo que hace pensar en que se deben seleccionar unos factores, unas características y unas métricas o indicadores que de acuerdo a las necesidades de medición permitan sacar conclusiones sobre las metas trazadas.

Otro tema importante a tener en cuenta son los tipos de aplicaciones web sobre los cuáles se aplicará esta investigación, para ello se tendrá en cuenta la clasificación planteada por Ginie y Murugesan (Piattini, 2003) quienes presentan las siguientes categorías:

Aplicaciones web Informacionales.

Aplicaciones web Interactivas.

Aplicaciones web Transaccionales.

Aplicaciones web de Flujos de Trabajo.

Aplicaciones web de Ambientes de trabajo colaborativos.

Aplicaciones web de Comunidades y mercados en línea.

Aplicaciones web de Portales.

Como se pudo ver en el estado del arte, se han venido desarrollando propuestas acerca de modelos de calidad enfocados a mercados en línea, correo electrónico, portales, informacionales y transaccionales; por ello es importante que se revise qué tipo de proyectos considerados como aplicaciones web se han o se vienen desarrollando en el Programa de Ingeniería de Sistemas de la Universidad Popular del Cesar Seccional Aguachica para así tener una guía de los modelos ya planteados para un tipo específico de aplicaciones web o para tener un marco de referencia que le proyecte una guía al modelo de indicadores a proponer.

Para el caso de la Universidad Popular del Cesar Seccional Aguachica, los proyectos que han sido desarrollados ó están en fase de desarrollo encontramos gracias a un listado entregado por el área de recursos bibliográficos donde reposan todos los proyectos de grado de la Institución que se cuenta con los siguientes proyectos clasificados como aplicaciones web transaccionales.

3 Modelo de Indicadores para la fase de diseño de aplicaciones web

El significado del término **modelo** (del italiano modello) reseña diferentes significados según el diccionario de la Real Academia Española (RAE). Entre ellos podemos destacar que un modelo es un arquetipo o punto de referencia para imitarlo o reproducirlo, o un ejemplar que se debe seguir e imitar por su perfección, o un esquema teórico, generalmente en forma matemática, de un sistema o una realidad compleja que se elabora para facilitar su comprensión y el estudio de su comportamiento.

Esta propuesta no pretende realizar una sugerencia de pasos sucesivos como un método sino proponer un punto de referencia o patrón para que pueda ser reproducido por personal que desee realizar un proceso de medición de aplicaciones para la web en su fase de diseño, obviamente teniendo en cuenta métricas y algunas propuestas encontradas en el marco teórico y estado del arte.

Se pretende medir a través de indicadores los productos o artefactos entregados en la fase de diseño de los proyectos de grado del programa de Ingeniería de Sistemas de la Universidad Popular del Cesar, con base en un modelo que tendrá una estructura que se validará en los proyectos en fase de desarrollo o ya terminados. Esta propuesta puede servir no sólo a los desarrolladores de la Institución sino también a los investigadores del área de calidad del software o desarrolladores que desean conocer cómo están sus entregables de la fase de diseño de una aplicación web, pues las mediciones se han propuesto sobre atributos que fácilmente se encuentran en artefactos de cualquier proyecto de desarrollo para la web.

Los modelos y estándares que se han estudiado hasta ahora se organizan en factores ócritérios, atributos, métricas e indicadores y explican cada uno de los elementos a tener en cuenta en la evaluación de calidad del producto, éstos modelos no se especifican para fases del producto como análisis, diseño, implementación o mantenimiento; por ello es deber del grupo o investigador de calidad realizar la respectiva aplicación del modelo según el caso y la necesidad a todo el producto en general. En este caso, se pretende establecer un modelo específico de medición en la fase de diseño.

Lo primero que se debe definir es sobre qué productos de diseño de aplicaciones para web se realizará el modelo de medición; como se menciona en el marco teórico, *Figura*

6, Pressman (2010) plantea cuatro modelos del diseño: el arquitectónico, el de datos o clases, el de interfaz y el de componentes; estos se tienen en cuenta para aplicaciones a nivel general pero a partir de la aparición de las aplicaciones web se agregan nuevos componentes por la particularidad de las mismas que hacen que los investigadores del tema de evaluación de calidad para la web creen nuevos modelos que tengan en cuenta éstos nuevos componentes. Es así como Pressman (2010) en el mismo documento tiene en cuenta tres componentes más para el diseño de aplicaciones web que son el diseño estético, el diseño de contenidos y el diseño navegacional, ver *figura 8*. Es sobre éstos productos se deberían plantear la medición de los proyectos catalogados como aplicaciones web.

Como se mencionó en un párrafo anterior los modelos estudiados en el marco teórico mantienen la siguiente estructura a nivel general:

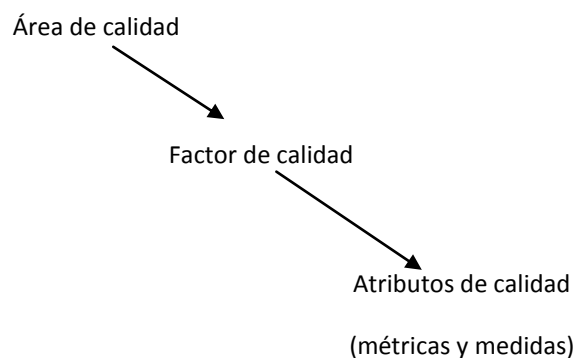


Figura 11. Estructura general de los modelos de calidad. Fuente: Los autores.

La propuesta del modelo de indicadores para la fase de diseño de aplicaciones web deberá llevar una estructura parecida, pues no se pretende ni son los alcances de esta investigación realizar una propuesta de estructura de modelo diferente a las conocidas.

Una de las primeras decisiones que se plantea es que la medición se realice en etapas tempranas del diseño y no cuando éste ya esté terminado y en fase de implementación, pues se pretende que el usuario principal de éste modelo sea el *programador*, es él quien recibe los artefactos de diseño, debe implementarlo y el que más se beneficiaría de un diseño de calidad.

Retomando el análisis realizado en el apartado 2.7 de esta propuesta se observan las características que se presentan más en los diferentes modelos y por ello se van a revisar para mirar de qué manera pueden hacer parte del modelo propuesto. Aquí en este

apartado las definiciones de éstas características se definirán desde el punto de vista del diseño que es nuestra fase de interés donde se realizará la medición.

La definición de **fiabilidad** desde el punto de vista del diseño es, la capacidad del diseño para mantener un nivel especificado de rendimiento cuando se utiliza en las condiciones especificadas. Si analizamos la palabra rendimiento las mediciones de esta característica deberían hacerse una vez la aplicación esté implementada y esta propuesta de modelo de medición espera hacer las mediciones sobre la calidad del diseño en etapas tempranas del desarrollo del software sin tener que esperar a la implementación por ello esta característica no sería de interés para el modelo a proponer.

La definición de **transportabilidad** desde el punto de vista del diseño sería, la capacidad del diseño de ser transferido de un entorno a otro; al pensar en la fase de diseño no se puede definir qué sería pasar el diseño de un entorno a otro en etapas tempranas sin pensar en implementación. Al igual que pasa con la anterior característica se realizarían las mediciones cuando esté el software en funcionamiento y no aplicaría a los objetivos del modelo a proponer.

La **facilidad de mantenimiento o mantenibilidad** desde el punto de vista del diseño, se define como la capacidad del diseño para ser modificado. Las modificaciones pueden incluir correcciones, mejoras o adaptación del software a cambios en el entorno, en los requisitos o en las especificaciones funcionales. Esta característica configura como una característica que se tendría como medición en el modelo a proponer.

Por último **eficiencia, funcionalidad y usabilidad** se definen desde el punto de vista del diseño respectivamente como: la capacidad del diseño para proporcionar el rendimiento apropiado, relativo a la cantidad de recursos utilizados, bajo las especificaciones determinadas; la capacidad del diseño para proporcionar funciones que satisfagan las necesidades especificadas e implícitas cuando el diseño se utiliza en las condiciones especificadas; la capacidad del diseño de ser entendido, aprendido, utilizado y atractivo al desarrollador, cuando se utiliza en las condiciones especificadas. Al igual que el análisis que se realizó anteriormente con las características mencionadas, éstas se medirían en etapas posteriores a la implementación del software, de las tres últimas características la que se puede proponer para medición en el modelo de indicadores para la fase de diseño de aplicaciones web es la de usabilidad.

Hecho este análisis se iniciará la propuesta por considerarse apropiado a las necesidades de medición de la misma el método GQM (Goal, Questions, Metric), donde se parte de un *objetivo claro de qué y porqué se quiere medir* para luego plantear unas preguntas que serán relacionadas con unas métricas de esta manera se construirá el modelo de indicadores para la fase de diseño de aplicaciones web.

Una vez desarrollado se aplicará el modelo en mínimos dos proyectos que tengan las características de aplicaciones web de los proyectos de grado del programa de ingeniería de sistemas y se realizará un análisis de estos resultados.

Se partirá de establecer los objetivos y las preguntas que harán parte de esta medición. A continuación se aplica una plantilla propuesta por Bassili en 1994 y expuesta por Piattini (2009) en su libro Calidad de Sistemas Informáticos para definir los objetivos de medición:

ANALIZAR	el objeto de estudio bajo medición
CON EL PROPÓSITO DE	entender, controlar o mejorar el objeto
CON RESPECTO A	el enfoque de calidad del objeto en el que se centra la medición
DESDE EL PUNTO DE VISTA DE	o perspectiva de las personas que miden el objeto
EN EL CONTEXTO DE	el entorno en el que la medición tiene lugar

Tabla 11 Plantilla de Definición de GQM. Piattini 2009, Calidad de Sistemas Informáticos

Una vez aplicada esta plantilla se llega al siguiente objetivo:

ANALIZAR	Los entregables de la fase de diseño de aplicaciones web
CON EL PROPÓSITO DE	Controlar los componentes de la fase de diseño de aplicaciones web
CON RESPECTO A	La mantenibilidad y usabilidad de las aplicaciones web
DESDE EL PUNTO DE VISTA DE	Los desarrolladores
EN EL CONTEXTO DE	La mejora de los proyectos de grado del programa de ingeniería de sistemas de la Universidad Popular del Cesar Seccional Aguachica

Tabla 12 Plantilla de Definición para modelo de indicadores para la fase de diseño de aplicaciones web. Fuente los autores.

A partir de la definición del objetivo principal de medición se empiezan a plantear preguntas como cuáles son las métricas para medir el objeto asociado a un determinado objetivo. La primera pregunta a considerar para el objetivo antes planteado es: *¿Cuál es el diseño arquitectónico, navegacional, diseño de interfaz y de datos que ayuda a*

mejorar la comprensión para que el programador desarrolle fácilmente (entendibilidad) una aplicación web, no sacrifique la calidad en el diseño y mejore la mantenibilidad?

Bajo la consideración de la anterior pregunta la propuesta se realizará entonces alrededor de la *arquitectura, la complejidad del modelo E-R, el modelo navegacional y la interfaz de usuario.*

Con ello estaríamos centrándonos en el objetivo propuesto y es mejorar mantenibilidad y la entendibilidad para hacerle más sencillo el trabajo de implementación al desarrollador y a los mantenedores de la aplicación, pues cada uno de los productos propuestos son los que entrarán a hacer parte fundamental de la fase que sigue al diseño y ayudarán a simplificarla. Al docente evaluador del proyecto le permitirá en fases tempranas tener claridad sobre los artefactos y cómo deberían estar diseñados para cumplir con estas expectativas de calidad.

El modelo de calidad hasta el momento estaría organizado así:

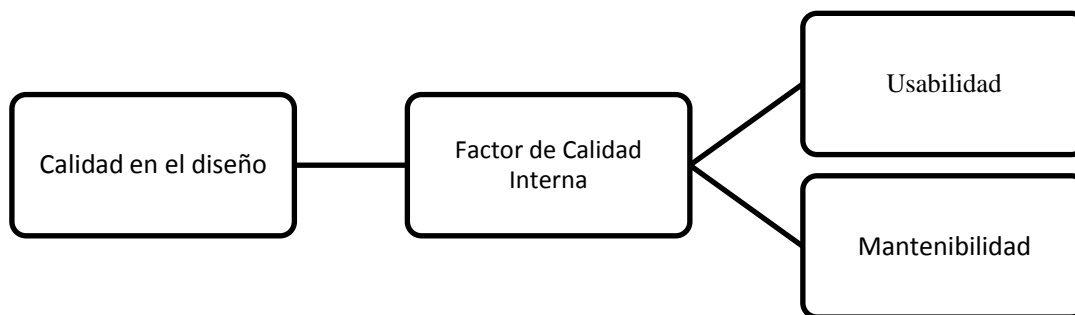


Figura 12. Modelo de Calidad para aplicaciones web en fase de diseño. Área-Factor-Características.

A continuación se presentan los entregables y se definen qué artefactos se medirían y cómo deberían entregarse.

3.1 Artefactos a medir en la fase de diseño de aplicaciones web

En el apartado 2.2.4 de este documento se mencionó las metodologías existentes para el diseño de aplicaciones web y llamó la atención una de ellas por estar enfocada al metalenguaje UML, esta metodología denominada UWE que plantea cuatro fases una que tiene que ver con el diseño conceptual enfocada ésta al diagrama de clases de UML, otra con el diseño navegacional enfocada ésta al modelo navegacional artefacto al que muchos investigadores del tema le dan gran importancia en la aceptación de una aplicación web, el diseño de la presentación o diseño de la interfaz vista desde el punto de vista de diagrama UML y la implementación de la aplicación como tal.

También se presentó una nueva filosofía de desarrollo denominada MDD que se conoce como desarrollo producido por modelos. Se propone por motivos de mejorar la entendibilidad y usabilidad de los diseños para los estudiantes desarrolladores que se realicen con esta orientación los diseños de los proyectos de la Universidad Popular del Cesar Seccional Aguachica, además de algunas ventajas que ya se expusieron en el mismo apartado.

Se solicitará al estudiante trabajar bajo tres modelos que representarían los artefactos de la fase de diseño de las aplicaciones web de la Universidad Popular del Cesar Seccional Aguachica: un modelo estructural, un modelo funcional y el modelo de interfaz de usuario. Estos a su vez se convertirían en la arquitectura de las aplicaciones entregadas por los desarrolladores.

Artefactos deDiseño para las aplicaciones web de la Universidad Popular del Cesar Seccional Aguachica	Modelo estructural
	Modelo funcional
	Modelo de interfaz de usuario

Tabla 13. Artefactos de diseño para una aplicación web para la Universidad Popular del Cesar. Fuente los autores.

3.1.1 Modelo Estructural

Con el modelo estructural se hace referencia al modelo de datos de la aplicación web a desarrollar, en él se deben representar las tablas principales de los datos del que trata el dominio del problema, éstas tablas se representarán gráficamente a través de un cuadro

que deberán contener en la parte superior los nombres que sean significativos a los datos que representan, y que posteriormente muchos de los programadores usan cuando desarrollan su labor; también debe aparecer en el modelo gráfico los atributos de cada entidad y por último las relaciones que pueda existir entre las tablas. Teniendo en cuenta que las relaciones que se representen son 1- 1, 1 – muchos ó viceversa y muchos a muchos. Las claves principales y las foráneas se deben explicitar también en éstos recuadros.

A continuación se presenta con un pequeño ejemplo cómo se deberían representar éstos modelos:

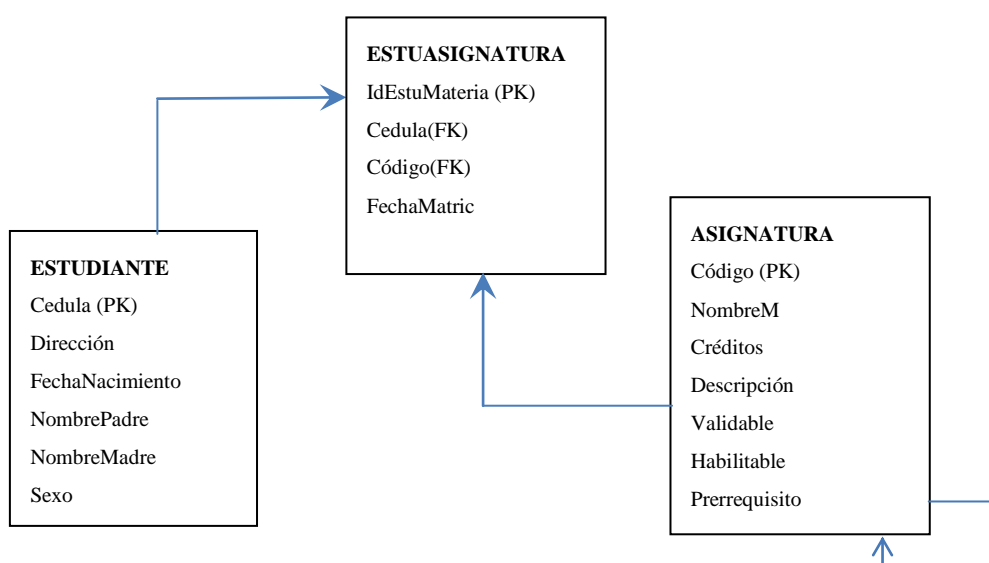


Figura 13. Ejemplo de modelo relacional. Fuente: Los autores.

Aquí se observa un pequeño ejemplo de cómo se graficarían los modelos relacionales de los proyectos de grado de la Universidad. Se observan las tablas, las llaves foráneas y dos relaciones uno a muchos: estudiante –estuassignatura y materia –estuassignatura. También se observa una relación uno – uno en la tabla materia.

Se completa el modelo de igual forma con una plantilla que por razones de practicidad y como un aporte a la documentación del diseño se deberá diligenciar, esto permitiría al evaluador de calidad tomar datos para las mediciones, sin tener que acudir al documento final del proyecto que contiene otro tipo de información que no es relevante para la medición. La plantilla mencionada se encuentra en el Anexo 1.

Igual esta información se debe evaluar en fases tempranas del proyecto de grado, por ello se recomienda hacerlo en el formato II de la asignatura proyecto de grado I.

3.1.2 Modelo Funcional

Cachero (2006) y otros manifiestan que el principal artefacto de la Ingeniería Web relacionado con la navegabilidad es el *modelo navegacional*. Y lo definen como los caminos de navegación que el usuario puede seguir a través del espacio de información con el fin de conseguir sus objetivos. Se encuentra en la bibliografía varias formas de estructurar el *modelo navegacional*, sin embargo, en esta propuesta se presenta un esquema de modelo que ayuda a la presentación de la estructura navegacional y facilita también a nivel funcional el entendimiento de éste tipo de artefactos tan necesarios en el ámbito del desarrollo web de manera muy sencilla.

Teniendo en cuenta que de la fase de análisis se generan los requisitos funcionales del programa que propondrán las funcionalidades del mismo, éstas se tomarán y serán organizadas por módulos; cada funcionalidad tendrá un conjunto de páginas cuya secuencia representará la navegabilidad por funcionalidad. Éste será el modelo propuesto para los diseños de los proyectos de la Universidad Popular del Cesar Seccional Aguachica.

Se documentará en una plantilla de funcionalidad propuesta, los roles y los permisos de acceso por cada rol, en este formato se organizarán los storyboard de la aplicación, donde se listarán las diferentes funcionalidades con las respectivas tablas a usar, gráficos de secuencia de las páginas con una descripción de los enlaces que existen a las otras páginas que ayudará a entender la navegabilidad. La plantilla se puede encontrar en el Anexo B de éste documento.

Los gráficos para representar las secuencias de cada enlace entre páginas se representará de la siguiente forma:

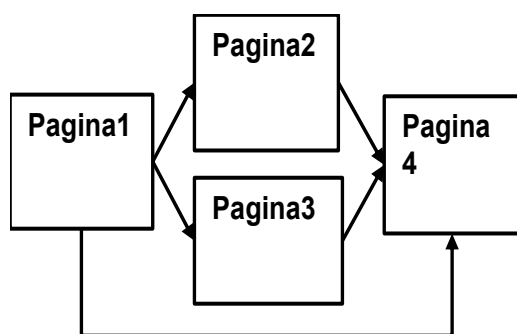


Figura 14. Ejemplo de cómo se representa una funcionalidad de forma gráfica. Navegabilidad.

Y los módulos con sus diferentes funcionalidades quedarán así:

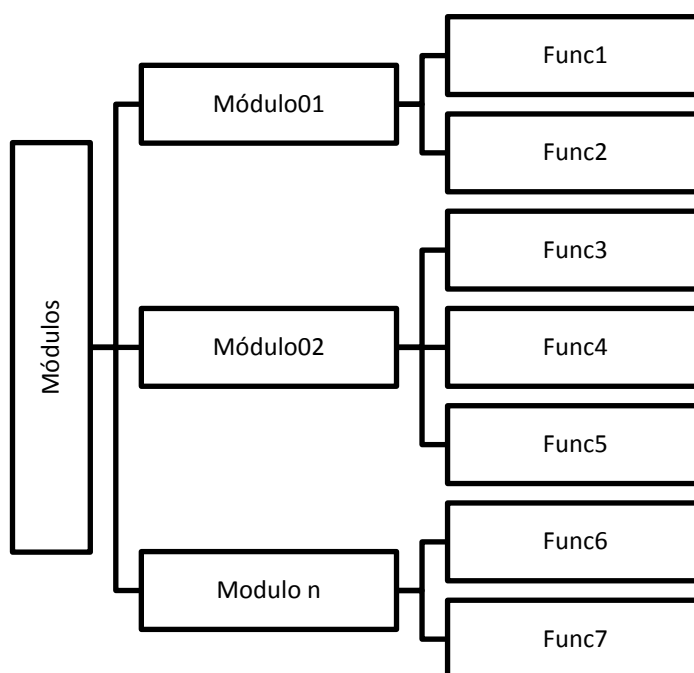


Figura 15. Gráfico del diseño funcional completo para una aplicación web.

Por la sencillez del ejemplo de estudiantes matriculados en asignaturas no se mostrará cómo quedaría en éste apartado un modelo funcional puesto que debería ser una aplicación con algunas funcionalidades para ver el modelo.

3.1.3 Modelo de Interfaz de usuario.

La interfaz de usuario de las aplicaciones web de la Universidad Popular del Cesar serán representadas gráficamente y de manera muy abstracta con un modelo de diseño de páginas que a nivel general puedan mantener la siguiente apariencia: un recuadro donde se dibuje de forma sencilla los contenidos de las páginas como: encabezados, barras de navegación vertical u horizontal, contenidos a mostrar, entre otros.

Se recomienda representar las regiones, preferiblemente con los colores que se pretenden para la implementación. Se propone el uso de plantillas que especifiquen los datos mencionados en los proyectos de grado del programa de ingeniería de sistemas pues se considera suficiente éstos para desarrollar toda la aplicación.

Una interfaz gráfica en esta fase del diseño se representaría así:

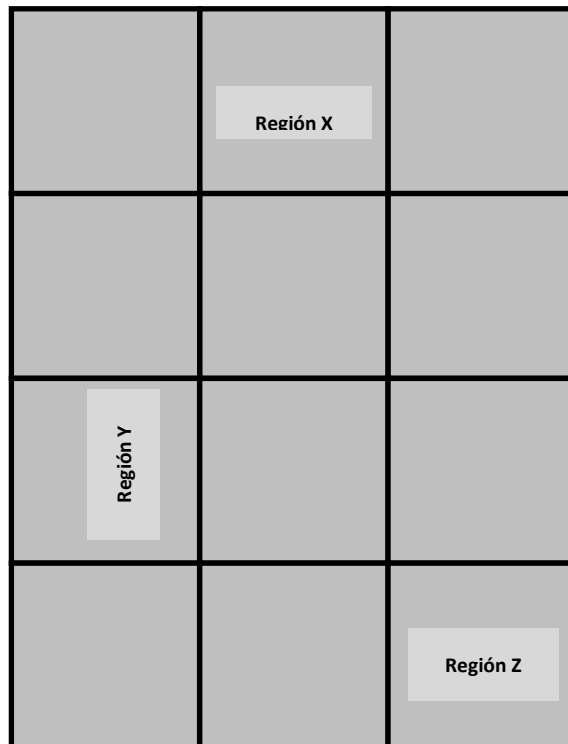


Figura 16. Ejemplo de diseño de interfaz gráfica abstracta para el modelo de aplicaciones web de la Universidad Popular del Cesar Seccional Aguachica. Fuente los autores.

Para éste caso también se propone una plantilla como en los artefactos anteriores, en ella se tendrán en cuenta datos como diseño de la plantilla, cantidad de regiones en la plantilla, conteo de color, conteo de fuente, y uso de guías de estilo que mejoran la apariencia de las páginas y la facilidad en detalles sencillos del mantenimiento de la interfaz gráfica.

El ejemplo de las asignaturas y los estudiantes, si éste fuera el caso de un módulo de matrícula académica, podría quedar en cuanto al diseño gráfico abstracto como sigue:

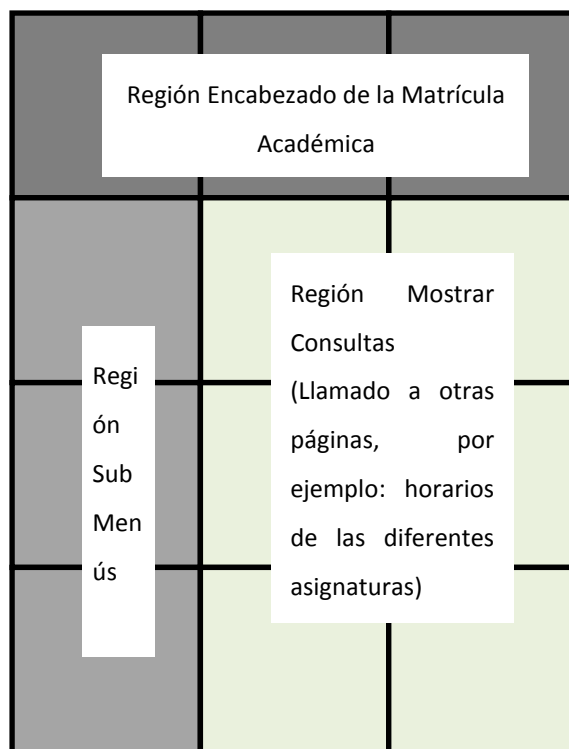


Figura 17. Interfaz gráfica abstracta. Caso ejemplo. Fuente los autores.

3.2 Propuesta atributos, métricas e indicadores y medición del modelo.

Partiendo del objetivo de la medición acordado con la plantilla expuesta en la *tabla 5*, es claro que se tienen dos características que se desean medir en los artefactos antes mencionados: mantenibilidad y entendibilidad desde el punto de vista del desarrollador. La mantenibilidad es ve como una características en varios de los modelos estudiados mientras el concepto de entendible es una sub-característica que se encuentra en la característica de usabilidad en el modelo de calidad para el producto denominado WQM, que es un modelo para las aplicaciones web. Por ello las dos características que se definieron como medibles fueron las de mantenibilidad y usabilidad.

En la siguiente tabla se muestran los modelos estudiados en el marco teórico y la presencia de éstas características seleccionadas.

MODELO O ESTÁNDAR	PRESENCIA DE LA CARACTERÍSTICA
McCall	Facilidad de Uso
	Mantenimiento
Boehm	Ninguna
Furps	Usabilidad
Dromey	Mantenibilidad
	Usabilidad
Stac	Mantenibilidad

ISO 9126	Mantenibilidad
	Usabilidad
SQUAE 25000	Mantenibilidad
WQM	Mantenibilidad
	Usabilidad

Tabla 14. Presencia de las características de Usabilidad y Mantenibilidad en los modelos de calidad del producto estudiados. Fuente: Los autores.

Como se puede observar estas características están una de ellas o las dos en casi todos los modelos exceptuando el de Boehm. Los modelos Droemy, ISO 9126, y WQM miden éstas dos características y por ser los dos últimos de los modelos más actuales se van a tener en cuenta para revisar las sub-características que se medirán en cada uno de los artefactos propuestos para la fase de diseño de aplicaciones web en el inciso anterior.

Para la ISO 9126 se definen las siguientes sub-características o atributos:

Usabilidad	<i>Capacidad de ser entendido</i> <i>Capacidad de ser aprendido</i> Capacidad de ser operado Capacidad de atracción Cumplimiento de usabilidad
Capacidad de mantenimiento	<i>Capacidad de ser analizado</i> <i>Capacidad de ser cambiado</i> Capacidad de ser probado Cumplimiento de la mantenibilidad

Tabla 15. Características de Usabilidad y Mantenibilidad de ISO 9126 con sus atributos. Fuente: Calidad de Sistemas Informáticos de Piattini (2009)..

Los atributos denominados: Capacidad de ser operado, Capacidad de atracción y Cumplimiento de usabilidad, son sensibles de medir cuando la aplicación está en funcionamiento pues la característica de usabilidad se mide cuando la aplicación se ha implementado, sin embargo existen investigaciones que demuestran que es bueno medirla en etapas tempranas del desarrollo, entre estas investigaciones podemos mencionar: Fernández (2010).

El modelo WQM está basado en la 9126 por tal motivo se acogen muchos de sus atributos.

La propuesta de atributos serían las siguientes:

Usabilidad	Entendibilidad vista desde la métrica de legibilidad.
	Comprensibilidad del diseño.

	Facilidad de aprendizaje del diseño.
--	--------------------------------------

Tabla 16. Atributos para la característica de Usabilidad para el modelo de indicadores de las Aplicaciones Web en fase diseño de la Universidad Popular del Cesar Seccional Aguachica. Fuente los autores.

Para la mantenibilidad los atributos seleccionados son:

Mantenibilidad	Capacidad de ser analizado el modelo estructural
	Capacidad de ser cambiado el modelo estructural

Tabla 17. Atributos para la característica de Usabilidad para el modelo de indicadores de las Aplicaciones Web en fase diseño de la Universidad Popular del Cesar Seccional Aguachica. Fuente los autores.

Se dejaron sin medir los atributos: capacidad de ser probado y cumplimiento de la mantenibilidad, pues se considera que no aplican para medir en la fase de diseño pues se medirían mejor cuando ya esté implementada la aplicación web.

Seleccionados los atributos que se consideran medirán las características se procede a seleccionar las métricas, los indicadores y la estructura de medición.

Pasaríamos a organizar entonces la fase de medición y se presentará por artefacto a medir.

3.2.1 Métricas del diseño estructural:

Una vez aclarado lo que se requiere del modelo estructural, se pasa a verificar cómo se medirán las características mantenibilidad y usabilidad a través de los atributos seleccionados. Para éste artefacto se conocen las métricas cuya autora Piattini y otros en el 2001 proponen medir la **complejidad estructural** de los modelos entidad relación con un conjunto de métricas que más adelante se detallan. Tomado de Piattini (2008).

Se afirma que entre **más complejidad** tenga un modelo entidad relación va a ser más complicado éste de usar y de mantener. Además de la propuesta de Piattini y otros se encuentran algunas métricas del conjunto propuesto para medir características como la compleción, integridad, flexibilidad entre otras que se denominan métricas de Moody, sin embargo, éstos no se tienen en cuenta porque miden características diferentes a los atributos seleccionados para la medición que se ha propuesto.

Se tendrán en cuenta las métricas de Piattini y otros y se aplicarán al modelo relacional propuesto, pues éstos modelos tienen muchas similitudes. De igual forma Calero y otros

mencionado por Piattini (2008) hacen una propuesta de métricas para evaluar la mantenibilidad de los modelos relacionales cuyas métricas son similares a las ya mencionadas de Piattini (2001).

Las métricas seleccionadas serán las siguientes:

- Número total de tablas dentro de un modelo relacional.
- Número total de atributos dentro de un modelo relacional.
- Número total de Relaciones en un modelo relacional.

Las métricas de relaciones, son validadas como métricas de complejidad y del tipo de escala de ratio.

Se podrían mencionar más métricas pero éstas son las que se consideran relevantes para la medición que se pretende realizar.

Genero y otros en el 2008 afirmaron que la entendibilidad de los modelos ER se veían afectados por la complejidad estructural que introducen en los modelos ER los atributos y las interrelaciones. Es decir, entre más atributos y relaciones (1:1 y 1:N) tenga el modelo ER más difícil será entenderlo, Piattini (2008). Se aplicarán entonces estos razonamientos al modelo relacional cuando se realice la medición.

Las métricas y los indicadores para medir los atributos serán:

Atributo	Métricas
Capacidad de ser analizado	Número de relaciones
Capacidad de ser cambiado	Número de tablas

Tabla 18. Atributos y métricas para la medición de mantenibilidad del modelo estructural.

La propuesta de métricas para hacer la medición de la usabilidad del modelo relacional es tomada de Berthoa y Vaecillo en su artículo medidas de usabilidad de componentes software, publicación realizada en la revista IEEE para Latinoamérica en el 2006 y será retomada para los otros dos artefactos en su característica de usabilidad.

Los atributos son los siguientes:

Atributo	Métricas
Legibilidad	Porcentaje de tablas entre 10 y 20 caracteres.
Comprensibilidad del diagrama de clases	Promedio de atributos por tabla
Facilidad de aprendizaje del modelo de clases	Porcentaje de tablas adecuadamente descritas

Tabla 19. Atributos y métricas de la medición de usabilidad del modelo estructural

Berthoa en su propuesta de medición de artefactos para diseño en la características de calidad toma la legibilidad como los nombres largos que se le asignen a cualquiera de los objetos que más adelante usarán los desarrolladores y propone un máximo de caracteres para los nombres, para medir la comprensibilidad y la facilidad de aprendizaje se guían por la medición de los tiempos que invierte un desarrollador en comprender y en usar con facilidad y sin equivocaciones el modelo. Aquí se proponen éstas las medidas *promedio de atributos por tabla* y *porcentaje de tablas adecuadamente descritas* por considerarse adecuadas para medir la comprensibilidad y la facilidad de aprendizaje, pues se pretende que con sólo mirar los artefactos el evaluador pueda tener un concepto del modelo sin necesidad de hacer encuestas ni entrevistas a los desarrolladores.

La forma de medir las métricas para éste modelo se explicará más adelante, con su respectiva interpretación.

3.2.2 Métricas del diseño funcional

Una vez clarificado el artefacto denominado modelo funcional se analizará cómo medir la mantenibilidad y usabilidad del mismo.

Cachero (2006) y otros proponen una métrica que denominan número de destinos navegacionales (NDN) a partir de la definición y validación de métricas para modelos

de navegación propuestos por Abrahão y otros. El fin es medir la **complejidad del modelo navegacional** para establecer la calidad de la navegación en lo referente a mantenibilidad. Esta métrica tiene una unidad de medida de tipo escala en la categoría ratio.

Sin embargo la pregunta es qué se puede considerar de calidad o no con esta medida; para ello se acogen a la regla de Miller que manifiesta, según estudios realizados, que para que un modelo no sea ni demasiado trivial ni demasiado complejo el NDN debe estar en el rango entre cinco y nueve (incluyendo éstos valores). Ésta sería la medida para la complejidad del modelo navegacional que tendría relación con la mantenibilidad.

Se tomará de Abrahão (2003) la medida denominada: distancia más larga de un contexto de navegación de la raíz a un contexto hoja, esta propone calcular las distancias más largas que hay desde la raíz del modelo navegacional hasta llegar a las hojas que se encuentran en los contextos del modelo navegacional. Los contextos son las diferentes divisiones que posee un modelo navegacional y hacia donde se bifurca la atención de un usuario de una aplicación web.

Atributo	Métricas
Capacidad de ser analizado	Promedio de distancia más larga de un contexto de navegación de la raíz a un contexto hoja
Capacidad de ser cambiado	Número de nodos

Tabla 20. Atributos y métricas para la medición del modelo funcional

Para la usabilidad se tomarán algunas medidas presentes en la investigación presentada por Berthoa y Valecillo (2006) y se puede analizar en el apartado 3.2.1., sin embargo para la comprensibilidad se tomarán las medidas relativas a los storyboard documentados y la facilidad de aprendizaje se tomará la métrica del número de contextos de navegación. Esta última medida se toma de Cachero (2006).

Las métricas con los atributos de la usabilidad del modelo funcional quedarían así:

Atributo	Métricas
Legibilidad	Porcentaje de nodos con nombres entre 10 y 20 caracteres.
Comprensibilidad del diagrama de clases	Porcentaje de storyboards documentados
Facilidad de aprendizaje del modelo de clases	Número de contextos de navegación

Tabla 21. Atributos y métricas para la medición de la Usabilidad del modelo funcional.

3.2.3 Medición del diseño de Interfaz de Usuario

Las métricas para la mantenibilidad de la interfaz de usuario se definirán a través de plantillas, regiones, conteo de color, conteo de fuentes y guías de estilo. Estas métricas son tomadas desde el planteamiento de Pressman (2010) y algunas recomendaciones encontradas de diseñadores web.

Las métricas que se consideran en este caso para medir la complejidad del diseño de interfaz son las relativas a la posibilidad de cambios en los colores, tipos de fuentes, posibilidades de validación de datos en el cliente y campos que existan en la interfaz que permitan los famosos atajos a los que los usuarios finales están acostumbrados por el sistema operativo Windows, esto afectaría la capacidad de análisis de una interfaz gráfica, si no tenemos bien especificados estos datos en la plantilla ésta es muy compleja de analizar. Por otro lado la posibilidad que se le permita al usuario final cambiar temas y guías de estilo hace más fácil la posibilidad de cambiar el diseño de la interfaz sin que tenga que hacerse un cambio en la aplicación y esto ayudaría a la capacidad de ser cambiada la interfaz gráfica y por ende a la mantenibilidad.

Las métricas que se obtendrían del formato de plantilla del modelo de interfaz gráfica que se planteó quedarían así:

Atributo	Métricas
Capacidad de ser analizado	Porcentaje de plantillas que especifican: conteo de color, conteo de fuente, ayuda en línea, validación de datos en el cliente, campos con atajos.
Capacidad de ser cambiado	Porcentaje de platillas que permiten la definición de temas por parte de cada usuario y la plantilla de estilo

Tabla 22. Métricas, indicadores para medir la mantenibilidad del modelo Interfaz gráfica. Fuente los autores.

La usabilidad del modelo de interfaz gráfica se tomaron algunas métricas encontradas en Pressman (2010), quien plantea la complejidad de la plantilla como el número de regiones que existan en la misma, aunque no especifica cómo sería su medición e interpretación se asume que una cantidad excesiva de regiones en una plantilla haría complejo la legibilidad de la misma y por ende afecta su usabilidad.

La comprensibilidad del modelo se pretende medir por lo bien documentada que se encuentre cada plantilla, así mismo la capacidad de aprendizaje se relacionó con la cantidad de plantillas que proponga el modelo, a mayor cantidad de éstas se disminuye para el desarrollador su capacidad de aprender el diseño de la interfaz gráfica.

Atributo	Métricas
Legibilidad	Promedio de número de regiones por plantillas
Comprensibilidad del diagrama de clases	Porcentaje de plantillas bien documentadas
Facilidad de aprendizaje del modelo de clases	Número de plantillas

Tabla 23. Atributos y métricas para la medición de la Usabilidad del modelo de interfaz gráfica.

3.3 Indicadores del modelo propuesto

Para cada métrica de las documentadas en los apartados anteriores hay diferente tipo de indicadores que ayudan al cálculo de las mismas y que una vez aplicada la fórmula de cálculo se interpretan éstos resultados para lograr conocer qué tan aceptable es el diseño entregado por el estudiante para una aplicación web en la Universidad Popular del Cesar Seccional Aguachica.

La diversidad de propuestas en lo relativo a calcular los diferente tipos de métricas que en su conjunto median atributos distintos y cuyos valores en cuanto a qué se consideraba un buen valor para considerar el modelo como excelente, aceptable o deficiente llevan a realizar la propuesta de un indicador que pudiese medir cada una de las métricas seleccionadas para el modelo de indicadores de aplicaciones web en la fase de diseño sin importar si para un atributo los rangos de valores aceptables fuesen bajos o altos. El indicador que se propone para la medición de las métricas seleccionadas es el siguiente:

Cuando en los valores de interpretación de la métrica el valor más bajo es malo y el más alto es bueno, sería un indicador que se denominará de tipo I y se calculará de la siguiente forma:

$$y = \frac{Min_{aceptable} - X_{valor-real}}{Min_{aceptable} - Max_{aceptable}}$$

Ecuación 1. Indicador para métricas de tipo I. Los autores.

Siendo $Min_{aceptable}$ el valor mínimo aceptable que el evaluador experto considera es el mínimo para ése modelo, $Max_{aceptable}$ el valor máximo aceptable que el evaluador experto considera es el máximo para ése modelo y $X_{valor-real}$ el valor real que se observa en el modelo.

Cuando en los valores de interpretación de la métrica el valor más bajo es considerado bueno y el más alto malo, se utilizará un indicador que se denominará de tipo II y se calculará de la siguiente forma:

$$y = \frac{Max_{aceptable} - X_{valor-real}}{Max_{aceptable} - Min_{aceptable}}$$

Ecuación 2. Indicador para métricas de tipo II. Fuente: Los autores.

La interpretación de los términos que hacen parte de la fórmula son los mismos que se explicaron en la *Ecuación 1*.

Esta fórmula permite normalizar todas las mediciones para realizar los cálculos independientes del tipo de métrica y de sus rangos de aceptación. Una vez normalizados los datos se pasa a la interpretación de los resultados, para ello se plantean unos rangos favorables en cuanto a la exigencia en el ámbito de los estudiantes de pregrado, de igual forma, si se desea ser más exigente con los modelos pues éstos rangos los puede cambiar el evaluador del proyecto. Los rangos de aceptabilidad serían los siguientes:

Rango de valores	Concepto
0 – 0,3	Malo
0,4 – 0,7	Aceptable
0,8 - 1	Excelente

Tabla 24. Rangos de aceptabilidad del modelo de indicadores para aplicaciones web en la fase de diseño. Fuente: Los autores.

Una vez calculados los valores de las métricas y los atributos se busca conocer el valor de aceptabilidad de la característica, para ello se le asignarán pesos a los atributos que indicarán qué tan importante es para el evaluador ése atributo con respecto a los otros cuidando que la suma de los pesos de los atributos para una característica no sea mayor que uno. Esta es la primera agregación que se hace de los atributos de cada característica; luego se deben agregar los valores de las características para sacar la conclusión completa del modelo en cuanto a la calidad interna del producto, también se proponen pesos para que el evaluador le dé la importancia a cada característica dentro del factor y así se pueda conocer el resultado de aceptación de todo el modelo evaluado.

En la siguiente tabla se presenta un ejemplo:

Característica	Peso	Atributo	Peso	Métrica
Usabilidad	0,6	Legibilidad	0,4	% 10-20
		Comprensibilidad	0,3	Prom. atribXtab
		Fac.Aprendizaje	0,3	% tablas adec. descritas
		Total suma atributos	1	
Mantenibilidad	0,4	Capac. De ser analizado	0,6	Nro. Relaciones
		Capac. De ser cambiado	0,4	Nro. tablas
		Total suma atributos	1	
	1	Total suma de pesos de características para el		

Tabla 25. Ejemplo de agregación de pesos a los atributos y características para el cálculo final del factor de calidad interna del producto. Fuente: Los autores.

Para la medición de todo el modelo se diseña una herramienta en Excel que permite hacer los cálculos alimentando los valores máximos, mínimos, el valor real y los pesos de las características y los atributos.

Finalmente el modelo queda de la siguiente manera:

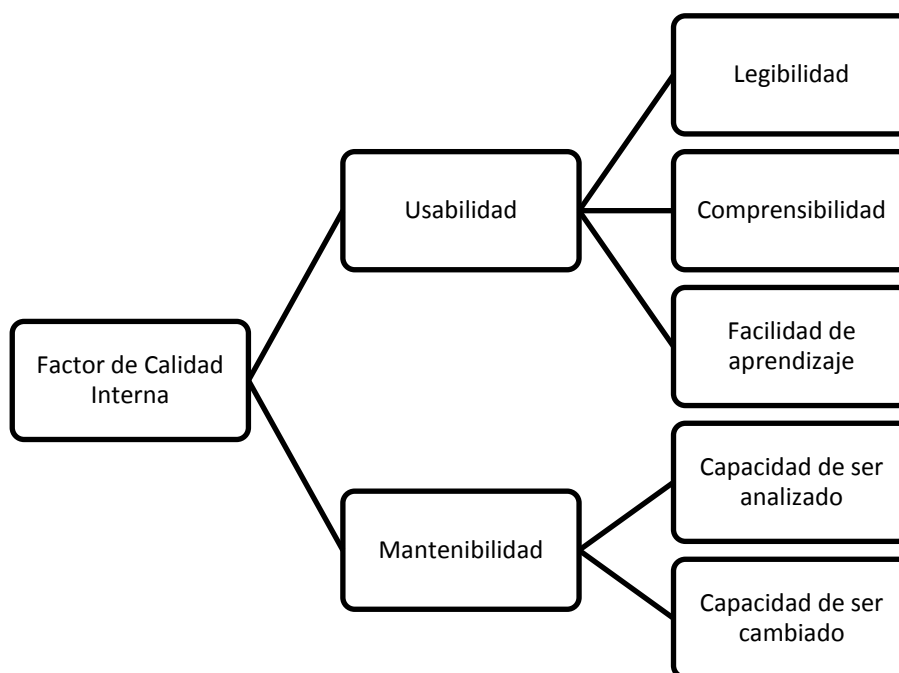


Figura 18. Modelo de calidad interna del producto para aplicaciones web en fase diseño. Características y atributos. Fuente: Los autores.

Cada atributo tiene su métrica de acuerdo al modelo estructural, funcional o de diseño de interfaz, el indicador se aplica de acuerdo a si es tipo I ó tipo II, se proponen unos pesos para hacer agregaciones de acuerdo a la importancia que se dé a los atributos y características que darán un resultado total y finalmente se propuso los rangos de aceptabilidad para conocer el estado de los artefactos a medir.

Para cada modelo se interpretan los indicadores tipo I y tipo II como sigue:

Atributo	Nombre	Definición	Tipo de Indicador
Legibilidad	Porcentaje de tablas con nombres entre 10 y 20 caracteres	Determina el porcentaje de nombres de tablas entre 10 y 20 caracteres	1 Entre más alto el porcentaje es mejor. Lo ideal es que el 100% de los nodos

			tengan nombres entre 10 y 20 caracteres, de lo contrario más alto de 20 ó menor de diez afecta la legibilidad del modelo.
Comprensibilidad	Promedio de atributos por tabla	Determina el promedio de atributos que poseen las tablas	2 Entre más bajo el promedio mejor. Entre atributos posea una tabla mayor dificultad de comprender el modelo y sus relaciones.
Facilidad de aprendizaje	Porcentaje de tablas adecuadamente descritas	Determina el número de tablas que están documentadas de manera adecuada en los manuales.	1 Entre más alto el número es mejor. Entre más tablas documentadas más fácilmente se puede aprender el modelo.

Tabla 26. Atributos, métricas y tipo de indicador usabilidad del modelo estructural.

Atributo	Nombre	Definición	Tipo de Indicador
Capacidad de ser analizado	Número de relaciones	Determina el número de relaciones existentes en el modelo relacional	1 Entre más alto el número es mejor. Entre más relaciones existan más fácil se puede analizar un modelo relacional.
Capacidad de ser cambiado	Número de tablas	Determina el número de tablas existentes en el modelo relacional	2 Entre más bajo el número mejor. Entre menos tablas más fácil de cambiar el modelo relacional sobre todo si hay fuerte cohesión entre tablas.

Tabla 27. Atributos, métricas y tipo de indicador mantenibilidad del modelo estructural.

Atributo	Nombre	Definición	Tipo de Indicador
Legibilidad	Porcentaje de nodos con nombres entre 10 y 20 caracteres	Determina el porcentaje de nombres de nodos de navegación entre 10 y 20 caracteres	1 Entre más alto el porcentaje es mejor. Lo ideal es que el 100% de los nodos tengan nombres entre 10 y 20 caracteres, de lo contrario más alto de 20 ó menor de diez afecta la legibilidad del modelo.
Comprensibilidad	Porcentaje de	Establece el porcentaje	1

	storyboard documentados	de las storyboard que están bien documentadas	Entre más alto el porcentaje mejor. Entre más storyboard documentadas más posibilidad de comprender el modelo.
Facilidad de aprendizaje	Número de contextos de navegación	Cuenta el número de contextos de navegación propuestas en el mapa de navegación	2 Entre más bajo el número es mejor. Entre menos contextos de navegación deba entender más fácilmente se puede aprender el modelo.

Tabla 28. Atributos, métricas y tipo de indicador usabilidad del modelo funcional.

Atributo	Nombre	Definición	Tipo de Indicador
Capacidad de ser analizado	Promedio de distancias más larga de un contexto de navegación de la raíz a un contexto hoja	Determina el promedio aceptable de profundidades que debe existir en un modelo, esto es sacar los promedios de los caminos considerados como largos desde una hoja de un contexto hasta la raíz.	2 Entre más bajo el número es mejor. Si existen caminos de navegación menos largos es más fácil de analizar el modelo.
Capacidad de ser cambiado	Número de nodos.	Establece el número de enlaces que se encuentran en el mapa navegacional. Para ello ya hay un número dado por Miller que expresa que el rango para ésta métrica debe estar entre 5 y 9 nodos.	2 Entre más bajo el número mejor. Entre menos nodos de navegación más fácil de cambiar el modelo.

Tabla 29. Atributos, métricas y tipo de indicador mantenibilidad del modelo funcional.

Atributo	Nombre	Definición	Tipo de Indicador
Legibilidad	Promedio número de regiones por plantilla	Determina el promedio de regiones que hay por las plantillas que se definan en el modelo de interfaz gráfica.	2 Entre más bajo el número es mejor. Entre menos regiones en las plantillas es más legible el modelo.
Comprensibilidad	Porcentaje de plantillas bien documentadas	Establece el porcentaje de las plantillas que están bien documentadas	1 Entre más alto el porcentaje mejor. Entre más plantillas documentadas más posibilidad de comprender el modelo.
Facilidad de aprendizaje	Número de plantillas	Cuenta el número de plantillas propuestas en	2 Entre más bajo el

		el modelo.	número es mejor. Entre menos plantillas deba interpretar más fácilmente se puede aprender el modelo.
--	--	------------	--

Tabla 30. Atributos, métricas y tipo de indicador usabilidad del modelo de interfaz gráfica.

Atributo	Nombre	Definición	Tipo de Indicador
Capacidad de ser analizado	Porcentaje de plantillas que especifican: conteo de color, conteo de fuente, ayuda en línea, validación de datos en el cliente, campos con atajos.	Establece el porcentaje de plantillas que especifican éstos elementos importantes para el diseño	1 Entre más alto el porcentaje mejor. Entre más plantillas especifiquen éstas características más fácil cambiar el diseño.
Capacidad de ser cambiado	Porcentaje de platillas que permiten la definición de temas por parte de cada usuario y la plantilla de estilo?	Determina el porcentaje de plantillas que usan éstas dos utilidades para el diseño.	1 Entre más alto el número es mejor. Si la mayor parte de las plantillas gozan de éstas utilidades es mejor el análisis del modelo.

Tabla 31. Atributos, métricas y tipo de indicador mantenibilidad del modelo de interfaz gráfica.

Finalmente se pasa a la validación del modelo propuesto en al menos dos proyectos de grado de la Universidad Popular del Cesar Seccional Agachica.

3.4 Validación del modelo

Después de haber desarrollado la propuesta del modelo de indicadores para la fase de diseño de aplicaciones web en la Universidad Popular del Cesar Seccional Aguachica y en cumplimiento del tercer objetivo específico de esta tesis, la autora, quien trabaja en la institución, en un tiempo como coordinadora del programa y actualmente como docente, procede a seleccionar dos proyectos del programa de ingeniería de sistemas como se había planteado.

No se eligió la opción de validación por expertos sino la de validación in situ en los proyectos de grado de la Institución debido a las condiciones de la autora dentro de ella, pues se pretende mejorar la fase de diseño de las aplicaciones web generadas en los proyectos de grado de la Institución. Los datos tomados son recopilados de información entregada por funcionarios de la Universidad y la validación real y pertinente del modelo dentro del campo de acción es responsabilidad de la autora.

Hecha una revisión del listado de proyectos ya terminados se encuentra que muchos de ellos no poseen los artefactos con las condiciones que se han planteado para los modelos por ello se harán algunas reconstrucciones del material que se encuentra en los proyectos para diligenciar en primer lugar las plantillas propuestas y de allí proceder a evaluarlas.

Dentro del listado de proyectos de grado de la Universidad Popular del Cesar Seccional Aguachica que configuran como aplicaciones web se seleccionan los siguientes pues se consideran de especial interés para el programa:

Estudio para conocer el estado actual del uso y apropiación de las tecnologías de la información y las comunicaciones (tic) en el municipio de aguachica-cesar, como fase de inicio del proyecto “Aguachica Digital”. Anyi Zuley Pallares Guillin, Deisy Chinchilla Machado, Tatiana Meneses Quintero.

Diseño e implementación de un sistema de información en ambiente web para el proceso de ventas de la empresa Ambiflores localizada en el municipio de Aguachica, Cesar. Lina Paola Barrios y Karina Vergel Rangel.

Diseño e implementación de una plataforma para soportar una comunidad virtual académica en entorno web que gestione y apoye los procesos de interacción de conocimiento en la Universidad Popular del Cesar, Seccional Aguachica. Karen Lorena LeonPerez, Ana Maria Rojas Sánchez.

3.4.1 Caso Estudio Proyecto 1:

Estudio para conocer el estado actual del uso y apropiación de las tecnologías de la información y las comunicaciones (tic) en el municipio de Aguachica-cesar, como fase de inicio del proyecto “Aguachica Digital”. AnyiZuley Pallares Guillin, Deisy Chinchilla Machado, Tatiana Meneses Quintero.

Resumen: Esta trabajo surge como una necesidad que tenía la Universidad Popular del Cesar Seccional Aguachica de conocer el estado actual de uso de las Tecnologías de la Información y las Comunicaciones TIC en el Municipio de Aguachica Cesar; y en la administración y actualización eficaz de los datos obtenidos mediante esta investigación, contribuyendo así al desarrollo de la fase de inicio del proyecto “Aguachica Digital” que se tiene contemplado en el Plan Educativo del Programa de Ingeniería de Sistemas de la institución. Para ello se implementa una herramienta tecnológica enfocada a la Web; esta aplicación es capaz de administrar de manera sencilla, organizada y eficiente los datos obtenidos del estudio TIC, además, permite que las empresas puedan actualizar su información en tiempo real vía web, por otra parte, las entidades gubernamentales, o personas encargadas de gestionar proyectos en beneficio del municipio se pueden vincular al proyecto por medio del aplicativo como sub-administradores de un determinado sector y plantear estrategias de solución que contrarresten las falencias encontradas en cada uno de ellos; permitiendo así que el municipio de Aguachica avance en el uso y apropiación de las tecnologías de la información y las comunicaciones (TIC).

Característica	Peso	Atributo	Peso	Métricas	Valor máximo	Valor mínimo	Tipo de valor	Valor obtenido	Valor normalizado	Concepto	Valor total Usabilidad	Concepto Usabilidad	Valor C.I.	Concepto C.I.
Usabilidad	0,5	Legibilidad	0,4	Porcentaje de tablas con nombres entre 10 y 20 caracteres	100	20	1	60	0,50	Aceptable	0,80	Excelente	0,43	Aceptable
		Comprensibilidad	0,4	Promedio de atributos por tabla	10	3	2	3	1,00	Excelente				
		Facilidad de aprendizaje	0,2	Porcentaje de tablas adecuadamente descritas	100	10	1	100	1,00	Excelente				
1														
Característica	Peso	Atributo	Peso	Métricas	Valor máximo	Valor mínimo	Tipo de valor	Valor obtenido	Valor normalizado	Concepto	Valor total Usabilidad	Concepto Usabilidad	Valor C.I.	Concepto C.I.
Mantenibilidad	0,5	Capacidad de ser analizado	0,7	Numero de relaciones	64	32	1	35	0,09	Deficiente	0,07	Deficiente	0,43	Aceptable
		Capacidad de ser cambiado	0,3	Numero de tablas	25	10	2	32	0,00	Deficiente				

Tabla 32. Evaluación de la calidad interna Modelo Estructural. Caso estudio: Proyecto 1.

Característica	Peso	Atributo	Peso	Métricas	Valor máximo	Valor mínimo	Tipo de valor	Valor obtenido	Valor normalizado	Concepto	Valor total Usabilidad	Concepto Usabilidad	Valor C.I.	Concepto C.I.
Usabilidad	0,7	Legibilidad	0,4	Porcentaje de nodos con nombres entre 10 y 20 caracteres	100	50	1	0	0,00	Deficiente	0,00	Deficiente	0	Deficiente
		Comprensibilidad	0,4	Porcentaje de storyboard documentados	100	50	1	0	0,00	Deficiente				

		Facilidad de aprendizaje	0,2	Número de contextos de navegación	20	10	2	0	0,00	Deficiente		
1												
Característica	Peso	Atributo	Peso	Métricas	Valor máximo	Valor mínimo	Tipo de valor	Valor obtenido	Valor normalizado	Concepto	Valor total Usabilidad	Concepto Usabilidad
Mantenibilidad	0,3	Capacidad de ser analizado	0,5	Promedio de distancias más larga de un contexto de navegación de la raíz a un contexto hoja	8	4	2	0	0,00	Deficiente	0,00	Deficiente
		Capacidad de ser cambiado	0,5	Número de nodos	9	5	2	0	0,00	Deficiente		

Tabla 33. Evaluación de la calidad interna Modelo Funcional. Caso estudio: Proyecto 1.

Característica	Peso	Atributo	Peso	Métricas	V _{max}	V _{min}	Tipo de valor	Valor obtenido	Valor normalizado	Concepto	Valor total Usabilidad	Concepto Usabilidad
Usabilidad	0,6	Legibilidad	0,4	Promedio de número de regiones por plantillas	3	2	2	2	1,00	Excelente	0,50	Aceptable
		Comprensibilidad	0,4	Porcentaje de plantillas bien documentadas	100	50	1	0	0,00	Deficiente		
		Facilidad de aprendizaje	0,2	Número de plantillas	3	1	1	2	0,50	Aceptable		
Característica	Peso	Atributo	Peso	Métricas	Valor máximo	Valor mínimo	Tipo de valor	Valor obtenido	Valor normalizado	Concepto	Valor total Usabilidad	Concepto Usabilidad

Mantenibilidad	0,6	Capacidad de ser analizado	0,5	Porcentaje de plantillas que especifican: conteo de color, conteo de fuente, ayuda en línea, validación de datos en el cliente, campos con atajos.	100	50	1	0	0,00	Deficiente	0,0	Deficiente		
		Capacidad de ser cambiado	0,5	Porcentaje de platillas que permiten la definición de temas por parte de cada usuario y la plantilla de estilo	100	20	1	0	0,00	Deficiente				

Tabla 34. Evaluación de la calidad interna Modelo Interfaz Gráfica. Caso estudio: Proyecto 1.

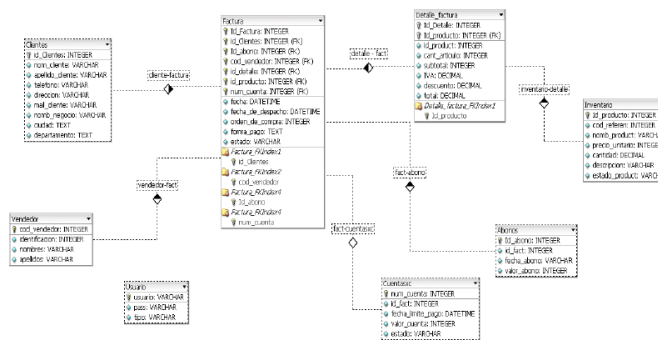
3.4.2 Caso Estudio Proyecto 2:

Diseño e implementación de un sistema de información en ambiente web para el proceso de ventas de la empresa Ambiflores localizada en el municipio de Aguachica, Cesar. Lina Paola Barrios y Karina Vergel Rangel.

Resumen: Este trabajo de grado se desarrolla en la empresa Ambiflores, localizada en la ciudad de Aguachica, Departamento del Cesar, dedicada a la fabricación y venta de productos para el aseo del hogar, partiendo de la necesidad que presentaba la entidad, se desarrolló una herramienta informática bajo plataforma web que le brinda la entidad la sistematización de los procesos del área de ventas (inventarios, facturación, clientes, cartera) que se efectuaban de manera manual y se registraban en hojas de cálculo de Excel, almacenando un inmenso volumen de información que era compleja de administrar.

3.4.2.1 Medición:

El modelo estructural que se encuentra en el proyecto de grado antes mencionado se muestra en la siguiente figura, a partir de éste se iniciará el conteo de las métricas que se puedan obtener para la medición.



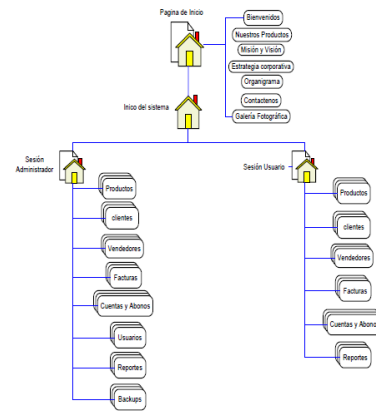
Fuente: Lina Paola Barrios Moreno, Karina Vergel Rangel

Figura 20. Modelo Estructural Caso Estudio Proyecto2. Fuente: Proyecto de grado. Barrios (2012).

Este modelo consta de ocho (8) tablas todas con sus respectivas llaves principales y llaves foráneas.

El modelo funcional que presentan en el documento dista del modelo propuesto por éste proyecto dista de la propuesta realizada en éste documento y es complejo realizar o tomar los datos necesarios para que se puedan realizar los cálculos por ello muchos datos se alimentan como cero en la plantilla de Excel. A continuación se muestra la figura presentada como diseño navegacional:

Figura 13. Mapa de navegación del sistema



Fuente: Lina Paola Baniño Moreno, Karina Vergel Rangel
99

Figura 21. Diseño Navegacional presentado Caso estudio Proyecto2. Proyecto de grado Barrios (2012).

Con el diseño de la interfaz donde no es claro qué tipo de plantillas y cuántas regiones irían a utilizar, sin embargo por los pantallazos finales se asumen las plantillas que usaron y las regiones que existen en éstas plantillas.

Los resultados luego de actualizar los datos en la plantilla de Excel son los siguientes:

Característica	Peso	Atributo	Peso	Métricas	Valor maximo	Valor minimo	Tipo de valor	Valor obtenido	Valor normalizado	Concepto	Valor total Usabilidad	Concepto Usabilidad	Valor C.I.	Concepto C.I.
Usabilidad	0,5	Legibilidad	0,4	Porcentaje de tablas con nombres entre 10 y 20 caracteres	100	20	1	30	0,13	Deficiente	0,21	Deficiente	0,28	Deficiente
		Comprensibilidad	0,4	Promedio de atributos por tabla	8	3	2	6	0,40	Aceptable				

		Facilidad de aprendizaje	0,2	Porcentaje de tablas adecuadamente descritas	100	10	1	0	0,00	Deficiente			
1													
Característica	Peso	Atributo	Peso	Métricas	Valor máximo	Valor mínimo	Tipo de valor	Valor obtenido	Valor normalizado	Concepto	Valor total Usabilidad	Concepto Usabilidad	
Mantenibilidad	0,5	Capacidad de ser analizado	0,7	Numero de relaciones	10	5	1	6	0,20	Deficiente	0,34	Aceptable	
		Capacidad de ser cambiado	0,3	Numero de tablas	10	7	2	8	0,67	Aceptable			

Tabla 35. Evaluación de la calidad interna Modelo Estructural. Caso estudio: Proyecto 2.

Característica	Peso	Atributo	Peso	Métricas	Valor máximo	Valor mínimo	Tipo de valor	Valor obtenido	Valor normalizado	Concepto	Valor total Usabilidad	Concepto Usabilidad	Valor C.I.	Concepto C.I.
Usabilidad	0,7	Legibilidad	0,4	Porcentaje de nodos con nombres entre 10 y 20 caracteres	100	50	1	0	0,00	Deficiente	0,20	Deficiente	0,14	Deficiente
		Comprensibilidad	0,4	Porcentaje de storyboard documentados	100	50	1	0	0,00	Deficiente				
		Facilidad de aprendizaje	0,2	Número de contextos de navegación	3	2	2	2	1,00	Excelente				
1														
Característica	Peso	Atributo	Peso	Métricas	Valor máximo	Valor mínimo	Tipo de valor	Valor obtenido	Valor normalizado	Concepto	Valor total Usabilidad	Concepto Usabilidad		
Mantenibilidad	0,3	Capacidad de ser analizado	0,5	Promedio de distancias más larga de un contexto de navegación de la raíz a un contexto hoja	8	4	2	0	0,00	Deficiente	0,00	Deficiente		

		Capacidad de ser cambiado	0,5	Número de nodos	9	5	2	0	0,00	Deficiente				
--	--	---------------------------	-----	-----------------	---	---	---	---	------	------------	--	--	--	--

Tabla 36. Evaluación de la calidad interna Modelo Funcional. Caso estudio: Proyecto 2.

Característica	Peso	Atributo	Peso	Métricas	Valor máximo	Valor mínimo	Tipo de valor	Valor obtenido	Valor normalizado	Concepto	Valor total Usabilidad	Concepto Usabilidad	Valor C.I.	Concepto C.I.
Usabilidad	0,6	Legibilidad	0,4	Promedio de número de regiones por plantillas	4	2	2	3	0,50	Aceptable	0,20	Deficiente	0,12	Deficiente
		Comprensibilidad	0,4	Porcentaje de plantillas bien documentadas	100	50	1	0	0,00	Deficiente				
		Facilidad de aprendizaje	0,2	Número de plantillas	3	1	1	1	0,00	Deficiente				
Característica	Peso	Atributo	Peso	Métricas	Valor máximo	Valor mínimo	Tipo de valor	Valor obtenido	Valor normalizado	Concepto	Valor total Usabilidad	Concepto Usabilidad	Valor C.I.	Concepto C.I.
Mantenibilidad	0,6	Capacidad de ser analizado	0,5	Porcentaje de plantillas que especifican: conteo de color, conteo de fuente, ayuda en línea, validación de datos en el cliente, campos con atajos.	100	50	1	0	0,00	Deficiente	0,0	Deficiente	0,12	Deficiente
		Capacidad de ser cambiado	0,5	Porcentaje de platillas que permiten la definición de temas por parte de cada usuario y la plantilla de estilo	100	20	1	0	0,00	Deficiente				

Tabla 37. Evaluación de la calidad interna Modelo Interfaz Gráfica. Caso estudio: Proyecto 2.

3.4.3 Caso Estudio Proyecto 3:

Diseño e implementación de una plataforma para soportar una comunidad virtual académica en entorno web que gestione y apoye los procesos de interacción de conocimiento en la Universidad Popular del Cesar, Seccional Aguachica. Karen Lorena LeonPerez, Ana Maria Rojas Sánchez.

Resumen: La Comunidad Virtual Académica COVA es un aplicativo web propio de la Universidad Popular del Cesar, Seccional-Aguachica donde los estudiantes podrán plantear preguntas académicas sin ninguna limitación física o de espacio sobre cualquier tema, con el fin de obtener respuesta, logrando así compartir sus conocimientos, opiniones y experiencias adquiridas. En este espacio los usuarios podrán interactuar, relacionarse, ofrecer y compartir conocimientos o dudas académicas, logrando que la información fluya de una manera más rápida veraz y espontánea, generando así cultura de participación, aceptación de la diversidad de información y voluntad de compartir entre los mismos estudiantes; eliminando así las barreras comunicativas aún a pesar de las barreras estructurales y por lo tanto entablar nuevos grupos de trabajo enfocados a la investigación.

3.4.3.1 Medición:

El modelo estructural que se encuentra en el proyecto de grado antes mencionado se muestra en la siguiente figura, a partir de éste se iniciará el conteo de las métricas que se puedan obtener para la medición.

Característica	Peso	Atributo	Peso	Métricas	Valor máximo	Valor mínimo	Tipo de valor	Valor obtenido	Valor normalizado	Concepto	Valor total Usabilidad	Concepto Usabilidad	Valor C.I.	Concepto C.I.
Usabilidad	0,5	Legibilidad	0,4	Porcentaje de tablas con nombres entre 10 y 20 caracteres	100	20	1	30	0,13	Deficiente	0,21	Deficiente	0,28	Deficiente
		Comprensibilidad	0,4	Promedio de atributos por tabla	8	3	2	6	0,40	Aceptable				
		Facilidad de aprendizaje	0,2	Porcentaje de tablas adecuadamente descritas	100	10	1	0	0,00	Deficiente				
1														
Mantenibilidad	0,5	Capacidad de ser analizado	0,7	Numero de relaciones	10	5	1	6	0,20	Deficiente	0,34	Aceptable	0,28	Deficiente
		Capacidad de ser cambiado	0,3	Numero de tablas	10	7	2	8	0,67	Aceptable				

Tabla 38. Evaluación de la calidad interna Modelo Estructural. Caso estudio: Proyecto3.

Característica	Peso	Atributo	Peso	Métricas	Valor máximo	Valor mínimo	Tipo de valor	Valor obtenido	Valor normalizado	Concepto	Valor total Usabilidad	Concepto Usabilidad	Valor C.I.	Concepto C.I.
Usabilidad	0,7	Legibilidad	0,4	Porcentaje de nodos con nombres entre 10 y 20 caracteres	100	50	1	0	0,00	Deficiente	0,20	Deficiente	0,14	Deficiente
		Comprensibilidad	0,4	Porcentaje de storyboard documentados	100	50	1	0	0,00	Deficiente				
		Facilidad de aprendizaje	0,2	Número de contextos de navegación	3	2	2	2	1,00	Excelente				
1														

Característica	Peso	Atributo	Peso	Métricas	Valor máximo	Valor mínimo	Tipo de valor	Valor obtenido	Valor normalizado	Concepto	Valor total Usabilidad	Concepto Usabilidad
Mantenibilidad	0,3	Capacidad de ser analizado	0,5	Promedio de distancias más larga de un contexto de navegación de la raíz a un contexto hoja	8	4	2	0	0,00	Deficiente	0,00	Deficiente
		Capacidad de ser cambiado	0,5	Número de nodos	9	5	2	0	0,00	Deficiente		

Tabla 39 .Evaluación de la calidad interna Modelo Funcional. Caso estudio: Proyecto3.

Característica	Peso	Atributo	Peso	Métricas	Valor máximo	Valor mínimo	Tipo de valor	Valor obtenido	Valor normalizado	Concepto	Valor total Usabilidad	Concepto Usabilidad
Usabilidad	0,6	Legibilidad	0,4	Promedio de número de regiones por plantillas	4	2	2	3	0,50	Aceptable	0,20	Deficiente
		Comprensibilidad	0,4	Porcentaje de plantillas bien documentadas	100	50	1	0	0,00	Deficiente		
		Facilidad de aprendizaje	0,2	Número de plantillas	3	1	1	1	0,00	Deficiente		
Característica	Peso	Atributo	Peso	Métricas	Valor máximo	Valor mínimo	Tipo de valor	Valor obtenido	Valor normalizado	Concepto	Valor total Usabilidad	Concepto Usabilidad
Mantenibilidad	0,6	Capacidad de ser analizado	0,5	Porcentaje de plantillas que especifican: conteo de color, conteo de fuente, ayuda en línea, validación de datos en el cliente, campos con atajos.	100	50	1	0	0,00	Deficiente	0,0	Deficiente

0,12
Deficiente

		Capacidad de ser cambiado	0,5	Porcentaje de platillas que permiten la definición de temas por parte de cada usuario y la plantilla de estilo	100	20	1	0	0,00	Deficiente		
--	--	---------------------------	-----	--	-----	----	---	---	------	------------	--	--

Tabla 40 .Evaluación de la calidad interna Modelo Interfaz Gráfica. Caso estudio: Proyecto3.

Si observamos los resultados de la evaluación de éstos tres proyectos por modelo y por proyecto revisado tendríamos el siguiente resumen:

Modelo	Calidad Interna del Producto		
	Proyecto 1	Proyecto 2	Proyecto 3
Estructural	Aceptable	Deficiente	Deficiente
Funcional	Deficiente	Deficiente	Deficiente
Interfaz Grafica	Aceptable	Deficiente	Deficiente

Tabla 41. Resumen de evaluación de los tres proyectos aplicando el modelo de indicadores para aplicaciones web en fase de diseño.

Los resultados para usabilidad y mantenibilidad por proyecto se resumen a continuación:

Modelo	Usabilidad		
	Proyecto 1	Proyecto 2	Proyecto 3
Estructural	Excelente	Deficiente	Deficiente
Funcional	Deficiente	Deficiente	Deficiente
Interfaz Grafica	Aceptable	Deficiente	Deficiente

Tabla 42. Resumen de evaluación de usabilidad de los modelos estructural, funcional e interfaz gráfica en los tres proyectos.

Modelo	Mantenibilidad		
	Proyecto 1	Proyecto 2	Proyecto 3
Estructural	Deficiente	Aceptable	Aceptable
Funcional	Deficiente	Deficiente	Deficiente
Interfaz Grafica	Deficiente	Deficiente	Deficiente

Tabla 43. Tabla 44. Resumen de evaluación de mantenibilidad de los modelos estructural, funcional e interfaz gráfica en los tres proyectos.

Al ver éstos resultados se concluye que los diseños realizados en los proyectos de grado de aplicaciones web de la Universidad Popular del Cesar Seccional Aguachica son deficientes, esto es una alerta para docentes y administrativos del programa quienes pueden a través de esta propuesta realizar una mejora en los aplicativos web que se están generando; si la mantenibilidad de éstos modelos en su mayoría es deficiente esto le generará problemas en el mantenimiento de muchas de las aplicaciones que sirven de soporte para algunas actividades de la Institución y que se han hecho a través de proyectos de grado.

Los soportes como manuales o el documento mismo aún no son suficientes para poder valorar la calidad del diseño. Se observa que cada proyecto presenta el diseño navegacional como lo entiende pero no hay estandarizado un formato o modelo que permita hacer una evaluación muy acertada, sólo viendo los pantallazos que colocan algunas veces en los manuales o porque la investigadora tiene acceso a algunas aplicaciones que se encuentran en el servidor de la Universidad.

El diseño de interfaz de usuario de los proyectos evaluados es muy pobre y no permite hacer una buena evaluación.

4 Conclusiones, resultados y trabajos futuros

4.1 Conclusiones

A partir de la revisión de algunos modelos de calidad y pautas de evaluación de software se observaron los diferentes factores, características y atributos que hacen parte de cada uno. Esto permitió conocer el estado de arte de los modelos de calidad y pretender proyectar hacia dónde está encaminándose la evaluación de software. El trayecto que queda por recorrer es largo y aún no se puede decir que ya hay una conceptualización establecida en el campo de la medición por ello se encuentran muchas propuestas de investigación sólo estableciendo la ontología de términos en la calidad del software y en la medición.

En el caso de los diseños y específicamente de las aplicaciones web, aunque se pueden parecer mucho los entregables a los de las aplicaciones convencionales, es claro que las necesidades de calidad en cada una es diferente debido a la diversidad de usuarios y a las características muy específicas de las aplicaciones web.

No se encontró un modelo específico para la fase de diseño ni de aplicaciones convencionales ni para la web; con la revisión del estado del arte se estableció según las necesidades de medición que el GQM era una metodología acertada para iniciar la propuesta del modelo de indicadores y una vez analizadas las principales características de cada modelo se seleccionaron dos por ser las que se consideraban más acertadas para medir un diseño en etapas tempranas del desarrollo. Una vez establecidas las necesidades y los objetivos de medición se plantean los artefactos que se deberían medir o solicitar a un proyecto de grado de la Universidad Popular del Cesar Seccional Aguachica y se estableció que fuese independiente de la plataforma, para ello se proponen tres modelos con los que se considera se puede representar el diseño de las aplicaciones web de la Universidad: el estructural, el funcional y el de interfaz gráfica. Estos diseños basados en modelos permiten la sencillez que tanto puede ayudar a que los estudiantes propongan los artefactos de esa fase que tiene tanto nivel de abstracción como es la fase de diseño. Para ayudar a la evaluación de éstos modelos se proponen tres plantillas una para cada uno, esto también permitiría que la evaluación realizada en la plantilla de Excel se pueda alimentar con mayor facilidad.

El diseño funcional propuesto es una combinación de las funcionalidades de la aplicación con lo que se estableció como diseño navegacional.

Se establecen unos atributos que pudiesen medir las características teniendo en cuenta que éstos se pudiesen observar en etapas tempranas del desarrollo dándole al modelo de indicadores un enfoque predictivo pues se cuidó que la evaluación no se hiciera cuando el desarrollo está en fase de pruebas. Para la medición se hace una propuesta novedosa y es poder medir aplicando la normalización con una fórmula que sirve para todas las métricas, lo recomendable para que la evaluación sea exitosa es que el evaluador sea conocedor del diseño de aplicaciones web pues la fórmula de normalización depende en gran medida de la experticia del evaluador.

Por último se aplica el modelo a tres proyectos de la Universidad Popular del Cesar permitiendo hacer una validación del modelo y así tener un concepto con medidas de cómo se encuentran los diseños de los proyectos de la Universidad que se encuentran catalogados como aplicaciones web. Estos resultados indican que hay mucho trabajo por hacer por parte de directores de grado, evaluadores y jefe del programa pues los resultados en general fueron deficientes, esto indica que los proyectos evaluados tienen diseño que en su mayoría no se podrían usar y también muy difíciles de mantener no sólo porque no cumplen con estándares de calidad sino porque en muchos casos no son explícitos en cómo eran esos diseños.

Esta propuesta puede mejorar en gran manera los diseños de las aplicaciones web desarrolladas como proyectos de grado de la Universidad Popular del Cesar Seccional Aguachica no sólo por la evaluación sino también por la propuesta de los modelos con sus respectivas plantillas.

Aunque no se aplicó un modelo de evaluación específico si se siguieron las pautas del estándar 14598 que especifica cómo realizar la medición del producto enfocado a la norma 9126 y que luego fue fusionado con la misma convirtiéndose en la norma ISO/IEC 25000.

4.2 Resultados

Una vez terminado la presente investigación se obtienen los siguientes resultados:

Un estado del arte de modelos de calidad.

Una propuesta de modelos de diseño para aplicaciones web independiente de la plataforma.

Una propuesta de indicador genérico.

Plantillas mínimas para diseño de aplicaciones web independiente de plataforma.

Artículos del estado de arte de modelos de Calidad para el producto y aplicaciones web.

4.3 Trabajos futuros

Uno de los trabajos futuros es la validación del indicador que se propuso para medir todas métricas pues es valioso tener una sola fórmula que permita la medición; esto hace el proceso de evaluación más sencillo y entendible.

Se recomienda la propuesta de más atributos por característica pues cada una se mide con un solo atributo por modelo; a estos nuevos atributos es recomendable buscarles métricas que ojalá sean posibles de medir en etapas tempranas del desarrollo del diseño que es una de las propuestas del modelo presentado y que además éstas métricas sean independientes de la plataforma o del tipo de metodología.

Es interesante probar éste modelo de indicadores a aplicaciones que aunque no sean web pueden aplicar a ésta propuesta sólo se debería planear cómo sería la navegación a través de cualquier otro tipo de aplicación.

Aunque Pressman en su libro Ingeniería del Software un enfoque práctico afirma que la calidad se establece en el diseño es bueno hacer una propuesta desde la fase de análisis y que esta pueda seguir su proceso con los modelos de diseño que se propusieron en este proyecto.

5 Bibliografía

5.1 Referencias Bibliográficas

(Barrios, 2012) Lina Paola Barrios y otros. *Diseño e implementación de un sistema de información en ambiente web para el proceso de ventas de la empresa Ambiflores localizada en el municipio de Aguachica, Cesar*. Proyecto de grado para optar al título de ingeniero de sistemas de la Universidad Popular del Cesar Seccional Aguachica. CD 26. Biblioteca UPC Aguachica

(Del Valle, 2009) Ana Nieves del Valle Rodríguez. *Metodología de diseño usadas en la Ingeniería web, su vinculación con las NTICS*.

(Hernández, 2003) Sergio Hernández y otros. *Metodología de la investigación*. México. Editorial McGraw-Hill.

(Olsina, 1999) Luis Olsina y otros. *Specifying Quality Characteristics and Attributes for Web Sites*. Proc. 1st Workshop on Web Engineering. ACM. Los Angeles. Mayo 1993.

(Pallares, 2011). Anyi Pallares Guillín y otros. *Estudio para conocer el estado actual del uso y apropiación de las tecnologías de la información y las comunicaciones (TIC) en el municipio de Aguachica-cesar, como fase de inicio del proyecto “Aguachica Digital”*. Proyecto de grado para optar al título de ingeniero de sistemas de la Universidad Popular del Cesar Seccional Aguachica. CD 23. Biblioteca UPC Aguachica.

(Piattini, 2003) Mario Piattini y otros. *Calidad en el desarrollo y mantenimiento del software*. Editorial Alfaomega. Madrid, España. 2003. Pág. 126.

(Piattini, 2008) Mario Piattini y otros. Medición y estimación del software. Técnicas y métodos para mejorar la calidad y la productividad. Editorial Alfaomega. Madrid, España. 2008. Pág. 115.

(Piattini, 2009) Mario Piattini G y otros. Calidad de Sistemas Informáticos .Editorial Alfaomega - RAMA.

(Pressman 2010) Roger Pressman. Ingeniería del Software.Un enfoque práctico Séptima edición. Editorial McGraw Hill.

(Shaw, 2002) Mary Shaw. What Makes Good Research in Software Engineering?.EuropeanJoint Conference of Theory and Practice of Software (ETAPS 2002), Grenoble,France.International Journal on Software Tools for Technology Transfer.

5.2 Referencias web

ISSO/IEC 9126-1. Modelo de calidad de software.
<http://www.cse.unsw.edu.au/~cs3710/PMmaterials/Resources/9126-1%20Standard.pdf>

Tomado de: <http://www.inf.utfsm.cl/~visconti/testing/Documentos/Usabilidad.pdf> Pág 33.

(Alonso, 2013) Cristina Alonso Sierra. Automatización de la medición de la usabilidad web mediante la métrica de Ivory. Tesis de Maestría. Universidad de Oviedo. Página 32.
Tomado de:
http://digibuo.uniovi.es/dspace/bitstream/10651/19429/6/TFM_AlonsoSierra.pdf, 3 de Agosto de 2014.

(Andrade, 2012) Jesús Alberto Andrade Castro. Diagramas entidad-relación y de clases de UML en el modelado de gobierno electrónico. Artículo científico de la revista CytaVol 1, Buenos Aires. Tomado de: <http://www.cyta.com.ar/ta1101/v11n1a1.htm> , 4 de Enero de 2015.

(Cachero, 2006) C. Cachero y otros. Transformación de modelos navegacionales dirigida por criterios de calidad. XV Jornadas de Ingeniería del Software y Bases de Datos. JISBD, Barcelona. 2006. <http://ceur-ws.org/Vol-227/paper10.pdf> 5 Octubre de 2015.

(Cataldi, 2000). Zulma Cataldi, (2000). Metodología de diseño, desarrollo y evaluación de software educativo. Tesis de Magíster en Informática. (Versión resumida). Facultad de Informática. Universidad Nacional de La Plata. Argentina. ISBN 960-34-0204-2. www.fi.uba.ar/laboratorios/lsi/catalditesisdemagistereninformatica.pdf

(Esparza, 2006). Pilar Esparza Salanova. Modelos de calidad Web. Clasificación de métricas. Proyecto de fin de carrera. http://www.google.com.co/url?sa=t&rct=j&q=&esrc=s&source=web&cd=3&ved=0CCMQFjAC&url=http%3A%2F%2Fwww.issi.uned.es%2FCalidadSoftware%2FNoticias%2FPFC_1.doc&ei=jH7eU_7qAa3LsATw3ICIDg&usq=AFQjCNGDLgiWeMJG9nUVJMZjwPp3Nx5Gw.

(Fernández, 2010). Adrián Fernández y otros. Evaluación de Usabilidad para Aplicaciones Web. Grupo de Investigación ISSI. Departamento de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia. <http://users.dsic.upv.es/~afernandez/files/publications/pdf/Fernandez-et-al-CLEI10.pdf> XXXVI Conferencia Latinoamericana de Informática (CLEI 2010), 18-22 Octubre 2010, Asunción, Paraguay. 2010. Consultado: Noviembre de 2015.

(Genero, 2000). Marcela Genero y otros. Earlymetricsforobjectorientedinformationsystems. Grupo de Investigación en Reutilización y Orientación a Objeto. V Jornadas de Trabajo MENHIR. Granada. <http://www.giro.infor.uva.es/Publications/2000/GMP00/vmenhir-calidad-uml.pdf>. Consultado: Octubre de 2015.

(IEEE, 1990). 610-1990 - IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries. <http://dis.unal.edu.co/~icasta/ggs/Documentos/Normas/610-12-1990.pdf>

(Insfrán, 2015). Emilio Insfrán y otros. Ingeniería de Requisitos aplicada al modelado conceptual de la interfaz de usuario. https://www.researchgate.net/publication/266583439_Ingenieria_de_Requisitos_aplicada_al_modelado_conceptual_de_interfaz_de_usuario. Consultado Octubre de 2015.

(Noriega, 2009). Lerma Noriega y otros. Usabilidad de los portales web de las cadenas de televisión mexicanas.

http://www.nosolousabilidad.com/articulos/television_mexico.htm 1o de Agosto de 2014-08-03

(Ocaña, 2015). Introducción a la Ingeniería Web Basada en UML. Jesús Ocaña Zúñiga y otros. Tomado de: <http://documents.mx/documents/ing-web-uml.html> 20 de Noviembre de 2015.

(Olsina, 1999). Metodología Cuantitativa para la Evaluación y Comparación de la Calidad de Sitios Web.

<http://www.dsi.uclm.es/personal/franciscomsimarro/cedasi/olsina-tesis.pdf>. Pag40.

(Olsina, 2002). Luis Olsina. Requerimientos de calidad para diseño y evaluación.

http://www.sel.unsl.edu.ar/ApuntesMaes/Anteriores/CursoOlsina/Transparencias/Olsina_SanLuis_Req.pdf. Consultado abril de 2014.

(Scalone, 2006). Fernanda Scalone. Estudio comparativo de los modelos y estándares de calidad del software. Tesis para optar al título de Magíster Ingeniería en Calidad. Universidad Tecnológica Nacional Facultad Regional Buenos Aires. Junio 2006. Tomada del link: <http://laboratorios.fi.uba.ar/lfi/scalone-tesis-maestria-ingenieria-en-calidad.pdf>, 19 de abril de 2014.

(Solano ,2013). Henry Solano e Israel Torres. Análisis de framework para el desarrollo de aplicaciones móviles en la plataforma android. Universidad del Azuay. Cuenca Ecuador. 2013. Link tomado de:

<http://www.google.com.co/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&sqi=2&ved=0CBoQFjAA&url=http%3A%2F%2Fdspace.uazuay.edu.ec%2Fbitstream%2Fdatos%2F3139%2F1%2F09914.pdf&ei=QlFAU8jLM5PF8QGj14DgAw&usg=AFQjCNH56Pm7jOcm1y295d0IUzR3MIqt2Q&bvm=bv.70810081,d.b2U>. Fecha de consultado 11 de Julio de 2014.

(W3C) .World Wide Web Consortium.W3C. <http://www.w3c.es/ayuda/#activity> Fecha:
1º DE Agosto de 2014-08-03

(Zhu, 2000) XiolanZhu y otros.Incorporating Quality Metrics in Centralized/Distributed Information Retrieval on the World Wide Web.Proc of 23th International ACM SIGIR on R.

<ftp://ftp10.us.freebsd.org/users/azhang/disc/disc01/cd1/out/papers/sigir/p288-zhu.pdf>.

Fecha de consultado 11 de Julio de 2014.

6 Anexos

A continuación se listan los anexos que se referenciaron en el trabajo:

Anexo A: Instrumento para la recolección de información de los proyectos de grado.

Anexo B: Plantilla para alimentar los datos referentes al modelo estructural.

Anexo C: Plantilla para alimentar los datos referentes al modelo funcional.

Anexo D: Plantilla para alimentar los datos referentes al modelo de interfaz gráfica.