



**UNIVERSIDAD AUTÓNOMA
DE AGUASCALIENTES**

CENTRO DE CIENCIAS BÁSICAS

MAESTRÍA EN CIENCIAS EXACTAS, SISTEMAS Y DE LA INFORMACIÓN

ÁREA: INGENIERÍA DE SOFTWARE

TESIS

**“ESTUDIO DEL IMPACTO DE LAS TEORÍAS GESTALT EN LA
CALIDAD DE LOS SISTEMAS DE INFORMACIÓN”**

PRESENTA

L.I. ERIKA GUADALUPE ECHAVARRÍA ÁLVAREZ

DIRECTOR DE TESIS

JUAN MANUEL GÓMEZ REYNOSO, Ph.D.

SINODALES

Dra. Estela Lizbeth Muñoz Andrade

M. en C. Marcelo de Jesús Pérez Ramos

Aguascalientes, Ags. Noviembre 2010



Centro de Ciencias Básicas

L.I. ERIKA GUADALUPE ECHAVARRÍA ALVAREZ
PASANTE DE LA MAESTRIA EN CIENCIAS EXACTAS,
SISTEMAS Y DE LA INFORMACIÓN
P R E S E N T E .

Estimado (a) alumno (a) Echavarría:

Por medio de este conducto me permito comunicar a Usted que habiendo recibido los votos aprobatorios de los revisores de su trabajo de tesis y/o caso práctico titulado: **“ESTUDIO DEL IMPACTO DE LAS TEORÍAS GESTALT EN LA CALIDAD DE LOS SISTEMAS DE INFORMACIÓN”**, hago de su conocimiento que puede imprimir dicho documento y continuar con los trámites para la presentación de su examen de grado.

Sin otro particular me permito saludarle muy afectuosamente.

A T E N T A M E N T E

Aguascalientes, Ags., 22 de noviembre de 2010

“SE LUMEN PROFERRE”

EL DECANO

DR. FRANCISCO JAVIER ÁLVAREZ RODRÍGUEZ



c.c.p.- Archivo
FJAR,mjda

Aguascalientes, Ags. Noviembre de 2010

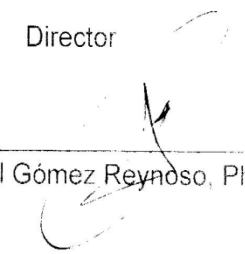
Dr. Francisco Javier Álvarez Rodríguez
Decano del Centro de Ciencias Básicas

Por medio de este conducto autorizamos a la tesista:

L.I. Erika Guadalupe Echavarría Álvarez


La impresión de su documento de tesis titulado "***Estudio del impacto de las teorías Gestalt en la calidad de los sistemas de información***", ya que cumple con los requisitos de contenido y forma exigidos por la Universidad Autónoma de Aguascalientes.

Director



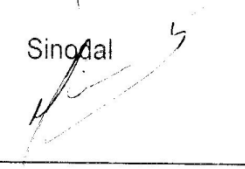
Juan Manuel Gómez Reynoso, Ph.D.

Sinodal



Dra. Estela Lizbeth Muñoz Andrade

Sinodal



M. en C. Marcelo de Jesús Pérez Ramos

Agradecimientos

Dedico el presente trabajo y doy las gracias a Dios por haberme permitido lograr una meta más en mi vida y en mi carrera, por haberme dado una gran familia y la fortaleza para superar los obstáculos encontrados.

Agradezco infinitamente a mis padres por haberme dado la oportunidad de estudiar una carrera e impulsarme a seguirme superando, por apoyarme incondicionalmente en todas mis decisiones, y ayudarme a cumplir mis objetivos. A mi madre por ser una gran mujer y ser humano, pero sobre todo una gran madre y amiga. A mi padre por alentarme a ser mejor cada día, por ayudarme a no darme por vencida, por respetarme y entenderme siempre. Gracias a ambos por estar conmigo siempre que los he necesitado, por educarme y enseñarme a valorar las cosas.

Gracias a mi hermana por haber llegado a mí y enseñarme cada día el valor de la vida, por ser una guerrera y un ejemplo de lucha constante, por ser un motivo para más para superarme cada día.

A Rodrigo por estar conmigo en los buenos y malos momentos, por apoyarme comprenderme y preocuparse por mí siempre.

A todas mis tías por su apoyo y preocupación por mis estudios.

A la Universidad Autónoma de Aguascalientes por darme la oportunidad de formarme y superarme como profesional.

A mi Director de Tesis Juan Manuel Gómez Reynoso por su paciencia y dedicación para guiarme en la elaboración de éste trabajo y por compartir conmigo sus conocimientos y experiencia.

Al CBTa. 88 por haberme proporcionado los medios para completar mi trabajo de investigación.

Finalmente gracias a mis compañeros de clase por colaborar conmigo en éste proyecto siempre que pudieron, por su solidaridad y compañerismo.

TESIS TESIS TESIS TESIS TESIS

Resumen

La calidad del software representa un importante reto en la actualidad, ya que frecuentemente se encuentran numerosos ejemplos de software que carecen de ésta. La frustración y la ansiedad forman parte de la vida cotidiana de muchos usuarios de sistemas de información computarizados. En el presente trabajo de investigación se destaca la importancia de realizar evaluaciones de la calidad del software desde el punto de vista del producto, involucrando al usuario final. Con el objetivo de lograr productos de calidad se recurre al empleo de estándares como el ISO/IEC 9126 (1991) y el modelo de McCall (1977), los cuales se utilizan en la presente investigación. Por lo general el usuario tiende a juzgar la calidad del software basándose en la interfaz gráfica de usuario (GUI), por lo que resulta vital realizar una evaluación previa de ésta. La construcción de GUIs representa un campo interdisciplinario es el que intervienen áreas como: la ergonomía, el diseño gráfico, y la psicología. La literatura reporta que existe carencia de investigaciones que involucren las ciencias cognitivas en el desarrollo de software, por lo que en éste trabajo se emplea la teoría psicológica Gestalt en el diseño de GUIs y se evalúa el impacto que dicha teoría tiene en la calidad del software, tomando como elementos de medición la facilidad de uso, la facilidad de aprendizaje y el nivel de aceptación por parte del usuario. En general éste trabajo de investigación busca contribuir con el conocimiento de la evaluación de la calidad del software desde el enfoque del producto, así como colaborar con los ingenieros de software en el conocimiento del diseño de GUIs utilizando la teoría Gestalt.

Índice de contenido

Capítulo 1

Introducción	1
1.1 Contexto general	1
1.2 Definición del problema y relevancia	7
1.3 Tipo de investigación.....	10
1.4 Pregunta de investigación	11
1.4.1 Pregunta general.....	11
1.4.2 Preguntas específicas	11
1.5 Objetivo general de investigación.....	11
1.6 Hipótesis de investigación	12

Capítulo 2

Marco Teórico	16
2.1 Calidad	16
2.2 Estándares	19
2.3 Calidad de software.....	21
2.3.1 Modelos de calidad de software	24
2.3.1.1 Modelo de McCall.....	24
2.3.1.2 Modelo ISO/IEC 9126	35
2.3.1.3 Modelo de FURPS	40
2.3.1.4 Modelo de Bohem	41
2.3.2 Resumen de los modelos de calidad de software	42
2.4 Métricas.....	44
2.5 Interfaces gráficas de usuario	50
2.5.1 Calidad de las interfaces gráficas de usuario	58
2.6 Teoría psicológica Gestalt.....	61
2.7 Impacto en la calidad de GUIs diseñadas con Gestalt	67

Capítulo 3

Metodología de investigación.....	69
--	-----------

3.1	Descripción del experimento	69
3.1.1	Población objetivo	71
3.1.2	Selección de la muestra	71
3.2	Diseño del instrumento de evaluación.....	78
3.3	Modelo de Investigación.....	84
3.4	Estudio piloto.....	84
3.4.1	Análisis de reactivos del instrumento de evaluación	86
3.4.1.1	Factor Facilidad de comprensión	86
3.4.1.2	Factor Facilidad de operación	88
3.4.1.3	Factor Atractividad	89
3.4.1.4	Factor Facilidad de aprendizaje	91
3.4.1.5	Factor Aceptación del usuario	92
3.5	Resumen de validez de los reactivos del instrumento.....	94
Capítulo 4		
Análisis de resultados.....		95
4.1	Prueba de medias	95
4.1.1	Factor Facilidad de comprensión	95
4.1.2	Factor Facilidad de operación	102
4.1.3	Factor Atractividad	109
4.1.4	Factor Facilidad de aprendizaje	113
4.1.5	Factor Aceptación del usuario	119
Capítulo 5		
Conclusiones		123
5.1	Conclusiones generales	123
5.2	Limitaciones de la investigación	131
5.3	Recomendaciones.....	132
Anexos		
	Productividad	134
Bibliografía		142

Índice de figuras

Figura 1. Modelo de McCall (Adaptado de Pressman, 2007, pág. 464).....	25
Figura 2. Modelo de McCall Factores y Criterios, (Adaptado de McCall A., Richards, & Walters, 1977).	28
Figura 3. Modelo ISO/IEC 9126.....	36
Figura 4. Factores de Calidad ISO/IEC 9126 (Adaptado de Abran, Khelifi, & Surnyn, 2003).....	37
Figura 5. Modelo de Bohem (Adaptado de Dávila & Mejía, 2003)	42
Figura 6. Marco de trabajo de las métricas de calidad de software (Adaptado de Khan, Mustafa, & Ahson, 2008).....	47
Figura 7. Similitud (Adaptado de (Graham, 2008))	62
Figura 8. Continuidad.....	63
Figura 9. Punto focal (Adaptado de Dempsey, Laurence, & Juhani, 2001)	63
Figura 10. Figura fondo (Adaptado de Graham, 2008).....	64
Figura 11. Unidad/Armonía (Adaptado de Dempsey, Laurence, & Juhani, 2001) .	64
Figura 12. Simetría.....	64
Figura 13. Proximidad (Adaptado de Graham, 2008).....	65
Figura 14. Cierre	65
Figura 15. Correspondencia isomorfa (Adaptado de Dempsey, Laurence, & Juhani, 2001)	66
Figura 16. Concisión/Pregnancia	66
Figura 17. Simplicidad.....	66
Figura 18. Histogramas. Edad de los usuarios.....	72
Figura 19. Histogramas. Género de los usuarios	72
Figura 20. Histogramas. Años de experiencia de los usuarios.....	73
Figura 21. Histogramas. Horas a la semana de uso de computadora.....	73
Figura 22. Histogramas. Horas a la semana de uso de Internet	74
Figura 23. Histogramas. Uso de internet para búsqueda de información.....	74
Figura 24. Histogramas. Uso de internet para realizar trabajo	75

Figura 25. Histogramas. Uso de internet para juegos 75

Figura 26. Histogramas. Uso se internet para descargar música..... 76

Figura 27. Histogramas. Uso de internet para ver videos 76

Figura 28. Histogramas. Otros usos de internet 77

Figura 29. Histogramas. Nivel de experiencia de los usuarios 77

Figura 30. Modelo Propuesto para la presente investigación..... 84

Figura 31. Histogramas. Factor: Facilidad de comprensión 87

Figura 32. Histogramas. Factor: Facilidad de operación 89

Figura 33. Histogramas. Factor: Atractividad 90

Figura 34. Histogramas. Factor: Facilidad de aprendizaje 92

Figura 35. Histogramas. Factor: Aceptación de usuario..... 93

Figura 36. Histogramas. Sub-factor: Lenguaje familiar 96

Figura 37. Histogramas. Sub-factor: Tamaño del texto 97

Figura 38. Histogramas. Sub-factor: Tipo de letra 98

Figura 39. Histogramas. Sub-factor: Contraste de color de fondo y texto 99

Figura 40. Histogramas. Sub-factor: Redacción de texto 100

Figura 41. Histogramas. Sub-factor: Significado de símbolos 101

Figura 42. Gráfico de barras para comparar las medias de los sub-factores de
Facilidad de comprensión. 101

Figura 43. Histogramas. Sub-factor: Tareas similares 102

Figura 44. Histogramas. Sub-factor: Control al usuario..... 103

Figura 45. Histogramas. Sub-factor: Control de errores 104

Figura 46. Histogramas. Sub-factor: Claridad de mensajes 105

Figura 47. Histogramas. Sub-factor: Ayuda de mensajes 106

Figura 48. Histogramas. Sub-factor: Navegación..... 107

Figura 49. Histogramas. Sub-factor: Movimientos de navegación 108

Figura 50. Gráfico de barras para comparar las medias de los sub-factores de
Facilidad de operación. 108

Figura 51. Histogramas. Sub-factor: Organización de elementos 109

Figura 52. Histogramas. Sub-factor: Combinación de colores 110

Figura 53. Histogramas. Sub-factor: Color de fondo 111

Figura 54. Histogramas. Sub-factor: Atractividad de símbolos..... 112

Figura 55. Gráfico de barras para comparar las medias de los sub-factores de Atractividad 113

Figura 56. Histogramas. Sub-factor: Recordar acciones anteriores..... 114

Figura 57. Histogramas. Sub-factor: Tareas fáciles de comprender 115

Figura 58. Histogramas. Sub-factor: Tareas fáciles de recordar 116

Figura 59. Histogramas. Sub-factor: Información ordenada de manera jerárquica 117

Figura 60. Histogramas. Sub-factor: Cantidad de elementos..... 118

Figura 61. Gráfico de barras para comparar las medias de los sub-factores de Facilidad de aprendizaje 118

Figura 62. Histogramas. Sub-factor: Beneficio percibido 119

Figura 63. Histogramas. Sub-factor: Confianza..... 120

Figura 64. Histogramas. Sub-factor: Satisfacción de necesidades 121

Figura 65. Gráfico de barras para comparar las medias de los sub-factores de Nivel de aceptación del usuario 121

Índice de tablas

Tabla 1. Criterios y métricas del modelo de McCall (Adaptado de (McCall A., Richards, & Walters, 1977))	34
Tabla 2. ISO 9126-1 Características y sub-características (Adaptado de (Chua & Dyson, 2004))	39
Tabla 3. Comparativo de Modelos de Calidad.....	43
Tabla 4. Actividades previas al experimento	70
Tabla 5. Actividades realizadas en el experimento con el grupo experimental y de control.....	70
Tabla 6. Hipótesis de la investigación.....	78
Tabla 7. Factor Facilidad de uso, ISO/IEC 9126 - McCall	79
Tabla 8. Características fundamentales de una GUI y factores de medición	81
Tabla 9. Desarrollo de ítems.....	83
Tabla 10. Estadística descriptiva. Factor: Facilidad de comprensión	87
Tabla 11. Estadística descriptiva. Factor: Facilidad de operación	88
Tabla 12. Estadística descriptiva. Factor: Facilidad de aprendizaje	91
Tabla 13. Estadística descriptiva. Factor: Atractividad	90
Tabla 14. Estadística descriptiva. Factor: Aceptación de usuario.....	93
Tabla 15. Valores de Alfa de Cronbach	94
Tabla 16. Resultado de las hipótesis planteadas	122

Capítulo 1

Introducción

1.1 Contexto general

Hoy en día el tratamiento de la información supone un proceso complejo dadas las características y volumen de los datos. Por lo anterior, se recurre al uso de software que brinde soporte en la administración de la información. “En la actualidad, el software de computadora es la tecnología individual más importante en el ámbito mundial” (Pressman, 2007, pág. 1). Enfocada al proceso de software se encuentra la disciplina de la ingeniería de software, la cual pretende ser un marco de trabajo para las tareas que se requieren en la construcción de software de alta calidad (Pressman, 2007). Existe una gran variedad de definiciones del término ingeniería de software, entre las cuales destaca la propuesta por la IEEE (1990, pág. 67), que la define como: “La aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento del software; es decir, la aplicación de la ingeniería al software”. Nasib (2005) afirma que la calidad del software está ganando mucha más atención en estos días, así como se está poniendo más énfasis en el desarrollo de productos de alta calidad. De acuerdo con Vega (2008) entre los objetivos básicos de la ingeniería del software se encuentra la satisfacción del cliente. Según Pressman (2007) si un producto de software satisface al cliente proporcionándole un beneficio sustancial, éste estará dispuesto a tolerar problemas ocasionales de confiabilidad y desempeño.

A pesar de la importancia que ha adquirido el concepto de calidad de software Howles (2003) argumenta que en la actualidad, y a pesar de los esfuerzos, la industria del software no puede garantizar ampliamente productos de calidad, ya que frecuentemente existe en el mercado software deficiente que no satisface las necesidades de los clientes, lo que repercute en pérdidas económicas. Fernández (2008) explica que los defectos del software se encuentran cotidianamente, por ejemplo en la repentina aparición de la pantalla azul de la computadora, en la falla del sistema de gestión aeroportuario que provoca retrasos, entre otros, expone que las consecuencias de estas fallas se ven reflejadas principalmente en pérdidas económicas y de tiempo, esto cuando se habla de problemas cotidianos. Sin embargo también explica que los problemas pueden resultar más graves cuando se trata de sistemas críticos, es decir, aquellos cuyo fallo puede provocar cuantiosas pérdidas económicas, ambientales o sociales, e incluso la vida humana, Fernández (2008) cita como ejemplo el caso del lanzamiento del cohete Ariane 5 de la Agencia Espacial Europea, en 1996, el cual llevó 10 años de construcción y 7000 millones de euros en costo, que explotó por un defecto en el software de control interno antes de que pasara un minuto de vuelo.

En su lista llamada “Riesgos en la utilización de sistemas informáticos y tecnologías” Neumann (2009) documenta una gran cantidad de incidentes relacionados con el uso de sistemas informáticos en diferentes sectores. Por ejemplo, muestra el caso de un hospital en el que se administraron dosis incorrectas de medicamentos debido a que el sistema mostraba información de signos vitales, resultados de laboratorio y padecimientos, bajo el nombre de otro paciente, aparte de mostrar información poco clara, lo que provocó que se administraran medicamentos y drogas controladas, a los pacientes equivocados. La falla del sistema se presentó después de una actualización de la versión del software.

Con ejemplos como los anteriores es claro que nos encontramos ante un problema que requiere ser atendido de manera inmediata, la calidad del software debe convertirse en el objetivo de toda persona o empresa dedicada a la

producción éste. De aquí que surge la siguiente incógnita ¿cómo lograr software de calidad? Para mejorar un producto es necesario contar con un punto de referencia del cual partir, es decir, tener una medición como base. Khan et al. (2008, pág. 103), explica que DeMarco afirma que “No se puede controlar lo que no se puede medir”. De esta manera el software tendrá que ser medido para poder ser mejorado. El objetivo de llevar a cabo la medición del software es obtener datos que nos ayuden a controlar el tiempo, el dinero y la calidad de los productos (Florac, 1992). La falta de pruebas de software disminuyen su confiabilidad y afecta negativamente la calidad (Nasib S., 2005). Para alcanzar un producto con calidad, la medición resulta esencial (Pressman, 2007). A pesar de la importancia que tiene la medición del software, Dávila et al. (2003) explican que hoy en día muy pocas industrias dedicadas al desarrollo de software utilizan procesos de evaluación y análisis para este efecto, también expone que la mayoría de los estudios se enfocan en las actividades de administración de los proyectos de desarrollo de software.

La calidad del software se puede dividir en dos amplios rubros: calidad del proceso y calidad del producto. Para el usuario final, la calidad del proceso de desarrollo de software resulta transparente, contrario a lo que sucede con la calidad del producto, ya que es con éste, con el que está en contacto constante, y a través de él busca obtener resultados que lo satisfagan. De aquí que la medición de la calidad del producto resulta indispensable.

Con la intención de contar con un punto de referencia del cual partir para lograr la calidad del software, han surgido los modelos de calidad, los cuales buscan ser una guía que proporcione las bases para obtener un producto de calidad. Entre los primeros modelos de calidad de software, se encuentra el modelo de McCall, surgido en 1977, el cual establece 3 áreas principales que intervienen en la calidad del software: 1. Calidad en la operación del producto, 2. Revisión de la calidad del producto de software y 3. Transición del producto. McCall distingue dos niveles de características de calidad: factores y criterios (Khan, Mustafa, & Ahson, 2008). Los factores identificados en el modelo son (Cavano & McCall, 1978):

TESIS TESIS TESIS TESIS TESIS

corrección, fiabilidad, eficiencia, integridad, usabilidad, mantenibilidad, facilidad de prueba, flexibilidad, portabilidad, reusabilidad e interoperabilidad. Derivado del modelo de McCall encontramos el ISO/IEC 9126 el cual según Abran et al. (2003), aunque no se trata de un modelo exhaustivo, constituye el modelo de calidad de software más completo hasta la fecha. El modelo está compuesto por 6 factores (Pressman, 2007), los cuales son: funcionalidad, confiabilidad, facilidad de uso, eficiencia, facilidad de mantenimiento y portabilidad.

Adicionalmente, en dichos modelos cada factor se compone de una serie de subcaracterísticas o atributos. Por ejemplo, la facilidad de uso en el ISO/IEC 9126, a su vez, se compone de cuatro subcaracterísticas: entendimiento, aprendizaje, operabilidad y atraktividad (Chua & Dyson, 2004). En el Modelo de McCall et al. (1977) la usabilidad está constituida por tres atributos: familiaridad, comunicación y operabilidad. Dado que el ISO/IEC 9126 está basado en el modelo de McCall se pueden encontrar factores que apuntan a evaluar elementos comunes entre sí. Por ejemplo, para el modelo ISO/IEC 9126, la facilidad de uso se refiere a "la facilidad con que se usa el software de acuerdo con los siguientes subatributos: facilidad de comprensión, facilidad de aprendizaje y operabilidad" (Pressman, 2007, pág. 466). De igual manera, el modelo de McCall hace referencia al factor usabilidad, el cual pertenece al área de operación del producto y está definido como el esfuerzo necesario para aprender, operar y preparar los datos de entrada de un programa e interpretar la salida (Cavano & McCall, 1978). Se puede observar que ambos se refieren a lo mismo aunque con nombre diferente. A pesar de ser ampliamente aceptados, los modelos anteriores son fuertemente criticados por proporcionar solamente un marco de referencia para evaluar la calidad del software, dejando de lado el hecho de cómo hacerlo. Sibisi et al. (2007) explican que las características de los modelos no pueden ser medidas directamente, por lo tanto se deben definir métricas relacionadas con la característica de calidad que se desea evaluar.

La calidad del software, como la calidad en general, es de naturaleza subjetiva, es decir, depende del punto de vista de la persona que la juzga. En el caso de la calidad del software Pressman (2007) afirma que ésta es juzgada de acuerdo a la interfaz de usuario. La interfaz de usuario representa el mecanismo que permite la comunicación entre el sistema y el usuario, de ésta manera, siendo el software de naturaleza intangible, la interfaz representa el elemento más visible para el usuario. Un mal diseño de interfaz reduce la capacidad del usuario para aprovechar las ventajas y funcionalidades de una aplicación, una interfaz deficiente puede llevar al fracaso una aplicación bien diseñada y con una implementación sólida. Una mala interfaz de usuario puede contribuir al error humano, incluso en daños personales y financieros (Perlman, 1996). “Es posible afirmar que la interfaz de usuario es el elemento más importante de un sistema o producto de cómputo” (Pressman, 2007, pág. 378). Es por esto que se debe poner especial atención en la calidad del diseño de la interfaz de usuario, ya que de ella depende que todo un sistema de software sea o no percibido como un producto de calidad. Dada la importancia de la calidad de una interfaz de usuario, resulta indispensable su evaluación. De forma ideal, la evaluación de una interfaz de usuario se debe llevar a cabo en torno a la usabilidad del sistema (Somerville, 2006).

Siendo la interfaz de usuario un elemento que compone el software, también se debe recurrir a las métricas para evaluar su calidad. Preece et al. (1994, pág. 602) definen la evaluación de una interfaz de usuario como la “reunión de datos sobre la usabilidad de un diseño o un producto por parte de un grupo de usuarios específicos para una actividad en concreto dentro de un entorno determinado de trabajo y como una parte integral del proceso de diseño centrado en el usuario”. Los usuarios demandan sistemas fáciles de aprender y que les ayuden a hacer su trabajo, que no los detenga, engañe o confunda, que no les lleve a cometer errores o les dificulte la terminación del trabajo (Constantine, 1995). Para obtener un producto de calidad que logre la satisfacción del usuario, la construcción y diseño de la interfaz de usuario debe implicar un proceso creativo y tecnológico en

el que deben intervenir un conjunto de disciplinas como el diseño gráfico, psicología, ergonomía, sociología, entre otras. En el área de diseño de interfaces existe una falta de directrices basadas en investigaciones actuales de la ciencia cognitiva (Aberg & Chang, 2005). Las ciencias cognitivas estudian los procesos de la mente humana: como aprende, como recuerda, como procesa la información y que hace con ella (Mercovich, 1999). El equilibrio y organización general de los elementos gráficos es determinante en la creación de interfaces gráficas (Buitrón de la Torre, 2004). En el campo de la psicología se encuentra la teoría Gestalt, la cual es considerada como la teoría formal de la percepción visual (Gordon, 2004), dicha teoría explica como la mente percibe los elementos gráficos que llegan a ella por medio de los sentidos. Dempsey et al. (2001) explican que ésta teoría está constituida por una serie de principios o leyes que se aplican para sugerir como presentar los elementos visuales, a fin de lograr resultados efectivos, también expone que entre las leyes que componen dicha teoría sobresalen 11, las cuales representan los principales aspectos de la teoría Gestalt acerca de las formas visuales (Dempsey, Laurence, & Juhani, 2001), éstas son: Similitud, Continuidad, Punto focal, Figura fondo, Armonía, Balance, Proximidad, Cierre, Correspondencia isomorfa, Prägnanz y Simplicidad. Dichas leyes o principios pueden ser interpretados como bases o reglas para el diseño de interfaces de usuario, con ellas se buscará construir productos cuyos elementos sigan una estructura que la mente reconozca y asocie de manera natural, para con ello buscar que la interfaz resulte con un alto grado de usabilidad y en consecuencia lograr un producto de calidad.

En el presente trabajo se propone aplicar las leyes de la teoría Gestalt al diseño de las interfaces gráficas de usuario (GUIs) y medir el impacto que dicha teoría tiene en la calidad del software. La evaluación de la calidad se llevará a cabo haciendo uso de los modelos de calidad de software de McCall e ISO/IEC 9126. La literatura no reporta un estudio en el cual se analice el impacto que Gestalt puede tener en el desarrollo de las GUIs desde el punto de vista de calidad, por lo que consideramos importante llevar a cabo el estudio. Nuestro trabajo pretende

evaluar la calidad de las interfaces gráficas de usuario tomando como elementos de medición la facilidad de uso, la facilidad de aprendizaje y el grado de aceptación por parte del usuario. Se pretende contar con dos tipos de interfaz: la primera construida sin los principios de la teoría Gestalt, y la segunda, construida siguiendo los principios que propone dicha teoría.

El sistema que se pretende evaluar es el portal Web de una institución educativa de nivel medio superior, que ya se encuentra construido, el cual tiene una interfaz desarrollada sin Gestalt. Se pretende rediseñar la primera interfaz, basándose en los principios de Gestalt; de ésta manera se podrán evaluar ambas interfaces y estimar el impacto que Gestalt tendría en la calidad de éstas.

1.2 Definición del problema y relevancia

El compromiso de un ingeniero de software es entregar a su cliente un producto con calidad, que satisfaga las necesidades para las cuales fue creado, e idóneamente, que cuente con un conjunto de características que lo hagan fácil de usar, de aprender y de operar (Pressman, 2007; Somerville, 2006). A pesar de ser éste el objetivo de los ingenieros de software, en la actualidad muchos usuarios manifiestan que el software que utilizan cotidianamente no les satisface completamente, por ejemplo Ko et al. (2006) reportan en un estudio realizado acerca del análisis lingüístico de cómo la gente describe los problemas del software, que el 26% de los problemas manifestados se refieren a un componente de la GUI. Otro estudio presentado por Ahonen et al. (2003) acerca de los problemas que afectan los proyectos de ingeniería de software reportan que respecto al diseño de GUI no se destinan los recursos necesarios o no hay tiempo suficiente para su desarrollo, lo que resulta en una mala interfaz de usuario que tiene como consecuencia que los usuarios consideren toda la aplicación como mala.

Desde la perspectiva de la ingeniería del software la interfaz del usuario juega un papel preponderante en el desarrollo y puesta en marcha de cualquier sistema.

Shneiderman (1999) declara que se debe aumentar la concientización y generar debate entre los diseñadores y organizaciones para alcanzar la facilidad de aprendizaje, el rápido desempeño y la disminución de las tasas de error en el desarrollo de GUIs, todo esto ayudados de las teorías y la investigación empírica. A una década de ésta propuesta, aún se presentan problemas importantes respecto a desarrollo de GUIs.

Somerville (2006) argumenta que aunque muchos especialistas a menudo trabajan en el diseño gráfico y en el diseño de sistemas ergonómicos, normalmente sólo las organizaciones grandes emplean diseñadores especialistas de interfaces para sus aplicaciones de software. Por lo tanto, los ingenieros de software a menudo deben tomar la responsabilidad de diseñar la interfaz de usuario, así como del diseño del software que permita su implementación, además explica que aun cuando los diseñadores y programadores de software son usuarios competentes en la tecnología utilizada en la implementación de interfaces, el problema radica en que a menudo resultan productos poco atractivos, complejos e inapropiados para sus usuarios objetivo. Machiraju (1996) afirma que la interfaz de usuario es uno de los factores de mayor impacto para la organización.

GUIs se producen muchas, todos los días, y en todo el mundo. Estas van dirigidas a una gran diversidad de usuarios con diferentes características sociales, etnográficas, económicas, culturales, religiosas, físicas, políticas, entre otras. Shneiderman (2000) explica que algunos usuarios necesitan solo unos pocos minutos para orientarse dentro del sistema y comenzar a usarlo con éxito; otros en cambio necesitan mas tiempo para adquirir el conocimiento básico acerca de los objetos y comportamiento de la herramienta.

El problema es que producir GUI eficiente no es una tarea sencilla puesto que es a través de ella se establece la comunicación del usuario con el sistema y viceversa. Debido a esta particularidad, ésta tiene que contar con características básicas cognitivas que permitan al usuario una interacción sencilla y eficiente con la

TESIS TESIS TESIS TESIS TESIS

aplicación. Cuando se tiene una interfaz de usuario poco atractiva, no amigable y/o compleja, también se tiene un usuario frustrado, molesto e insatisfecho, yendo todo lo anterior en contra del compromiso y objetivo de un ingeniero de software.

Tomando en cuenta todo lo anterior se puede decir que el desarrollo de interfaces de usuario representa un campo interdisciplinario, donde el empleo de las áreas de carácter humano resulta esencial. Shneiderman (1981) argumenta que los factores humanos no pueden ser agregados como refinamiento o complemento en el diseño de interfaces de usuario, sino que deben ser la parte central durante el análisis de requerimientos y a través de todo el desarrollo del sistema, también explica que dichos factores ya se reconocen como importantes componentes de la calidad de un sistema de software, y expone que a pesar de esto, dichos factores solo se miden de manera vaga o superficial, en caso de que se lleve a cabo alguna medición al software.

Shneiderman (1981) argumenta que los factores humanos que intervienen en la calidad del software no resultan fácil de medir, también resalta la importancia de llevara a cabo una evaluación adecuada, ya que ésta podría ahorrar costos y problemas posteriores a la implementación. También explica que la colaboración de profesionales en áreas como la psicología puede ser de gran utilidad.

Garlitz (2007) manifiesta que en el área de interfaces gráficas de usuario existen problemas como las limitaciones de la comprensión humana y la falta de guías de diseño que proporcionen un punto de referencia. Así mismo también expone la siguiente reflexión: “Si la gente no puede utilizar eficazmente un sistema, entonces ¿de quién es la culpa? al parecer, la falla radica en el diseño del sistema, ya que las personas no pueden ser rediseñadas” (Galitz, 2007, pág. 88). Dado lo anterior, surge la siguiente incógnita ¿cómo hacer GUIs fáciles de usar? Atendiendo a esta problemática, existen varias alternativas propuestas por diversos autores. Por ejemplo, las 3 “reglas de oro” para del diseño de las interfaz de Mandel (1997) o los principios propuestos por Nielsen (2005), quien se desenvuelve en el área de usabilidad en web.

Entre los beneficios que trae un buen diseño de interfaz de usuario Galitz (2007) resalta el incremento en la productividad de las personas que las usan, la disminución del tiempo invertido en realizar transacciones y la disminución de errores. De la misma manera Vera (2006) afirma que con una interfaz bien diseñada, el usuario debe sentirse arropado en el momento de realizar sus búsquedas y cubrir sus necesidades de información, gracias al equilibrio entre contenidos y gráficos. De otra forma sería un fracaso, porque la interfaz no cubriría bien las necesidades de los usuarios, a quienes se dirige y por quienes se planea.

En base a los párrafos anteriores podemos decir que es de suma importancia poner especial atención en el desarrollo de interfaces gráficas de usuario así como en su evaluación. Es por esto que la presente investigación se propone implementar una serie de principios de una teoría psicológica en el desarrollo de interfaces gráficas de usuario, interviniendo de ésta manera los factores humanos en la construcción de las interfaces, con la finalidad de averiguar el impacto que tiene su implementación en la calidad de un sistema de software.

1.3 Tipo de investigación

De acuerdo con Newman (2000), el estudio realizado constituye una investigación aplicada, dado que se hizo uso de teoría existente con el fin de generar conocimiento nuevo en un ámbito de estudio particular, el cual demandó desarrollar una herramienta especial a través de la que se pudo medir el impacto en la calidad de software que tiene la teoría Gestalt. El estudio que se llevó a cabo de manera experimental es con dos grupos; uno sirvió como grupo de control y el otro como grupo de estudio. De acuerdo a la forma en que el tiempo entra en el estudio, este se considera de tipo transeccional, dado que solo se aplicará en un momento único.

1.4 Pregunta de investigación

El presente trabajo de investigación se enfoca en la evaluación del impacto que puede tener la aplicación de las teorías Gestalt en la calidad de los sistemas de información. Para llevar a cabo dicha evaluación se han tomado los siguientes elementos de medición: Facilidad de uso, Facilidad de aprendizaje y el Nivel de aceptación por parte del usuario. En base a esto se han formulado las siguientes preguntas.

1.4.1 Pregunta general

¿La aplicación de los principios de la teoría Gestalt ayuda a construir interfaces gráficas de usuario con mayor calidad, que resultan más fáciles de usar, más fáciles de aprender y más aceptadas por los usuarios en comparación a las que se elaboran sin estos principios?

1.4.2 Preguntas específicas

- ¿La aplicación de los principios de la teoría Gestalt ayuda a construir GUIs que resultan más **fáciles de usar** en comparación a las que se construyen sin éstos principios?
- ¿La aplicación de los principios de la teoría Gestalt ayuda a construir GUIs que resultan más **fáciles de aprender** en comparación a las que se construyen sin éstos principios?
- ¿La aplicación de los principios de la teoría Gestalt ayuda a construir GUIs que tienen un alto **nivel de aceptación por parte del usuario** en comparación a las que se construyen sin éstos principios?

1.5 Objetivo general de investigación

Con la finalidad de establecer qué es lo que pretende lograr el presente trabajo de investigación se ha establecido el siguiente objetivo:

Medir el impacto del diseño de las interfaces gráficas de usuario utilizando las teorías Gestalt en la calidad del sistema, tomando como elementos de medición la facilidad de uso, la facilidad de aprendizaje y el grado de aceptación por parte del usuario.

1.6 Hipótesis de investigación

Debido a que el presente trabajo de investigación trata de evaluar el impacto que pueden tener las teorías Gestalt en la calidad de los sistemas de información, se han establecido las siguientes hipótesis:

H₁: Las GUI construidas bajo los principios de la teoría Gestalt son más **fáciles de usar**, comparadas con las GUI que se construyen sin estos principios.

H_{1.A}: Las GUI construidas bajo los principios de la teoría Gestalt son más **fáciles de comprender** comparadas con las GUI que se construyen sin estos principios.

H_{1.A1}: Las GUIs construidas bajo los principios de la teoría Gestalt tienen un **lenguaje más familiar** comparadas con las GUIs que se construyen sin estos principios.

H_{1.A2}: Las GUIs construidas bajo los principios de la teoría Gestalt tienen un **tamaño de texto más adecuado** comparadas con las GUIs que se construyen sin estos principios.

H_{1.A3}: Las GUIs construidas bajo los principios de la teoría Gestalt tienen un **tipo de texto más adecuado** comparadas con las GUIs que se construyen sin estos principios.

H_{1.A4}: Las GUIs construidas bajo los principios de la teoría Gestalt tienen un **texto más fácil de leer debido al contraste entre el color del texto y el color de fondo** comparadas con las GUIs que se construyen sin estos principios.

TESIS TESIS TESIS TESIS TESIS

H_{1.A5}: Las GUIs construidas bajo los principios de la teoría Gestalt tienen un **texto más fácil de comprender debido a que la redacción es más adecuada** comparadas con las GUIs que se construyen sin estos principios.

H_{1.A6}: Las GUIs construidas bajo los principios de la teoría Gestalt tienen **símbolos más fáciles de comprender** comparadas con las GUIs que se construyen sin estos principios.

H_{1.B}: Las GUI construidas bajo los principios de la teoría Gestalt son más **fáciles de operar** comparadas con las GUI que se construyen sin estos principios.

H_{1.B1}: Las GUIs construidas bajo los principios de la teoría Gestalt tienen **operaciones más consistentes** comparadas con las GUIs que se construyen sin estos principios.

H_{1.B2}: Las GUIs construidas bajo los principios de la teoría Gestalt proporcionan al **usuario** más **control** sobre las tareas que realiza comparadas con las GUIs que se construyen sin estos principios.

H_{1.B3}: Las GUIs construidas bajo los principios de la teoría Gestalt manejan **más fácilmente los errores** que puede cometer el usuario comparadas con las GUI que se construyen sin estos principios.

H_{1.B4}: Las GUIs construidas bajo los principios de la teoría Gestalt tienen **mensajes más claros** comparadas con las GUIs que se construyen sin estos principios.

H_{1.B5}: Las GUIs construidas bajo los principios de la teoría Gestalt tienen **mensajes más útiles** comparadas con las GUIs que se construyen sin estos principios.

H_{1.B6}: Las GUIs construidas bajo los principios de la teoría Gestalt resultan más **fáciles de navegar** comparadas con las GUIs que se construyen sin estos principios.

TESIS TESIS TESIS TESIS TESIS

H_{1.B7}: Las GUIs construidas bajo los principios de la teoría Gestalt tienen tareas más fáciles de acceder debido a que la **cantidad de movimientos** es más adecuada, comparadas con las GUIs que se construyen sin estos principios.

H_{1.c}: Las GUI construidas bajo los principios de la teoría Gestalt son más **atractivas** comparadas con las GUI que se construyen sin estos principios.

H_{1.c1}: Las GUIs construidas bajo los principios de la teoría Gestalt están **organizadas** de manera más simple comparadas con las GUIs que se construyen sin estos principios.

H_{1.c2}: Las GUIs construidas bajo los principios de la teoría Gestalt resultan más **agradables debido a la combinación de colores** utilizados, comparadas con las GUIs que se construyen sin estos principios.

H_{1.c3}: Las GUIs construidas bajo los principios de la teoría Gestalt resultan más **agradables debido al color de fondo** utilizado, comparadas con las GUIs que se construyen sin estos principios.

H_{1.c4}: Las GUIs construidas bajo los principios de la teoría Gestalt tienen **símbolos e iconos más agradables**, comparadas con las GUIs que se construyen sin estos principios.

H₂: Las GUI construidas bajo los principios de la teoría Gestalt son más **fáciles de aprender** comparadas con las GUI que se construyen sin estos principios.

H_{2.A}: Las GUIs construidas bajo los principios de la teoría Gestalt ayudan a **recordar más fácilmente acciones anteriores** comparadas con las GUIs que se construyen sin estos principios.

H_{2.B}: Las GUIs construidas bajo los principios de la teoría Gestalt ayudan a **comprender como realizar las tareas más fácilmente** comparadas con las GUIs que se construyen sin estos principios.

TESIS TESIS TESIS TESIS TESIS

H_{2.c}: Las GUIs construidas bajo los principios de la teoría Gestalt ayudan a **recordar más fácilmente como acceder a las tareas** comparadas con las GUI que se construyen sin estos principios.

H_{2.d}: En las GUIs construidas bajo los principios de la teoría Gestalt la **información se encuentra más ordenada de manera jerárquica**, comparadas con las GUIs que se construyen sin estos principios.

H_{2.e}: En las GUIs construidas bajo los principios de la teoría Gestalt la **cantidad de elementos** utilizados resulta más adecuada, comparadas con las GUIs que se construyen sin estos principios.

H₃: Las GUI construidas bajo los principios de la teoría Gestalt son más **aceptadas por los usuarios** comparadas con las GUI que se construyen sin estos principios.

H_{3.A}: Las GUIs construidas bajo los principios de la teoría Gestalt proporcionan más **beneficios al usuario** comparadas con las GUIs que se construyen sin estos principios

H_{3.B}: Las GUIs construidas bajo los principios de la teoría Gestalt inspiran más **confianza al usuario** comparadas con las GUIs que se construyen sin estos principios

H_{3.c}: Las GUIs construidas bajo los principios de la teoría Gestalt **satisfacen mejor las necesidades de los usuarios** comparadas con las GUIs que se construyen sin estos principios

TESIS TESIS TESIS TESIS TESIS

Capítulo 2

Marco Teórico

2.1 Calidad

En nuestro entorno todos los días nos vemos rodeados de publicidad, la cual trata de llamar nuestra atención y despertar el interés por algún producto o servicio. Por ejemplo, a través de mensajes subliminales, los cuales llegan a nuestra mente sin ser percibidos como tales, y tienen la finalidad de estimular al individuo para comprar. Los consumidores adquieren productos y servicios para satisfacer por medio de éstos alguna necesidad. De acuerdo con Juran (1996) la satisfacción del cliente es el resultado alcanzado cuando las características del producto responden a las necesidades del cliente; generalmente es sinónimo de satisfacción con el producto, de ésta manera, dicho producto podría ser percibido como de calidad. Asimismo, explica que si un cliente se siente satisfecho con un producto continuará adquiriéndolo. La insatisfacción con un producto tiene su origen en las no-conformidades con las características definidas para el mismo, y es debido a esto que los clientes pueden presentar quejas al respecto. De aquí que éstos consideren que el producto carece de calidad.

Los productores emplean como una estrategia competitiva el factor calidad ya que puede permitir el establecimiento de un punto de diferenciación entre las opciones disponibles.

La calidad no es un concepto nuevo, según expone Pyzdek (2003) la humanidad se ha preocupado por ella a lo largo de la historia. Desde el año 1760 A.C. en el

TESIS TESIS TESIS TESIS TESIS

código de Hammurabi ya se establecían leyes que buscaban la obtención de productos de calidad. Por ejemplo, se les daba la pena de muerte a los constructores cuyos edificios se derrumbaran y mataran a sus ocupantes. Lo anterior buscaba que los constructores pusieran el máximo empeño en su trabajo para producir edificios de alta calidad. Según lo relatado por Groocock (2005) en la edad media solo se podía considerar que una obra era perfecta si en ella no se encontraban errores. Se podía decir que calidad era sinónimo de excelencia o perfección.

Si bien es cierto que la importancia de la calidad tiene una aceptación universal, no se puede decir lo mismo de su definición, la cual causa controversia. Existen varias definiciones de dicho concepto entre las que destacan las enunciadas por algunas de las personas más notables en éste ámbito. Por ejemplo Pyzdek (2003) compiló una serie de definiciones de varios autores al respecto, entre ellas existen las siguientes: Crosby sostiene que “la calidad es el cumplimiento de los requisitos” (pág. 131), esto desde el enfoque de elaboración. Adicionalmente, Deming define calidad como “satisfacción del cliente” (pág. 131), dicha definición planteada bajo la perspectiva de quien hace uso del producto o servicio. Por otro lado Groocock (2005) explica que Juran la plantea como “idoneidad o aptitud para su uso” (pág. 27), esto desde el punto de vista desde el cual el usuario percibe las características del producto como beneficiosas para él.

La calidad puede obtener connotaciones diferentes según la persona que haga uso de ella. Pero, siempre permanece la idea central de que la calidad de un producto o servicio se aprecia como satisfactoria si responde a las necesidades del consumidor (Hansen & Ghare, 1990).

En un intento por contar con una definición más amplia, Groocock (2005) propone la siguiente definición en base a las propuestas de Crosby y Juran, agregando otros aspectos importantes:

“La calidad de un producto es el grado de conformidad de todas las propiedades y características pertinentes del producto relativas a todos los aspectos de la

TESIS TESIS TESIS TESIS TESIS

necesidad de un cliente, limitado por el precio y la entrega que él o ella está dispuesto a aceptar” (pág. 31).

Aunque en ésta definición se considere el precio como una limitante, puede no serlo completamente. Existe una famosa frase del productor Walt Disney que dice: Yo no me preocupo si algo es caro o barato. Solo me preocupa si es bueno. Si algo es suficientemente bueno, el público pagará por ello (Uvigo, 2005). De aquí, que podemos decir que si el producto o servicio es percibido por el cliente como “suficientemente bueno”, el precio no será un aspecto determinante para adquirirlo.

Como se puede apreciar, el tema de la satisfacción del cliente se encuentra ampliamente remarcado. El hecho de que la calidad sea juzgada bajo la percepción de una persona puede volverla subjetiva, ya que será apreciada según el criterio de cada cliente, lo que puede variar considerablemente de persona a persona.

En conclusión, es importante mencionar que la calidad de cualquier producto o servicio es vital para su éxito (Grocock, 2005), por lo cual debe ser considerado como un factor crítico.

“Al analizar las definiciones e implicaciones de la calidad para distintos autores, se puede establecer un criterio de selección basado en las acepciones del término tales como perfeccionamiento, mejora, tendencia hacia la excelencia, estandarización y la característica de poder ser medible” (Montaudon, 2004, pág. 12).

La calidad, en muchas ocasiones, se puede medir de forma directa, es decir, está relacionada con un atributo del producto o servicio. Por ejemplo, la resistencia en una construcción de concreto es tangible, se puede obtener directamente, y expresar de manera numérica. Pero no en pocas la calidad se mide de manera subjetiva. Por ejemplo, acerca del diseño de un mueble de madera solo se puede emitir una opinión, es decir, depende de la percepción de quien lo juzga. Es en

este último caso en el cual es muy difícil que los usuarios finales se pongan de acuerdo en el valor de calidad de un producto o servicio en particular. Situación que afecta grandemente a los productores. Entonces, ¿cómo hacer para reducir la subjetividad y producir un producto con calidad? Una posible respuesta es el desarrollo de dicho producto siguiendo estándares de calidad, los cuales nos dan una guía inicial del valor mínimo de cada atributo.

En conclusión, se puede afirmar que la calidad como factor crítico debe estar presente en todo producto o servicio si se desea lograr la satisfacción del cliente o persona que lo consume.

2.2 Estándares

Según Juran (1990) un estándar de calidad es un modelo, el cuál es obligatorio seguir y generalmente han sido probados: se ha demostrado que se pueden cumplir y tienden a permanecer por largo tiempo.

En general, un estándar es definido como aquel que sirve como tipo, modelo, norma, patrón o referencia (Real Academia, 2001). El objetivo de los estándares es lograr que las cosas sean más fáciles. Existen dos tipos de estándares: los llamados estándares de “facto” y los estándares de “jure”. Los primeros no son reconocidos de manera legal u oficial por alguna autoridad sino por el uso actual, éste tiene el reconocimiento público y es percibido como una garantía (Martínez, 2007). En cambio, los estándares de “jure” cuentan con un estatus legal, es decir, tienen reconocimiento oficial, la ley determina la estandarización para que un producto o servicio pueda ser conocido, y si no cuenta con el estándar impuesto legalmente el producto o servicio no será certificado (Martínez, 2007). Sea de “facto” o de “jure”, un estándar busca a través de una serie de conocimientos y prácticas reducir el número de errores que se cometen, lo cual es una práctica importante en cualquier proceso.

Existen diferentes organismos que promueven el desarrollo de normas o estándares internacionales de fabricación, comercio y comunicación para

TESIS TESIS TESIS TESIS TESIS

diferentes sectores. Algunos ejemplos de ellas son el International Electrotechnical Commission (IEC), el Institute of Electrical and Electronics Engineers (IEEE), el International Telecommunication Union (ITU), el American National Standards Institute (ANSI) y el International Organization for Standardization (ISO).

De los organismos mencionados, el ISO sobresale como uno de los más importantes proveedores de estándares de calidad para el área de tecnología (Lewis, 2009). Dicho organismo fue fundado en 1946 con el fin de facilitar el comercio internacional, la coordinación y la unificación de las normas industriales, proporcionando un único conjunto de normas que serían reconocidas y respetadas (Chua & Dyson, 2004). Dentro de las normas pertenecientes a dicha organización, se encuentra la norma ISO 9000, la cual se orienta hacia los procesos de gestión de calidad (Somerville, 2006) de cualquier tipo de organización y se compone de ISO 9001, 9002, 9003 y 9004 (Lewis, 2009).

Algunos de los estándares orientados hacia la calidad del software, pertenecientes a la familia ISO son: ISO/IEC 9126, ISO/IEC 14598 (ISO/IEC, 2005), ISO 9241-11. Sommerville (2006) argumenta que en el ámbito del desarrollo de software, algunas personas piensan que la calidad se puede lograr definiendo estándares y procedimientos organizacionales de calidad que comprueban si estos estándares son seguidos por el equipo de desarrollo. Asimismo, proporciona un argumento de que los estándares deben encapsular las buenas prácticas, las cuales nos llevan inevitablemente a productos de alta calidad. Adicionalmente, explica que mientras estándares y procedimientos son las bases de la gestión de la calidad, los gestores de calidad experimentados reconocen que hay aspectos intangibles en la calidad del software (i.e. elegancia, legibilidad, etc.) que no pueden ser incorporados en los estándares.

Si bien es cierto que los estándares intentan reunir un conjunto de buenas prácticas, el hecho de seguirlos no garantiza de manera total la obtención de productos de alta calidad.

Según Somerville (2006), existen dos tipos de estándares, los cuales forman parte del proceso de garantía de calidad. Estos son:

- **Estándares de Producto.** Estos se aplican sobre el producto de software que se comienza a desarrollar. Incluyen estándares de documentación, como cabecera de comentarios estándar para definición de clases, y estándares de codificación, que definen cómo debe utilizarse el lenguaje de programación.
- **Estándares de proceso.** Estos definen los procesos que deben seguirse durante el desarrollo del software. Pueden incluir definiciones de procesos de especificación, diseño y validación, así como una descripción de los documentos que deben escribirse en el curso de estos procesos.

Khan et al. (2008) explican que la utilización de estándares no necesariamente representa una ventaja, pero si una parte necesaria en operaciones industriales y en el logro de la satisfacción del cliente.

Siendo el desarrollo de software una tarea compleja el empleo de estándares también puede llegar a serlo dado que pueden ser percibidos como procesos burocráticos y tediosos, dejando de lado sus beneficios reales.

Dada la relevancia que tiene la calidad de un producto, es importante el uso de los estándares, ya que éstos nos proporcionan una guía para el logro de dicha calidad.

Es importante agregar que los estándares definen que es lo que se debe lograr, mas nunca explican el cómo lograrlo, siendo esto una debilidad de los mismos. La presente investigación intenta resolver parte de este problema.

2.3 Calidad de software

Según Pressman (2007), la calidad del software “es el cumplimiento de los requisitos de funcionalidad y desempeño explícitamente establecidos, de los

estándares de desarrollo explícitamente documentados y de las características implícitas que se esperan de todo software desarrollado profesionalmente” (pág. 463). Por otra parte, la IEEE la considera “como el grado en el que el software posee una combinación claramente definida y deseable de atributos de calidad” (Piattini & García, 2008, pág. 37).

Somerville (2006) expone que la calidad del software no es directamente comparable con la calidad de un producto de manufactura, ya que el software cuenta con características y especificaciones distintas a las de un producto manufacturado. La actividad de desarrollar software implica un proceso creativo donde intervienen las habilidades humanas y afectan factores externos. A pesar de lo anterior, también argumenta que la calidad del software se ha mejorado significativamente desde la década de los 90s y una de las razones para ello, ha sido que las compañías han adoptado nuevas técnicas y tecnologías. Por ejemplo, el uso de desarrollo orientado a objetos y el soporte asociado a través de herramientas CASE (Computer Aided Software Engineering). No obstante, también ha habido una mayor conciencia de la importancia de la gestión de la calidad y de la adopción de éstas técnicas para desarrollo en la industria del software.

Hablar de calidad de software implica hablar de calidad en el proceso y calidad en el producto, conceptos que se encuentran compleja y estrechamente relacionados. De acuerdo con Somerville (2006), la experiencia ha mostrado que la calidad del proceso tiene una influencia significativa en el producto. Aunque no necesariamente tener un proceso de calidad implica obtener un producto con calidad. La calidad del software es una compleja combinación de factores que variarán entre diferentes aplicaciones y los distintos clientes que las solicitan (Pressman, 2007).

Como el ciclo de vida útil del software puede llegar a ser largo y este estar cambiando continuamente debido a nuevos requerimientos; la calidad puede ser difícil de lograr (Amasaki, Yoshitomi, Mizuno, Takagi, & Kikuno, 2005).

Indiscutiblemente, el aseguramiento de la calidad del software es importante, no solamente por la satisfacción del usuario de acuerdo a sus requerimientos, sino también por la reducción de costos de mantenimiento (Andersson & Aimo, 1992).

Vega (2008) explica que existen estudios realizados que muestran que un alto porcentaje del éxito o fracaso de un proyecto está en que se lleve a cabo un adecuado control de la calidad. Los datos del Standish Group International muestran que un 25% de los proyectos de software son cancelados por atraso o por salirse del presupuesto, o tener una baja calidad, o por experimentar alguna combinación de ambos. Por otra parte, Howles (2003) argumenta que en la actualidad, y a pesar de los esfuerzos, la industria del software no puede garantizar ampliamente productos de calidad ya que frecuentemente existe en el mercado software deficiente que no satisface las necesidades de los clientes, lo que puede repercutir en pérdidas económicas. Además, se estima que el impacto económico de los defectos del software en Los Estados Unidos representa alrededor del 1% del producto interno bruto, lo cual da una idea clara de esta situación. Por cifras como esta muchos sectores de la industria no confían en la introducción de software como componente de sus productos. Lo cierto es que para lograr la calidad del software aún falta mucho por recorrer.

Según Pressman “el software de computadora es la tecnología más importante en el ámbito mundial” (2007, pág. 1), por lo que su calidad resulta ser un aspecto claramente relevante, lo que Vega (2008) cataloga como una necesidad.

De acuerdo con Somerville (2006), para tener como resultado un software de alta calidad es necesario, primordialmente, que el concepto de éste sea definido previamente, lo que se intenta lograr con un plan de calidad, el cual selecciona los estándares adecuados para el proyecto, vigila el proceso de desarrollo para asegurar que se sigan los procedimientos y los estándares de garantía de calidad; y finalmente, evalúa la calidad obtenida en el proceso y en el producto. “La función primaria del aseguramiento de la calidad es determinar si las necesidades de los usuarios están siendo satisfechas adecuadamente” (Vega, 2008, pág. 14).

Adicionalmente, el axioma fundamental de la calidad de un producto es que las características tangibles internas o propiedades de un producto determinan sus atributos de calidad externa (Ortega, 2003).

La preocupación por la calidad en el área del software ha llevado a buscar diversas alternativas para lograrla. Por ejemplo, las guías o puntos de referencia de los cuales se pueden auxiliar los desarrolladores. Como parte de estas alternativas han surgido los procesos, estándares y modelos.

2.3.1 Modelos de calidad de software

Según Ortega (2003) entre los modelos más conocidos dentro de la industria del software están: McCall, ISO 9126, FURPS, Boehm y Dromey. Khan et al. (2008) explica que la mayoría de los modelos anteriores están basados en el Factor-Criteria-Metrics (FCM), el cual es un modelo generalmente aceptado como una base para la medición de software. El principio básico de este modelo es que cada atributo puede ser dividido en un grupo de factores, los cuales a su vez pueden ser divididos en un conjunto de criterios, y expresa su idea de ésta manera: “FCM es un típico modelo jerárquico de métricas de calidad” (pág. 105).

Las siguientes secciones describen más a detalle los modelos mencionados previamente.

2.3.1.1 Modelo de McCall

Entre los primeros trabajos realizados referentes a la calidad del software se encuentra la clasificación de los factores propuesto por McCall, Richards y Walters en 1977. Khan et al. (2008) explican que el modelo de McCall fue un proyecto conjunto entre la US Air Force Electronic System Decision (ESD), la Rome Air Development Center (RADC) y General Electric (GE); creado con la finalidad de mejorar y aumentar la calidad del producto, y expresa que “el objetivo básico del modelo de calidad de McCall (1977) es evaluar la relación entre los factores de calidad externos y los criterios de calidad del producto” (pág. 106).

El modelo de McCall (1977) (ver Figura 1) establece tres áreas principales que intervienen en la calidad del software (Dávila & Mejía, 2003), las cuales se describen a continuación:

1. Calidad en la operación del producto.

Requiere que el software pueda ser entendido fácilmente, que opere eficientemente y que los resultados obtenidos sean los requeridos inicialmente por el usuario.

2. Revisión de la calidad del producto de software.

Tiene como objetivo realizar revisiones durante el proceso de desarrollo para detectar los errores que afecten a la operación del producto.

3. Transición del producto.

Requiere de la definición de estándares y procedimientos que sirvan como base para el desarrollo del software.

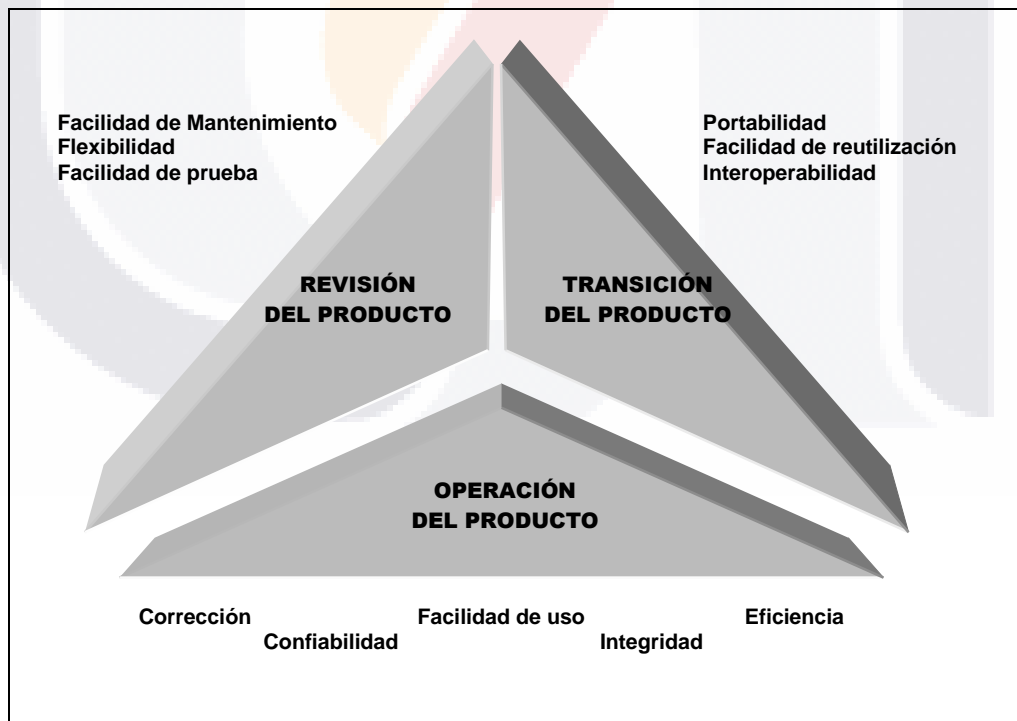


Figura 1. Modelo de McCall (Adaptado de Pressman, 2007, pág. 464)

Literatura previa (Khan, Mustafa, & Ahson, 2008) describe que el modelo de McCall distingue dos niveles de características de calidad: factores y criterios. Los criterios, a su vez, tienen una serie de métricas para ser evaluados.

A continuación se explican los términos anteriores tomado de los creadores (McCall A., Richards, & Walters, 1977) mismos:

Factor: Es una condición o característica que contribuye activamente a la la calidad del software. Para propósitos de normalización, todos los factores deben estar relacionados con un costo normalizado o bien realizar la actividad caracterizada por el factor o para operar con el grado de calidad.

Por ejemplo, el mantenimiento es el esfuerzo necesario para localizar y corregir un error en un programa. Este esfuerzo requerido puede ser expresado en unidades de tiempo, monetarias, o mano de obra. Se utilizaron las siguientes normas para determinarlos:

- Una condición o característica que contribuye a la calidad del software,
- Una característica relacionada con el usuario,
- Relacionado con los costos ya sea para realizar la actividad o para operar con ese grado de calidad,
- Característica relativa entre los productos de software.

Criterio: Es un atributo del software o de los procesos de producción del mismo por el cual los factores pueden ser juzgados y definidos. Las siguientes reglas se aplicaron para la determinación:

- Atributos del software o productos del proceso de desarrollo de software, es decir, los criterios son orientado al software mientras que los factores están orientadas al usuario,
- Puede mostrar una relación jerárquica con subcriterios,
- Puede afectar a más de un factor.

Métrica: Son las medidas de los criterios o subcriterios relacionados con la los factores de calidad. Estas pueden ser objetivas o subjetivas. Las unidades de mediciones son elegidas como la proporción de casos reales para el mayor número de casos posible.

Los factores definidos para el modelo se describen a continuación (Cavano & McCall, 1978):

Corrección: Es la medida en que un programa cumple con sus especificaciones y satisface los objetivos propuestos por el cliente.

Fiabilidad: Es el grado en el que se espera que un programa desempeñe su función con la precisión requerida.

Eficiencia: Es la cantidad de código y de recursos de cómputo necesarios para que un programa realice su función.

Integridad: Es el grado de control sobre el acceso al software o los datos por parte de personas no autorizadas.

Facilidad de uso: Es el esfuerzo necesario para aprender, operar y preparar los datos de entrada de un programa e interpretar la salida.

Facilidad de mantenimiento: Es el esfuerzo necesario para localizar y corregir un error en un programa.

Flexibilidad: Es el esfuerzo necesario para modificar un programa en operación.

Facilidad de prueba: Es el esfuerzo que demanda probar un programa con el fin de asegurar que realiza su función.

Portabilidad: Es el esfuerzo necesario para transferir el programa de un entorno de hardware o software a otro.

Facilidad de reutilización: Es la medida en que un programa puede ser utilizado en otras aplicaciones relacionadas, y el alcance de las funciones que realiza el programa.

Interoperabilidad: Es el esfuerzo necesario para acoplar un sistema con otro.

Estos factores, a su vez, se descomponen en una serie de 23 criterios que sirven de guía a los desarrolladores de software (ver Figura 2). Por último, se proponen un total de 41 métricas para evaluar estos criterios (McCall A., Richards, & Walters, 1977).

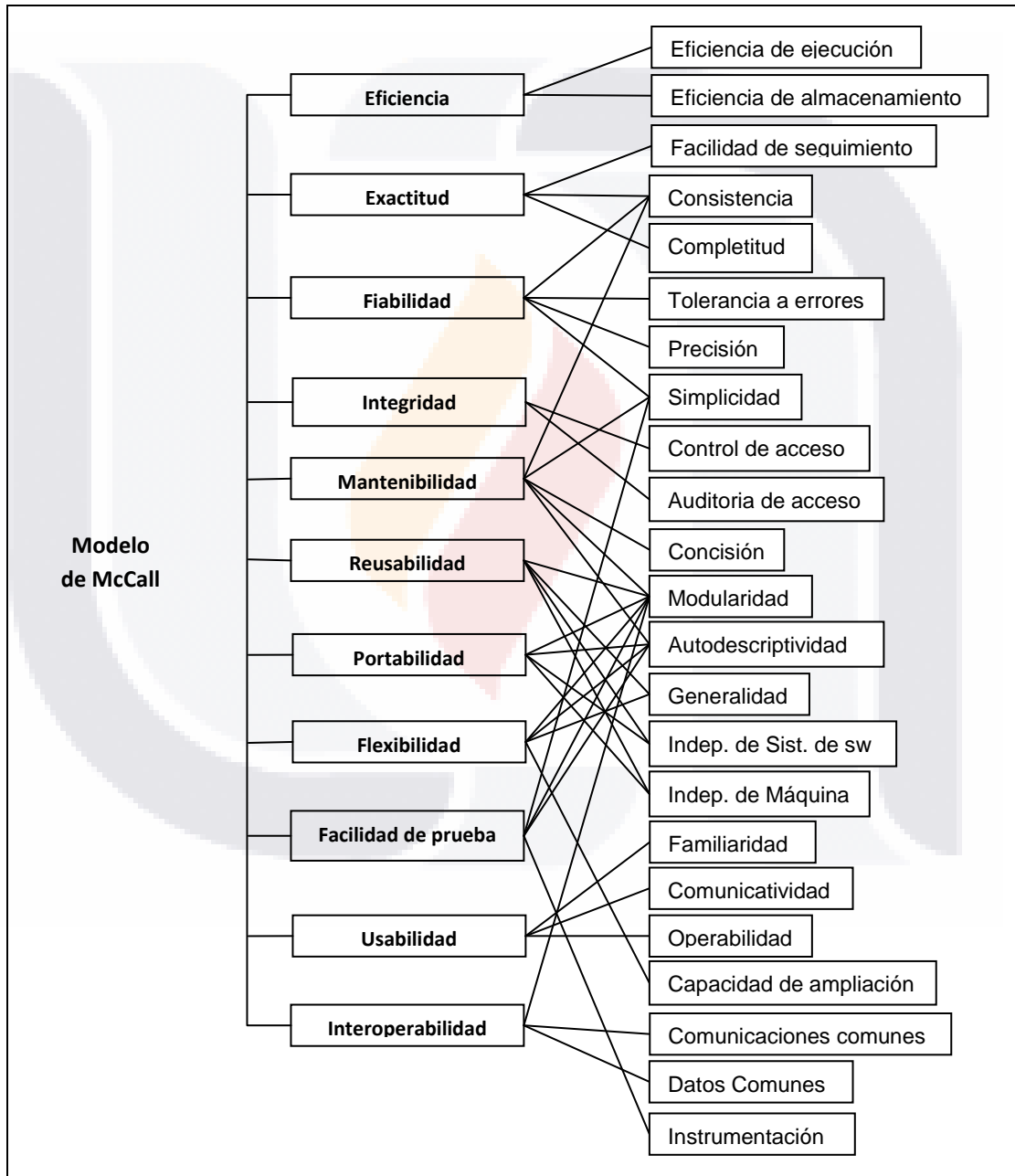


Figura 2. Modelo de McCall Factores y Criterios, (Adaptado de McCall A., Richards, & Walters, 1977).

A continuación se describen cada una de las 41 métricas para los diferentes criterios que propone el modelo (ver Tabla 1).

Facilidad de seguimiento	
<i>Métrica 1</i>	<p>Referencia cruzada relativa a los módulos de requisitos. La métrica es el número de requisitos diseñados detallados dividido por el número total de los requisitos detallados.</p> $SMV = \frac{\text{Número de Requisitos diseñados detallados}}{\text{Número total de los requisitos detallados}}$
Compleitud	
<i>Métrica 2</i>	<p>Lista de comprobación de Compleitud. La métrica es la suma de las puntuaciones de los elementos aplicables dividido por el número de elementos aplicables.</p> $SMV = \frac{\sum_1^9 \text{Puntuación por elemento}}{9}$
Consistencia	
<i>Métrica 3</i>	<p>Medida de consistencia del procedimiento. La métrica es la suma de las puntuaciones de los elementos aplicables dividido por el número de elementos aplicables.</p> $SMV = \frac{\sum \text{Puntuación de elementos aplicables}}{\text{Número total de elementos aplicables}}$
<i>Métrica 4</i>	<p>Medida de consistencia de los datos. La métrica es la suma de las puntuaciones de los elementos aplicables dividido por el número de elementos aplicables.</p> $SMV = \frac{\sum \text{Puntuación de elementos aplicables}}{\text{Número total de elementos aplicables}}$
Precisión	
<i>Métrica 5</i>	<p>Lista de comprobación de Precisión. Cada elemento es una medida binaria que indica la existencia, o ausencia de los elementos. La métrica es la suma de las puntuaciones de los elementos aplicables dividido por el número de elementos aplicables</p> $SMV = \frac{\sum \text{Puntuación de elementos aplicables}}{\text{Número total de elementos aplicables}}$
Tolerancia a fallas	
<i>Métrica 6</i>	<p>Lista de verificación de Tolerancia a fallas. La métrica es la suma de las puntuaciones de los elementos aplicables dividido por el número de elementos aplicables.</p> $SMV = \frac{\sum \text{Puntuación de elementos aplicables}}{\text{Número total de elementos aplicables}}$

Métrica 7	<p>Recuperación de la lista de comprobación de datos de entrada incorrectos. La métrica es la suma de las puntuaciones de los elementos aplicables dividido por el número de elementos aplicables.</p> $SMV = \frac{\sum \text{Puntuación de elementos aplicables}}{\text{Número total de elementos aplicables}}$
Métrica 8	<p>Recuperación de la lista de comprobación de fallas computacionales. La métrica es la suma de las puntuaciones de los elementos aplicables dividido por el número de elementos aplicables.</p> $SMV = \frac{\sum \text{Puntuación de elementos aplicables}}{\text{Número total de elementos aplicables}}$
Métrica 9	<p>Recuperación de la lista de comprobación de fallas de hardware. La métrica es la suma de las puntuaciones de los elementos aplicables dividido por el número de elementos aplicables.</p> $SMV = \frac{\sum \text{Puntuación de elementos aplicables}}{\text{Número total de elementos aplicables}}$
Métrica 10	<p>Recuperación de la lista de comprobación de fallas de dispositivos. La métrica es la suma de las puntuaciones de los elementos aplicables dividido por el número de elementos aplicables.</p> $SMV = \frac{\sum \text{Puntuación de elementos aplicables}}{\text{Número total de elementos aplicables}}$
Simplicidad	
Métrica 11	<p>Medida de simplicidad del diseño. La métrica es la suma de las puntuaciones de los elementos aplicables dividido por el número de elementos aplicables.</p> $SMV = \frac{\sum \text{Puntuación de elementos aplicables}}{\text{Número total de elementos aplicables}}$
Métrica 12	<p>Uso de lenguaje estructurado o preprocesador de lenguaje estructurado. La métrica es una medida binaria de la existencia (1) o ausencia (0) del código de lenguaje estructurado.</p> $SMV = \frac{\sum_1^n \text{Puntuación de los módulos}}{\text{Número de módulos}}$
Métrica 13	<p>Medida de la complejidad de control de flujo de datos. La métrica se puede obtener desde la representación de diseño (por ejemplo, diagramas de flujo) y el código de forma automática.</p> $SMV = \frac{\sum \text{Complejidad de las medidas por cada módulo}}{\text{Número total de módulos}}$
Métrica 14	<p>Medida de la simplicidad de técnicas de codificación. La métrica es una cantidad promedio de todas las medidas de módulo para el sistema. La medida del módulo es la suma de las puntuaciones de los elementos aplicables dividido por el número de elementos aplicables.</p> $SMV = \frac{\sum \text{Medidas de simplicidad del código de cada módulo}}{\text{Número total de módulos}}$ $MMV = \frac{\sum \text{Puntuación de elementos aplicables}}{\text{Número total de elementos aplicables}}$
Modularidad	

Métrica 15	<p>Medida de estabilidad. Categorización de los módulos por su fuerza y acoplamiento. El módulo de la fuerza es una medida de la cohesión o la relación de los elementos dentro de un módulo. Módulo de acoplamiento es una medida de la relación entre los módulos. La métrica combina estas dos medidas para calcular el número esperado de los módulos que requieren modificar si se han hecho cambios en cualquier módulo, dividido por el número total de módulos.</p> $SMV = \frac{\text{Número esperado de módulos cambiados}}{\text{Número total de módulos}}$
Métrica 16	<p>Medida de la implementación modular. La métrica es la suma de las puntuaciones de los elementos aplicables, divididos por el número de elementos aplicables.</p> $SMV = \frac{\sum \text{Puntuación de elementos aplicables}}{\text{Número total de elementos aplicables}}$
Generalidad	
Métrica 17	<p>Medida en la que los módulos son referenciados por otros módulos. La métrica proporciona una medida de la generalidad de los módulos que se utilizan en el sistema actual. Un módulo se considera de carácter más general, si se utiliza (referencia) por más de un módulo. El número de estos módulos comunes dividido por el número total de módulos que proporciona la métrica.</p> $SMV = \frac{\text{Número de módulos comunes}}{\text{Número total de módulos}}$
Métrica 18	<p>Medida de la generalidad de la implementación. La métrica es la suma de las puntuaciones de los elementos aplicables dividido por el número de elementos aplicables.</p> $SMV = \frac{\sum \text{Puntuación de elementos aplicables}}{\text{Número total de elementos aplicables}}$
Capacidad de ampliación	
Métrica 19	<p>Medida de expansión de almacenamiento de datos. La métrica es la suma de las puntuaciones de los elementos aplicables dividido por el número de elementos aplicables.</p> $SMV = \frac{\sum \text{Puntuación de elementos aplicables}}{\text{Número total de elementos aplicables}}$
Métrica 20	<p>Medida de extensibilidad. La métrica es la suma de las puntuaciones de los elementos aplicables dividido por el número de elementos aplicables.</p> $SMV = \frac{\sum \text{Puntuación de elementos aplicables}}{\text{Número total de elementos aplicables}}$
Instrumentación	
Métrica 21	<p>Medida de prueba de módulos. La métrica es un promedio de todas las medidas del módulo. La medida del módulo es el puntaje promedio de los dos elementos.</p> $MMV = \frac{\sum \text{Puntuación de elementos aplicables}}{\text{Número total de elementos aplicables}}$ $SMV = \frac{\sum \text{Medidas de prueba de los módulos por cada módulo}}{\text{Número total de módulos}}$

Métrica 22	<p>Medida de prueba de integración. La métrica es el puntaje promedio de los elementos.</p> $SMV = \frac{\sum \text{Puntuación de elementos aplicables}}{\text{Número total de elementos aplicables}}$
Métrica 23	<p>Medida de prueba del sistema La métrica es el puntaje promedio de los elementos.</p> $SMV = \frac{\sum \text{Puntuación de elementos aplicables}}{\text{Número total de elementos aplicables}}$
Autodescriptividad	
Métrica 24	<p>Cantidad de comentarios. La métrica es el número de líneas de comentarios, dividido por el número total de líneas en cada módulo. Las líneas en blanco no se cuentan. El valor medio se calcula para el nivel del sistema métrico.</p> $SMV = \frac{\sum \text{Cantidad de comentarios por cada módulo}}{\text{Número total de módulos}}$
Métrica 25	<p>Medida de efectividad de los comentarios. La métrica es la suma de las puntuaciones de los elementos aplicables dividido por el número de elementos aplicables.</p> $SMV = \frac{\sum \text{Puntuación de elementos aplicables}}{\text{Número total de elementos aplicables}}$
Métrica 26	<p>Medida de descriptividad del lenguaje de implementación. La métrica es la suma de las puntuaciones de los elementos aplicables dividido por el número de elementos aplicables.</p> $SMV = \frac{\sum \text{Puntuación de elementos aplicables}}{\text{Número total de elementos aplicables}}$
Eficiencia de ejecución	
Métrica 27	<p>Requisitos de desempeño asignados al diseño. La métrica sólo identifica si los requisitos de rendimiento se han asignados durante el diseño (1) o no (0).</p>
Métrica 28	<p>Medida de eficiencia del proceso iterativo. La métrica es la suma de las puntuaciones de los elementos aplicables dividido por el número de elementos aplicables.</p> $MMV = \frac{\sum \text{Puntuación de elementos aplicables}}{\text{Número total de elementos aplicables}}$ $SMV = \frac{\sum \text{Medidas de proceso iterativo por cada módulo}}{\text{Número total de módulos}}$
Métrica 29	<p>Medida de eficiencia del uso de datos. La métrica es la suma de las puntuaciones de los elementos aplicables dividido por el número de elementos aplicables.</p> $MMV = \frac{\sum \text{Puntuación de elementos aplicables}}{\text{Número total de elementos aplicables}}$ $SMV = \frac{\sum \text{Medidas del uso de datos por cada elemento}}{\text{Número total de módulos}}$
Eficiencia de almacenamiento	

Métrica 30	<p>Medida de la eficiencia de almacenamiento. La métrica es la suma de las puntuaciones de los elementos aplicables dividido por el número de elementos aplicables.</p> $MMV = \frac{\sum \text{Puntuación de elementos aplicables}}{\text{Número total de elementos aplicables}}$ $SMV = \frac{\sum \text{Medidas de eficiencia de almacenamiento por cada elemento}}{\text{Número total de módulos}}$
Control de acceso	
Métrica 31	<p>Lista de verificación de control de acceso. La métrica es la suma de las puntuaciones de los elementos aplicables dividido por el número de elementos aplicables.</p> $SMV = \frac{\sum \text{Puntuación de elementos aplicables}}{\text{Número total de elementos aplicables}}$
Auditoria de acceso	
Métrica 32	<p>Lista de verificación de auditoría de acceso. La métrica es el puntaje promedio de los siguientes dos elementos.</p> $SMV = \frac{\sum \text{Puntuación de elementos aplicables}}{\text{Número total de elementos aplicables}}$
Operabilidad	
Métrica 33	<p>Lista de verificación de operabilidad. La métrica es la suma de las puntuaciones de los elementos aplicables dividido por el número de elementos aplicables.</p> $SMV = \frac{\sum \text{Puntuación de elementos aplicables}}{\text{Número total de elementos aplicables}}$
Entrenamiento	
Métrica 34	<p>Lista de verificación de entrenamiento. La métrica es la suma de las puntuaciones de los elementos aplicables dividido por el número de elementos aplicables.</p> $SMV = \frac{\sum \text{Puntuación de elementos aplicables}}{\text{Número total de elementos aplicables}}$
Comunicatividad	
Métrica 35	<p>Medida de entradas de interfaz de usuario. La métrica es la suma de las puntuaciones de los elementos aplicables dividido por el número de elementos aplicables.</p> $SMV = \frac{\sum \text{Puntuación de elementos aplicables}}{\text{Número total de elementos aplicables}}$
Métrica 36	<p>Medida de salida de interfaz de usuario. La métrica es la suma de las puntuaciones de los elementos aplicables dividido por el número de elementos aplicables.</p> $SMV = \frac{\sum \text{Puntuación de elementos aplicables}}{\text{Número total de elementos aplicables}}$
Independencia del sistema de software	

<i>Métrica 37</i>	<p>Medida de la independencia del sistema de software. La métrica es la suma de las puntuaciones de los elementos aplicables dividido por el número de elementos aplicables.</p> $SMV = \frac{\sum \text{Puntuación de elementos aplicables}}{\text{Número total de elementos aplicables}}$
Independencia de máquina	
<i>Métrica 38</i>	<p>Medida de independencia de máquina. La métrica es la suma de las puntuaciones de los elementos aplicables dividido por el número de elementos aplicables.</p> $SMV = \frac{\sum \text{Puntuación de elementos aplicables}}{\text{Número total de elementos aplicables}}$
Comunicaciones comunes	
<i>Métrica 39</i>	<p>Lista de verificación de comunicaciones comunes. La métrica es la suma de las puntuaciones de los elementos aplicables dividido por el número de elementos aplicables.</p> $SMV = \frac{\sum \text{Puntuación de elementos aplicables}}{\text{Número total de elementos aplicables}}$
Datos comunes	
<i>Métrica 40</i>	<p>Lista de verificación de datos comunes. La métrica es la suma de las puntuaciones de los elementos aplicables dividido por el número de elementos aplicables.</p> $SMV = \frac{\sum \text{Puntuación de elementos aplicables}}{\text{Número total de elementos aplicables}}$
Concisión	
<i>Métrica 41</i>	<p>Medida de Halstead. La métrica se basa en el concepto de Halstead de longitud. La longitud observada de un módulo es: $N_0 = N_1 + N_2$ donde: N_1 Total de operadores usados en un módulo N_2 Total de operandos usados en un módulo La longitud calculada del módulo es: $N_c - n_1 \log_2 n_1 + n_2 \log_2 n_2$ N_1 = Número de operadores únicos en el módulo. La métrica está normalizada de la manera siguiente:</p> $1 - \frac{N_c - N_0}{N_0} \text{ ó si } \frac{N_c - N_0}{N_0} \text{ es mayor de 0 a 1}$ <p>la métrica es el valor promedio de todas las métricas de los valores del módulo.</p> $SMV = \frac{\sum \text{Medida de Halstead por cada módulo}}{\text{Número total de módulos}}$

Tabla 1. Criterios y métricas del modelo de McCall (Adaptado de McCall A., Richards, & Walters, 1977)

Al emplear este modelo para evaluar la calidad en el software es común que las métricas adquieran una lista de comprobación que se emplea para asignar una graduación a atributos específicos del software (Cavano & McCall, 1978).

Ortega (2003) argumenta que una de las mayores contribuciones del modelo de McCall es la relación creada entre las características de calidad y las métricas, aunque ha sido criticado porque no todas las métricas son objetivas.

2.3.1.2 Modelo ISO/IEC 9126

De acuerdo con Chua et al. (2004), el modelo ISO/IEC 9126 pretende proporcionar un marco de referencia para la evaluación de la calidad del software e identifica seis atributos clave.

Pressman (2007) explica que los factores que lo componen no necesariamente representan la medición directa, sin embargo, proporcionan una base valiosa para las medidas indirectas y una lista de verificación excelente para evaluar la calidad de un sistema.

De acuerdo con Khan et al. (2008), el estándar ISO/IEC 9126 resulta ser una derivación del modelo de McCall. Abran et al. (2003) consideran que aunque no se trata de un modelo exhaustivo, el ISO/IEC 9126 constituye el modelo de calidad de software más completo hasta la fecha. Dicho modelo comienza por definir el concepto de calidad de software como “la totalidad de las funciones y características de un producto de software que influyen en su capacidad para satisfacer necesidades explícitas o implícitas” (Khan, Mustafa, & Ahson, 2008, pág. 109).

De acuerdo con Minguet (2003), la primera edición de la norma ISO/IEC 9126 se realizó en 1991 con la finalidad de promover un entorno que permita la evaluación de la calidad del software. Posteriormente, en 1994 se realizó una nueva adaptación de la norma donde se introdujeron los conceptos de calidad interna y externa surgiendo también la norma ISO/IEC 14598, la cual asume el modelo del proceso de evaluación. La literatura (Abran, Khelifi, & Suryan, 2003) explica que

posteriormente, entre el 2001 y el 2003, aparece una nueva edición de la norma ISO IEC/9126, la cual está dividida en 4 partes (ver Figura 3):

- Parte 1: Modelo de la calidad
- Parte 2: Métricas internas
- Parte 3: Métricas externas
- Parte 4: Calidad en las métricas de uso

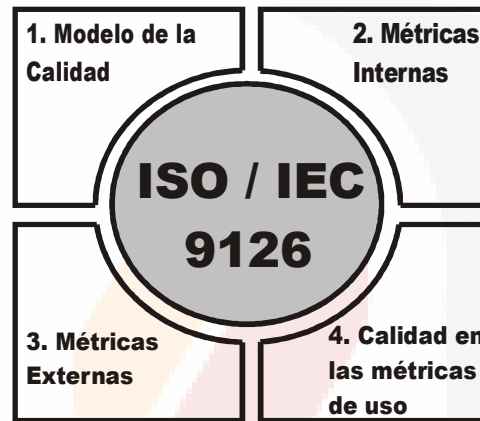


Figura 3. Modelo ISO/IEC 9126

A continuación se describen cada una de las partes del modelo:

1. ISO/IEC 9126-1: Modelo de Calidad.

La norma define un modelo de calidad basado en dos estructuras distintas:

- a) La calidad externa e interna es modelada con las siguientes características: funcionalidad, confiabilidad, efectividad, usabilidad, mantenibilidad y portabilidad.
- b) La calidad de uso es modelada por las características: efectividad, productividad, seguridad y satisfacción.

2. ISO/IEC 9126-2: Métricas externas

En esta parte se describen las medidas que pueden ser usadas para especificar o evaluar el comportamiento del software cuando es operado por el usuario.

3. ISO/IEC 9126-3: Métricas internas

En esta parte se describen las medidas que pueden ser usadas para crear los requerimientos que describen las propiedades estáticas de la interfaz, las cuales pueden ser evaluadas por inspección sin la funcionalidad del software.

4. ISO/IEC 9126-4: Calidad en las métricas de uso

En esta parte se describen las medidas que pueden ser usadas para especificar o evaluar el impacto del uso del software cuando es operado por el usuario.

El modelo ISO/IEC 9126 define un marco de referencia, en el cual se explican las relaciones entre diferentes enfoques de calidad (Khan, Mustafa, & Ahson, 2008) y se proporcionan las bases para su evaluación (Chua & Dyson, 2004).

ISO/IEC 9126 no establece requisitos específicos de calidad de software, sino que define un modelo de calidad, que puede ser aplicado a todo tipo de software. La norma incluye la perspectiva del usuario e introduce el concepto de calidad en el uso (Abran, Khelifi, & Suryn, 2003).

El estándar divide la calidad del software dentro de seis categorías de características (Abran, Khelifi, & Suryn, 2003), las cuales se muestran en la Figura 4.

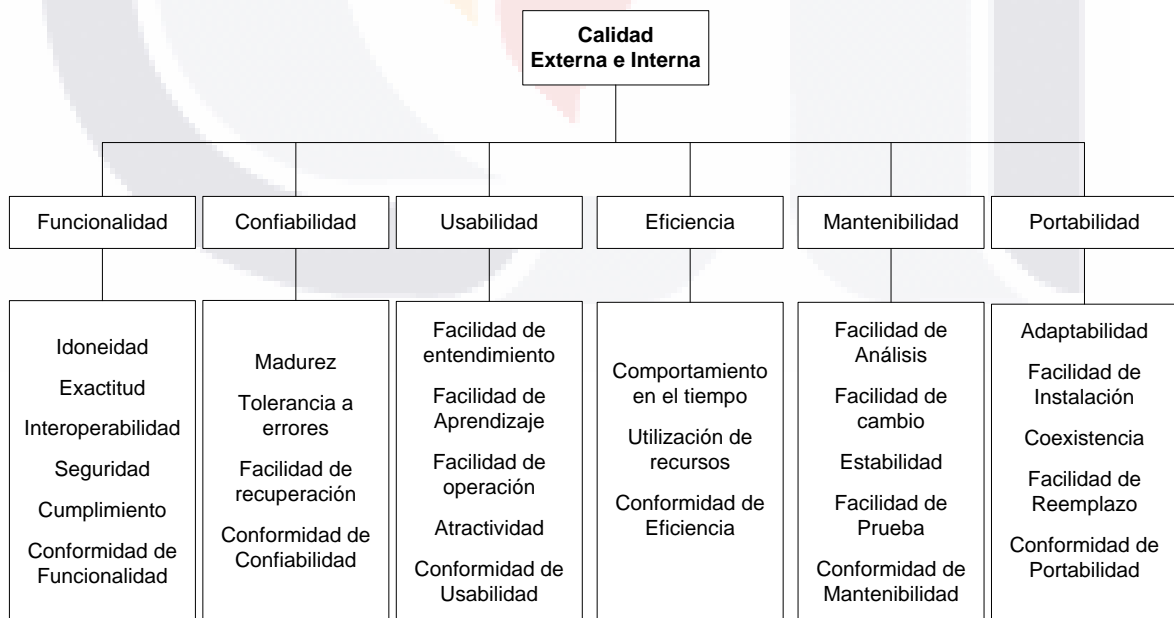


Figura 4. Factores de Calidad ISO/IEC 9126 (Adaptado de Abran, Khelifi, & Suryn, 2003)

Descripción de características según Pressman (2007):

Funcionalidad: El grado en que el software satisface las necesidades que indican los siguientes subatributos: idoneidad, exactitud, interoperabilidad, cumplimiento y seguridad.

Confiabilidad: La cantidad de tiempo en que el software está disponible para usarlo según los siguientes subatributos: madurez, tolerancia a fallas y facilidad de recuperación.

Facilidad de uso: La facilidad con que se usa el software de acuerdo con los siguientes subatributos: facilidad de comprensión, facilidad de aprendizaje y operabilidad.

Eficiencia: El grado en que el software emplea en forma óptima los recursos del sistema, como lo indican los siguientes subatributos: comportamiento en el tiempo, comportamiento de los recursos.

Facilidad de mantenimiento: La facilidad con que se repara el software de acuerdo con los siguientes subatributos: facilidad de análisis, facilidad de cambio, estabilidad y facilidad de prueba.

Portabilidad: La facilidad con que se lleva el software de un entorno a otro según los siguientes subatributos: adaptabilidad, facilidad para instalarse, cumplimiento, facilidad de reemplazarse.

En la Tabla 2 se explica cada concepto:

Característica	Sub-Característica	Explicación
Funcionalidad	Idoneidad	¿El software puede realizar las tareas necesarias?
	Exactitud	¿Es el resultado como se esperaba?
	Interoperabilidad	¿El sistema puede interactuar con otro sistema?
	Seguridad	¿El software prevenir el acceso no autorizado?
Confiabilidad	Madurez	¿La mayoría de las fallas del software han sido eliminadas en el transcurso del tiempo?
	Tolerancia a fallos	¿Es el software capaz de manejar los errores?
	Facilidad de recuperación	¿Puede el software de reanudar su trabajo y restaurar datos perdidos después de una falla?

Usabilidad	Facilidad de entendimiento	¿El usuario comprender cómo utilizar el sistema con facilidad?
	Facilidad de aprendizaje	¿Puede el usuario aprende a usar el sistema con facilidad?
	Facilidad de operación	¿El usuario puede utilizar el sistema sin mucho esfuerzo?
	Atractividad	¿Tiene la interfaz un buen aspecto?
Eficiencia	Comportamiento en el tiempo	¿Con qué rapidez el sistema responde?
	Comportamiento de los recursos	¿El sistema utiliza los recursos eficientemente?
Facilidad de mantenimiento	Facilidad de análisis	¿Pueden ser fácilmente diagnosticadas las fallas?
	Facilidad de cambio	¿El software puede ser fácilmente modificado?
	Estabilidad	¿El software puede seguir funcionando si se hacen cambios?
	Facilidad de prueba	¿El software puede ser probado fácilmente?
Portabilidad	Adaptabilidad	¿El software puede ser trasladado a otros entornos?
	Facilidad de instalación	¿El software puede ser instalado fácilmente?
	Cumplimiento	¿El software cumple con las normas de portabilidad?
	Facilidad de reemplazo	¿El software puede ser fácilmente reemplazado por otro software?
Todas las características	Conformidad	¿El software cumple con las leyes o reglamentos?

Tabla 2. ISO 9126-1 Características y sub-características (Adaptado de Chua & Dyson, 2004)

Existen varios estudios (Evan, 1989), (Botella, Burgués, Carvallo, Franch, Grau, & Quer, 2004), (Henderson, Smith, Podd, & Varela-Alvarez, 1995) que critican la norma ISO/IEC 9126 por no prescribir requisitos de calidad específicos, sino la definición de un marco general para la evaluación de la calidad de Software. Los modelos para evaluar la calidad del software proporcionan un marco de referencia que indica que evaluar, dejando de lado el hecho de cómo hacerlo, aspecto que puede representar flexibilidad por una parte e incertidumbre por otra.

En conclusión, los modelos de calidad son entendidos como “Un conjunto de conceptos medibles y relaciones entre ellos que proporciona la base para especificar requisitos de calidad y evaluar la calidad de las entidades de una

determinada categoría de entidad” (Piattini & García, 2008, pág. 74). En esta definición se resalta la naturaleza de ser medible.

Son muchos los factores involucrados en dichos modelos, algunos de ellos no son factibles de medir de manera directa, tal es el caso de la mantenibilidad, la comprensión y la usabilidad, los cuales son atributos externos que nos dicen cómo ven el software los desarrolladores y los usuarios. Éstos se ven afectados por diversos factores y no existe un camino simple para medirlos (Somerville, 2006).

Dolado y Fernández (2000) explican que en relación al software existen resultados con poca o débil credibilidad dado que rara vez son validados. Expone un dato que resulta interesante: para el año 2000 un tercio de los artículos publicados en IEEE no contenían ninguna validación experimental y otro tercio solo aportaba validaciones informales con poco valor. Asimismo asegura que la posibilidad de medir es el fundamento de las disciplinas científicas y de la ingeniería.

2.3.1.3 Modelo de FURPS

Modelo surgido en 1987 por Grady y Caswel. El modelo toma su nombre de los cinco aspectos que contempla principalmente: Funcionalidad (functionality), usabilidad (usability), fiabilidad (reliability), desempeño (performance) y compatibilidad (supportability) (Ortega, 2003). El modelo considera dos categorías:

1. Funcionales (F). Establecimiento de prioridades
2. No Funcionales (URPS). Definición de los atributos de calidad que se pueden medir

A continuación se describen brevemente de acuerdo Milicic (2005):

Funcionalidad: Incluye un conjunto de características, capacidades y seguridad.

Usabilidad: Incluye factores humanos, estética, coherencia en la interfaz de usuario, documentación de usuario y material de capacitación.

Fiabilidad: Incluye la frecuencia, recuperación, prevención y tiempo medio entre fallas.

Desempeño: Impone condiciones a los requisitos funcionales, tales como la velocidad, eficiencia, disponibilidad, precisión, tiempo de procesamiento y de respuesta, tiempo de recuperación y uso de los recursos.

Compatibilidad: incluye la capacidad de prueba, la extensibilidad, adaptabilidad, capacidad de mantenimiento, la compatibilidad, capacidad de configuración, capacidad de servicio y capacidad de instalación.

Ortega (2003) también explica que una de las desventajas de este modelo es que no toma en consideración la portabilidad de los productos de software.

2.3.1.4 Modelo de Bohem

Modelo desarrollado en 1978 por Bohem (ver Figura 5). Dicho modelo se presenta en forma jerárquica, resaltando por ser uno de los mejores definidos según literatura previa y Mejía (Dávila & Mejía, 2003). Los criterios de calidad se dividen en tres grandes áreas:

1. Portabilidad: servicios que el sistema ofrece.
2. Usabilidad: operación del producto
3. Mantenibilidad: mantenimiento del producto

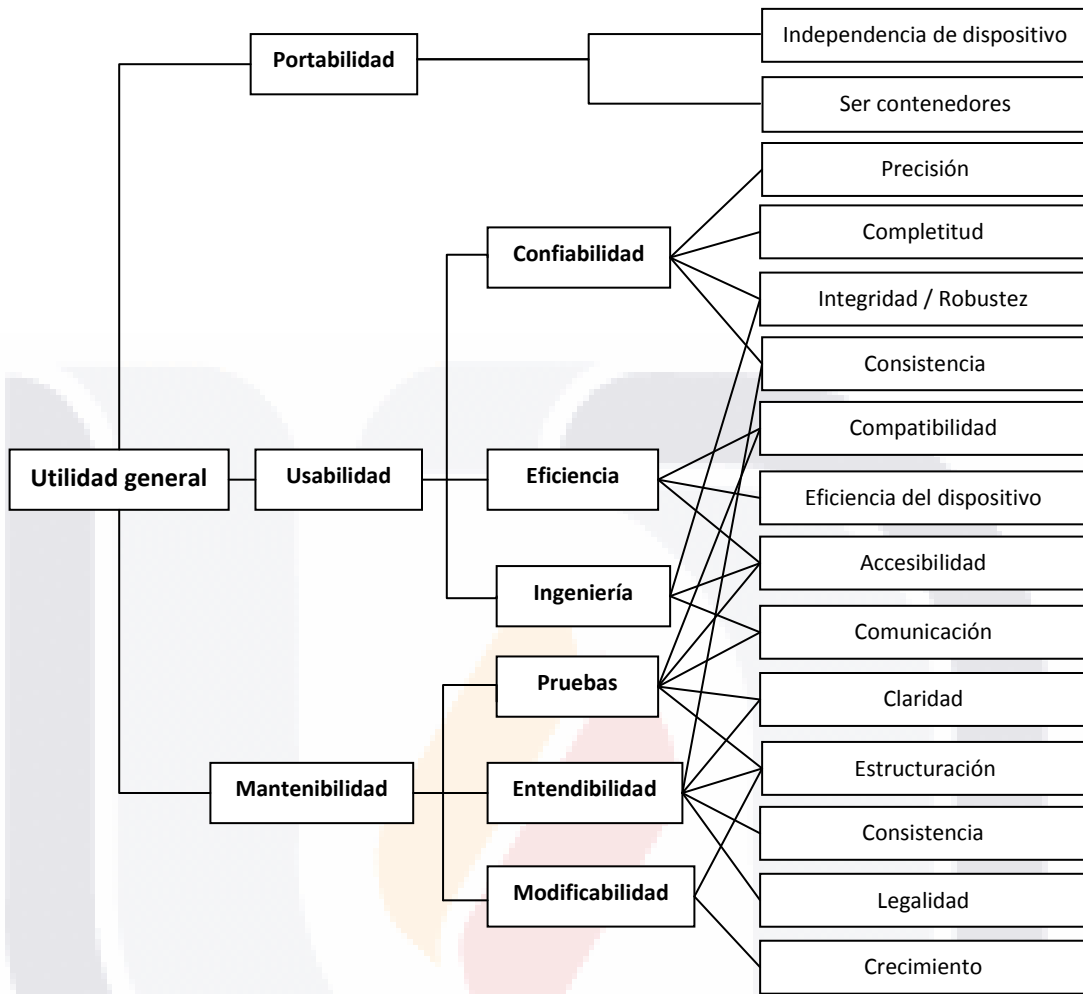


Figura 5. Modelo de Bohem (Adaptado de Dávila & Mejía, 2003)

2.3.2 Resumen de los modelos de calidad de software

En la sección anterior se describieron detalladamente algunos de los modelos de calidad de software más populares, éstos se centran en definir una serie de atributos de calidad que contribuyen a la calidad total (Kumar, Grover, & Kumar, 2009) y establecen una relación entre factores, criterios y métricas. Ninguno de los modelos descritos tiene una aceptación total; aunque todos tienen como objetivo proporcionar un marco de referencia que sirva como base para lograr la calidad del software, tienen coincidencias y diferencias en algunos puntos.

Pressman (2007) afirma que los factores de la calidad del software no cambian con el tiempo, resaltando que el modelo de McCall propuesto en la década de 1970 es tan valido ahora como lo fue entonces. Los modelos de McCall y Bohem sirvieron como base para la creación de modelos posteriores como el ISO/IEC 9126, FURPS y Dromey, entre otros; éstos modelos resultan bastante similares, ambos poseen una estructura jerárquica y contienen características que a su vez se descomponen en sub-características, sin embargo Ortega et al. (2003) resaltan que a diferencia del modelo de McCall, el modelo de Bohem si considera el rendimiento del hardware.

Los modelos de calidad descritos en éste trabajo sugieren diferentes maneras de unir las características de calidad, en la Tabla 3 se pueden apreciar las características que son similares en cada modelo.

Modelos	Modelo de McCall	ISO/IEC 9126	Bohem	FURPS
Característica				
Facilidad de Mantenimiento - Mantenibilidad	●	●	●	●
Flexibilidad – Facilidad de modificación	●		●	
Facilidad de prueba	●	●	●	●
Portabilidad – Movilidad	●	●	●	
Facilidad de reutilización - Reusabilidad	●	●		
Interoperabilidad-Compatibilidad	●	●	●	●
Corrección – Idoneidad – Precisión	●	●	●	
Confiabilidad - Fiabilidad	●	●	●	●
Facilidad de uso - Usabilidad	●	●	●	●
Integridad - Robustez	●	●	●	
Eficiencia - Desempeño	●	●	●	●
Funcionalidad	◐	●		●
Ingeniería			●	

- Característica contenidas en el modelo
- ◐ Característica contenida parcialmente en el modelo

Tabla 3. Comparativo de Modelos de Calidad

De la misma manera, como se aprecian similitudes entre los modelos, también existen divergencias, por ejemplo, el modelo ISO/IEC 9126 hace hincapié en el mantenimiento, a diferencia del modelo de McCall que también lo contempla pero con una definición muy limitada. Los factores de calidad son expresados con nombres diferentes aunque hagan referencia al mismo concepto.

El presente trabajo se basa en el modelo de McCall y el ISO/IEC 9126, puesto que el primero ha servido de base para la creación de los demás modelos y el segundo representa el modelo más completo realizado hasta la fecha de acuerdo con Khan et al. (2008) y está amparado por un amplio consenso internacional.

2.4 Métricas

Khan et al. (2008) exponen que la medición de la calidad de software no es un tema nuevo ya que ha sido investigada por más de dos décadas en la disciplina de ingeniería de software. También aseguran que las mediciones se han convertido en una parte esencial para la buena ingeniería de software. Adicionalmente, ellos explican que DeMarco señala la importancia de éstas con la siguiente afirmación: “No se puede controlar lo que no se puede medir” (pág. 103) de aquí que solamente se podrá controlar y mejorar un elemento, del cual se tenga un punto de referencia del cual partir, es decir, que se cuente con una medición como base.

Es recomendable que el software sea evaluado periódicamente de manera cuantitativa durante su desarrollo, con la finalidad de saber si al final será capaz de cumplir sus objetivos (Cavano & McCall, 1978).

Pressman (2007) explica que en el contexto de la ingeniería de software, una medida aporta una indicación cuantitativa de la extensión, la cantidad, la dimensión, la capacidad o el tamaño de algún atributo de un producto o proceso. Sin embargo, también expone que el esfuerzo por desarrollar medidas precisas de la calidad del software en ocasiones resulta frustrante por la naturaleza subjetiva de éste.

De acuerdo al IEEE Standard Glossary of Software Engineering Terms, una métrica es una “medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo determinado” (Khan, Mustafa, & Ahson, 2008, págs. 81-82).

Khan et al. (2008) explican que las métricas de software generalmente son aplicadas a los siguientes aspectos:

- Productos de ingeniería de software. Por ejemplo diseño, código fuente y casos de prueba.
- Procesos de ingeniería de software. Por ejemplo las actividades de análisis, diseño y codificación.
- Equipo de ingeniería de software. Por ejemplo la eficiencia o productividad.

Se resalta también que las métricas a pesar de ser aplicadas a los tres aspectos anteriores están más estrechamente relacionadas con el proceso y el producto que con el equipo. De la misma manera, cuando se establece una métrica para evaluar la calidad de un producto y se obtienen resultados satisfactorios, de alguna manera se podría inferir que el proceso por medio del cual se elaboró dicho producto cuenta con calidad similar a la de su resultado, no siendo de la misma manera cuando se evalúa en primera instancia el proceso.

También exponen que las métricas de software facilitan y permiten lo siguiente:

- Definir cuantitativamente el éxito o fracaso, de un producto, de un proceso o persona,
- identificar y cuantificar las mejoras, la falta de mejora o deficiencias de un producto, proceso o persona,
- tomar decisiones significativas y útiles,
- identificar desviaciones y
- cuantificar y estimar significativamente.

Según Khan et al. (2008), otra manera en que las métricas pueden ser categorizadas es: primitivas y calculadas. Las métricas primitivas son aquellas que pueden ser directamente observables, como la cantidad de líneas de código o el número de defectos de un software. Mientras que las métricas calculadas no pueden ser directamente observadas pero son obtenidas de alguna manera en función de otras métricas, por ejemplo la métrica de Halstead que utiliza McCall para el criterio de Concisión (Ver Tabla 1).

Pressman (2007) explica que un ingeniero de software recopila medidas y desarrolla métricas con lo que obtiene indicadores, los cuales proporcionan conocimiento acerca del proceso, proyecto o producto.

A continuación se presenta el marco de trabajo de las métricas de calidad de software (ver Figura 6), el cual presenta uno de los principales enfoques de una métrica:

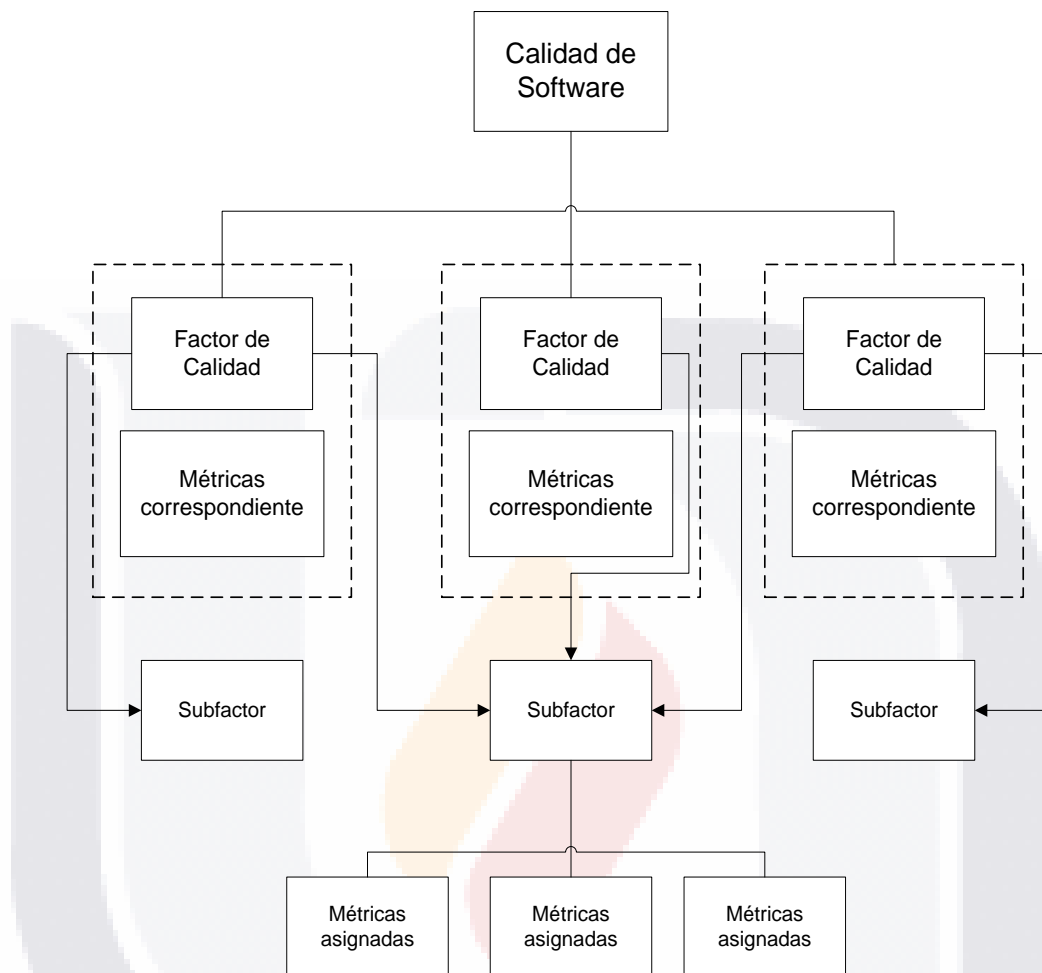


Figura 6. Marco de trabajo de las métricas de calidad de software (Adaptado de Khan, Mustafa, & Ahson, 2008)

Khan et al. (2008) explican también las características esenciales que debe tener una métrica de calidad, las cuales son:

- **Cumplimiento:** Es la capacidad de cubrir todos los aspectos de los factores de calidad y las características de diseño.
- **Ortogonalidad:** Es la capacidad de representar los diferentes aspectos del sistema en la medición

- **Formalidad:** Es la capacidad para obtener el mismo valor para los mismos sistemas para diferentes personas en diferentes momentos a través de una especificación precisa y objetiva.
- **Minimalidad:** Es la capacidad de ser usada con un número mínimo de métricas.
- **Implementabilidad/Usabilidad:** Es la capacidad de implementación independiente de la tecnología.
- **Exactitud:** Es una medida cuantitativa de la magnitud de error, preferentemente expresada en función del error relativo.
- **Validez:** se refiere al grado en que un estudio refleja con precisión o evalúa el concepto específico que el investigador está tratando de medir
- **Confiabilidad:** Es la probabilidad de falla del software en operación por un periodo de tiempo determinado en un entorno determinado.
- **Interpretabilidad:** Es la facilidad con la cual el usuario puede entender, usar propiamente y analizar los resultados de las métricas.

Pressman (2007) explica que no todas las métricas utilizadas actualmente cumplen con las características ideales, lo cual no significa que carezcan de valor, sino que se deben usar de manera cuidadosa y comprender que su objetivo es proporcionar conocimiento y no una comprobación científica sólida. Asimismo, expone que la medición asigna números o símbolos a atributos de entidades reales, y que su objetivo es ayudar al ingeniero de software a establecer una manera sistemática y objetiva de conocer a fondo su trabajo y, como resultado, mejorar la calidad del producto. “La medición es esencial si se desea alcanzar la calidad” (Pressman, 2007, pág. 469).

Según Piattini (2008), una vez que se ha identificado lo que se desea medir, se deben especificar aspectos como la unidad en la que se expresa, la escala a la que pertenece, él o los atributos para los que se define la métrica. La escala es el conjunto de valores con propiedades definidas, el tipo de escala indica la relación

entre los valores de ésta; entre las escalas más destacadas se encuentran las siguientes: nominal, ordinal, intervalo, ratio y absoluta.

Como resultado de la importancia de la medición han surgido varios marcos de trabajo entre los que destacan el GQM (Goal Question Metric) o PSM (Practical Software Measurement), estándares como el ISO 15939 y el IEEE Std 1061-1998; el objetivo principal de los anteriores es proporcionar una referencia para poder medir de forma efectiva y sistemática, en base a una serie de objetivos (Piattini & García, 2008).

A pesar de los modelos surgidos para facilitar la medición del software y la gran cantidad de métricas existentes, Somerville (2006) argumenta que medir los atributos de calidad de un software no resulta del todo sencillo. Por ejemplo: evaluar el atributo de mantenibilidad implica la evaluación de facilidad de análisis, facilidad de cambio, estabilidad y facilidad de prueba, así como este atributo existen otros como la usabilidad que contempla la facilidad de entendimiento, facilidad de aprendizaje, facilidad de operación y estética.

La usabilidad o facilidad de uso puede llegar a ser uno de los elementos del software más difíciles de medir dada su naturaleza subjetiva. Somerville (2006) propone que las métricas de usabilidad se pueden idear, y expone como ejemplo una métrica de aprendizaje, la cual consiste en que a un operador familiarizado con las tareas implementadas le debe ser posible utilizar el 80% de la funcionalidad de un software después de tres horas de formación. También explica que en la evaluación de interfaces de usuario están involucrados el criterio y la experiencia de los diseñadores.

De acuerdo con Pressman (2007), el usuario juzga la calidad de un sistema de software basándose principalmente en la calidad de la interfaz de usuario. En base a esto, se puede decir que aunque un software tenga la calidad suficiente en otros aspectos como la confiabilidad, portabilidad, entre otros, si no cuenta con una interfaz de usuario de calidad, el usuario emitirá un juicio negativo acerca de la totalidad del sistema. Asimismo, explica que un mal diseño de interfaz reduce la

capacidad del usuario para aprovechar las ventajas y funcionalidades de una aplicación. Además, argumenta que una interfaz deficiente puede llevar al fracaso una aplicación bien diseñada y con una implementación sólida.

2.5 Interfaces gráficas de usuario

Galitz (2007) explica que el diseño de la interfaz de usuario es una sub-área perteneciente al campo de estudio de Human-Computer Interaction (HCI). Asimismo, Machiraju (1996) afirma que la interfaz de usuario es uno de los factores de mayor impacto para la organización. El diseño de la interfaz se concentra en tres áreas (Pressman, 2007):

1. Diseño de interfaces entre componentes de software
2. Diseño de interfaces entre el software y otros productos y consumidores de información que no son humanos (entidades externas).
3. Diseño de la interfaz entre un ser humano (usuario) y la computadora.

Pressman (2007) afirma que “El diseño de la interfaz de usuario crea un medio de comunicación efectiva entre un ser humano y una computadora” (pág. 350). De acuerdo con esta afirmación, es de suma importancia que el software cuente con una interfaz de usuario que sea capaz de satisfacer las necesidades y requerimientos de este, para así, en realidad se pueda dar la comunicación efectiva.

Dentro de la tercera área, se puede distinguir entre las interfaces de hardware y software. En el presente trabajo se estudian únicamente las segundas.

De acuerdo con Garlitz (2007), HCI es el campo que se enfoca en el estudio, planificación y diseño de cómo las personas trabajan en conjunto con una computadora con el fin de satisfacer las necesidades de los usuarios de manera eficaz; también menciona que HCI considera aspectos tales como: que es lo que la gente quiere y espera, cuales son las limitaciones y habilidades físicas que posee, como es su percepción y procesamiento de información y que es lo que le parece atractivo y agradable. Adamaczyk (2005) explica que el área de HCI tiene

como principal objetivo hacer que las computadoras sean más amigables para el usuario.

“El diseño de la interfaz de usuario es el proceso de diseñar el modo en el que los usuarios del sistema acceden a la funcionalidad de este y la forma en la que se visualiza la información producida por el sistema” (Somerville, 2006, pág. 651).

De acuerdo con la definición anterior se pueden distinguir dos componentes principales en la interfaz de usuario: las entradas y las salidas. Galitz (2007) las define de la siguiente manera:

Entradas: es la forma en que las personas comunican sus necesidades o deseos al equipo, entre los mecanismos de entrada más comunes se encuentran el teclado, el ratón, la voz, entre otras.

Salidas: es la forma en que la computadora transmite los resultados de los cálculos o requerimientos del usuario, entre los mecanismos de salida más comunes están la pantalla y el audio.

Una interfaz de usuario tendrá una mezcla de mecanismos de entrada y salida que logren satisfacer eficazmente las necesidades de los usuarios. Galitz (2007) argumenta que una buena interfaz es aquella que no se nota y permite al usuario centrarse en la información y en la tarea que realiza, en vez de distraerse con los mecanismos que utiliza para presentar la información y realizar la tarea.

A lo largo de la historia de las computadoras se han desarrollado diferentes tipos de interacción entre éstas y los usuarios. Al principio la comunicación se llevaba a cabo por medio del teclado y lenguaje de comandos. Posteriormente, la empresa Xerox implementó el dispositivo mouse y se crea el ambiente gráfico.

Galitz (2007) afirma que los gráficos revolucionaron el diseño de las interfaces de usuario. Estas se caracterizan por ser representadas como un escritorio y contar con elementos como folders, contenedores de basura, bandejas, entre otros. A este tipo de interfaz también se le conoce como WIMP (Windows) ventanas,

(Icons) iconos, (Menus) menús y (pointing device) puntero. De la misma manera asegura que cuando la información presentada de manera gráfica es más fácil de entender y procesar por el usuario.

Para llevar a cabo un buen diseño de interfaz de usuario es necesario tomar en cuenta una serie de aspectos importantes. Según Pressman (2007), el conocimiento del usuario es fundamental, ya que se deben tomar en cuenta aspectos como el género, la edad, la religión, cultura, condiciones físicas, entre otras. De acuerdo a estos aspectos variarán las preferencias. Por ejemplo, según Sharp (2007) las personas mayores prefieren en pantalla texto y elementos gráficos grandes debido al deterioro visual que pudieran tener, también da otro ejemplo referente a la cultura: en Estados Unidos el formato de fecha es escrito en el orden de mes, día y año, a diferencia de otros países en las que se escribe como día, mes y año.

Otro factor relevante es la experiencia del usuario para el que está dirigido el sistema. Pressman (2007) explica que los usuarios expertos prefieren interfaces que puedan manejar, esperan un rápido desempeño y tareas más complejas, a diferencia de los usuarios novatos quienes prefieren interfaces que los guíen, tareas simples, menor número de opciones, mayor ayuda e información. En relación a esto, Galitz (2007) menciona una clasificación de usuarios en base a la experiencia: expertos, intermedios y novatos, también explica que lo que es fácil para un usuario “experto” no lo es para un usuario “novato” y viceversa, y asegura que “el desafío en el diseño es satisfacer las necesidades de los expertos, sin introducir complejidad para los menos experimentados” (pág. 90).

Conocer al usuario final puede representar una tarea bastante compleja, ya que en muchas ocasiones éste opone resistencia. En torno a los aspectos anteriores, Galitz (2007) explica que hasta últimamente los investigadores han invertido más esfuerzo en el área de interfaces de usuario, ahora ya se estima que el código de programación dedicado a esta excede el 50%. Sin embargo, aún en la actualidad quedan muchos ejemplos de malos diseños.

Entre los beneficios que trae un buen diseño de interfaz de usuario, Galitz (2007) resalta el incremento en la productividad de las personas que las usan, la disminución del tiempo invertido en realizar transacciones y la disminución de errores.

Pressman (2007) expone que Shneiderman explica que algunas personas se encuentran con problemas tan serios de choque de computadora que evitan el empleo de sistemas de cómputo. Asimismo, afirma que aún en un “mundo de ventanas” se pueden encontrar interfaces de usuario que resultan difíciles de aprender y usar, poco intuitivas y en ocasiones sumamente frustrantes.

Galitz (2007) expone la siguiente reflexión: “Si la gente no puede utilizar eficazmente un sistema, entonces ¿de quién es la culpa? al parecer, la falla radica en el diseño del sistema, ya que las personas no pueden ser rediseñadas” (pág. 88).

Dado lo anterior, surge la siguiente incógnita ¿cómo hacer GUIs fáciles de usar? Atendiendo a esta problemática, existen varias alternativas propuestas por diversos autores. Por ejemplo:

Las 3 “reglas de oro” para del diseño de las interfaz de Theo Mandel (Pressman, 2007, pág. 351)

- Dar el control de la interfaz al usuario
- Reducir la carga de memoria de los usuarios
- Hacer la interfaz de usuario consistente

Las 10 heurísticas de usabilidad propuestos por Jakob Nielsen (2005)

1. Visibilidad del estado del sistema. El sistema debe siempre mantener a los usuarios informados del estado del sistema, con una realimentación apropiada y en un tiempo razonable.
2. Utilizar el lenguaje de los usuarios. El sistema debe hablar el lenguaje de los usuarios, con las palabras, las frases y los conceptos familiares, en lugar de

- que los términos estén orientados al sistema. Utilizar convenciones del mundo real, haciendo que la información aparezca en un orden natural y lógico.
3. Control y libertad para el usuario. Los usuarios eligen a veces funciones del sistema por error y necesitan a menudo una salida de emergencia claramente marcada, esto es, salir del estado indeseado sin tener que pasar por un diálogo extendido. Es importante disponer de deshacer y rehacer.
 4. Consistencia y estándares. Los usuarios no deben tener que preguntarse si las diversas palabras, situaciones, o acciones significan la misma cosa. En general siga las normas y convenciones de la plataforma sobre la que está implementando el sistema.
 5. Prevención de errores. Es importante prevenir la aparición de errores que mejor que generar buenos mensajes de error.
 6. Minimizar la carga de la memoria del usuario. El usuario no debería tener que recordar la información de una parte del diálogo a la otra. Es mejor mantener objetos, acciones, y las opciones visibles que memorizar.
 7. Flexibilidad y eficiencia de uso. Las instrucciones para el uso del sistema deben ser visibles o fácilmente accesibles siempre que se necesiten. Los aceleradores no vistos por el usuario principiante, mejoran la interacción para el usuario experto de tal manera que el sistema puede servir para usuario inexpertos y experimentados. Es importante que el sistema permita personalizar acciones frecuentes.
 8. Los diálogos estéticos y diseño minimalista. No deben contener la información que sea inaplicable o se necesite raramente. Cada unidad adicional de la información en un diálogo compite con las unidades relevantes de la información y disminuye su visibilidad relativa.
 9. Ayudar a los usuarios a reconocer, diagnosticar y recuperarse de los errores. Que los mensajes de error se deben expresar en un lenguaje claro, se debe indicar exactamente el problema, y deben ser constructivos.
 10. Ayuda y documentación. Aunque es mejor si el sistema se puede usar sin documentación, puede ser necesario disponer de ayuda y documentación. Ésta

tiene que ser fácil de buscar, centrada en las tareas del usuario, tener información de las etapas a realizar y que no sea muy extensa.

Principios de diseño de interacción de Bruce Tognazzini (2001).

Anticipación: Las aplicaciones deberían intentar anticiparse a las necesidades y deseos del usuario. No esperes que el usuario busque o recuerde información o herramientas. Muestra al usuario toda la información y herramientas necesarias para cada etapa en su trabajo.

Autonomía: El ordenador, la interfaz y el entorno de la tarea pertenecen al usuario, pero esto no significa que abandonemos todas las reglas.

Dale al usuario algo de "cancha". Los usuarios aprenden rápido y ganan confianza cuando se sienten que tienen el control del sistema. Mantén informado al usuario del estado del sistema. No existe autonomía en ausencia de control; y el control no se puede tener sin información suficiente. Comunicar el estado es fundamental para que el usuario responda apropiadamente con la información disponible. Mantener la información de estado fácilmente visible y actualizada.

Daltonismo: Si utilizas el color para transmitir información debes utilizar otros elementos complementarios para la gente con daltonismo. Aproximadamente un 10% de los hombres adultos sufren daltonismo. Las pistas secundarias pueden consistir en distintos tonos de gris, gráficos complementarios o etiquetas de texto.

Consistencia: Los siguientes principios, vistos en su conjunto, dan al diseñador de interacción mucho margen para la evolución de un producto sin perjudicar los aspectos más importantes para el usuario.

Niveles de consistencia: mantener una consistencia estricta depende del caso.

Inconsistencia: es tan importante ser visualmente inconsistente con los objetos que se comportan de forma distinta, como ser consistente con los que se comportan de igual manera.

La consistencia más importante es aquella que espera el usuario. La única manera de comprobar las expectativas del usuario es hacer pruebas con ellos.

Valores por defecto: Los valores por defecto deberían poder ser descartados con facilidad y rapidez. Los campos de texto con valores por defecto deben aparecer seleccionados, para que el usuario sólo tenga que teclear y no seleccionar todo, borrar y escribir. Los valores por defecto deben tener sentido.

Eficacia del usuario: Busca la productividad del usuario, no del ordenador. La gente cuesta mucho más dinero que los ordenadores, y aunque parezca que aumentando la productividad de la máquina aumentamos la del humano, lo habitual suele ser lo contrario. Cuando juzgues la eficacia de un sistema, vete más allá de la simple eficacia de la máquina.

Mantén ocupado al usuario: El gasto más alto en un negocio es el trabajo humano. Cada vez que el usuario tiene que esperar la respuesta del sistema, es dinero perdido. Para maximizar la eficacia de un negocio u organización debes maximizar la eficacia de todos y no sólo de un grupo

Las grandes organizaciones tienden a estar segmentadas, con cada grupo mirando por sus intereses, a veces en detrimento de la organización. Los saltos cualitativos en eficacia se encuentran en la arquitectura del sistema, no en su superficie, en el diseño visual de la interfaz. Esta es la razón por la que es tan importante que todo el mundo involucrado en un proyecto de software aprecie la importancia de hacer de la productividad del usuario el objetivo principal y entender la diferencia entre diseñar un sistema eficaz o potenciar la productividad del usuario. Esto implica que es fundamental la colaboración, comunicación y complicidad entre ingenieros y diseñadores de interacción si se quiere conseguir este objetivo.

Escribe mensajes de ayuda concisos y que ayuden a resolver el problema: un buen texto ayuda mucho en comprensión y eficacia.

Interfaces explorables: Dale al usuario caminos bien señalizados; luego deja que se metan a través del monte. Imita la seguridad, suavidad y consistencia del medio natural. No encierres al usuario en un único camino, pero ofrécele la ruta de menos resistencia. Esto facilita a los nuevos usuarios o a aquellos que sólo quieren hacer acabar la tarea, hacerlo rápidamente y sin esfuerzo; pero también hay que dejar posibilidades abiertas a aquellos que quieran explorar. Da a los usuarios nociones estables para saber cómo llegar al inicio. Los elementos visuales estables no sólo ayudan a navegar más rápido, sino que también actúan como una referencia necesaria para sentirse seguro. Haz que las acciones sean reversibles, siempre deja una salida abierta, utiliza objetos grandes para las funciones importantes.

Objetos humanos: Los objetos humanos de la interfaz se pueden ver, escuchar, tocar o percibir de otra manera, los objetos humanos visibles de la interfaz son bastante familiares. Aquellos que utilizan otros sentidos lo son menos, los objetos humanos de la interfaz deben ser comprensibles, consistentes y estables.

Reducción de latencia: Cuando sea posible, utiliza el multihilo para dejar la latencia en un segundo plano.

Aprendizaje: Lo ideal sería que no hubiese periodo de aprendizaje: los usuarios se sentarían delante del sistema por primera vez y sabrían cómo utilizarlo. Esto, sin embargo, nunca pasa. Es un mito que la facilidad de uso se consiga a costa de la facilidad de aprendizaje.

Uso de metáforas: Escoge aquellas metáforas que permitan al usuario comprender los detalles del modelo conceptual. Las buenas metáforas son historias que crean imágenes mentales.

Protege el trabajo del usuario: Asegúrate de que el usuario nunca pierde su trabajo como resultado de un error suyo, los problemas de internet u otro tipo de problemas inevitables, como un apagón.

Legibilidad: Utiliza texto con alto contraste. Procura utilizar negro sobre blanco o amarillo pálido. Evita fondos grises cuando haya texto. Utiliza tamaños de letra que se lean bien en los monitores más comunes. Da mayor importancia a los datos e información que quieres presentar, más que a instrucciones y etiquetas.

Guardar el estado: Debido a que la web utiliza un protocolo sin estado, nosotros debemos guardarlo en su lugar.

Navegación Visible: Evita la navegación invisible. La mayoría de los usuarios no pueden mantener mapas mentales complejos. Si tienen que hacerlo, se cansarán o se perderán.

Somerville (2006) expone que aunque muchos especialistas a menudo trabajan en el diseño gráfico y en el diseño de sistemas ergonómicos, normalmente sólo las organizaciones grandes emplean diseñadores especialistas de interfaces para sus aplicaciones de software. Asimismo, expresa que los ingenieros de software a menudo deben tomar la responsabilidad de diseñar la interfaz de usuario, así como del diseño del software a utilizar para la implementación de esta. También expone que aún cuando los diseñadores y programadores de software son usuarios competentes en la tecnología utilizada en la implementación de interfaces, el problema radica en que a menudo resultan productos poco atractivos, complejos e inapropiados para sus usuarios objetivo. “Es posible afirmar que la interfaz de usuario es el elemento más importante de un sistema o producto de cómputo” (Pressman, 2007, pág. 378). De acuerdo con esta afirmación, resulta sumamente crítico que ésta sea de calidad, dado que de lo contrario el sistema será juzgado como un producto carente de ésta.

2.5.1 Calidad de las interfaces gráficas de usuario

Al igual que la definición de calidad, la definición de calidad de interfaces de usuario resulta compleja ya que depende de la percepción de quien la juzga. Según Somerville “de forma ideal, una evaluación de una interfaz de usuario se debe llevar a cabo en torno a la usabilidad del sistema” (pág. 350).

Preece et al. (1994, pág. 602) define la evaluación del diseño como la “reunión de datos sobre la usabilidad de un diseño o un producto por parte de un grupo de usuarios específicos para una actividad en concreto dentro de un entorno determinado de trabajo” y como “una parte integral del proceso de diseño centrado en el usuario”.

La usabilidad es una de las seis características de producto que propone evaluar el modelo de calidad ISO/IEC 9126 (ver Tabla 2). De la misma manera, el modelo de calidad de McCall (1978) contempla la usabilidad dentro de los once factores que lo componen, el cual pertenece al área de operación del producto.

Galitz (2007) explica que Nielsen, considera que la usabilidad se puede dividir en cinco atributos principales:

- Aprendizaje: ¿Qué tan fácil es para los usuarios acoplarse a las tareas básicas del sistema la primera vez que lo usan?
- Eficiencia: Una vez que los usuarios conocen el diseño del sistema, ¿Qué tan rápido pueden desempeñar las tareas?
- Memorabilidad: Cuando los usuarios vuelven a utilizar el sistema después de un largo periodo ¿Qué tan fácil resulta volver a utilizarlo de manera competente?
- Errores: ¿Cuántos errores cometieron los usuarios?, ¿Qué tan grave es el error? ¿Qué tan fácil resulta recuperarse de los errores?
- Satisfacción: ¿Qué tan agradable resulta usar el sistema?

Asimismo, afirma que la usabilidad es una de las cualidades más importantes de la interfaz de usuario, y puede ser la causa de que sistemas Web no se utilicen o que aplicaciones de negocio den como resultado baja productividad. También explica que la usabilidad debe estar ampliamente relacionada con la utilidad, la cual se entiende como lo que el usuario desea hacer, ya que pueden existir aplicaciones con un alto grado de usabilidad que no cumplen con las tareas que el

usuario desea realizar. Por lo tanto, la aplicación no representará valor alguno para el usuario.

De acuerdo con Bosh (2003), el principal objetivo de medir la usabilidad de un sistema es mejorarlo y, en consecuencia, elevar su calidad; también asegura que pese a investigaciones realizadas en torno a la importancia de evaluar la calidad en las interfaces de usuario y a los elevados costos que representa entregar al cliente una interfaz deficiente, éstas continúan sin evaluarse apegadas a aspectos como la usabilidad o la aceptación, lo que aumenta el riesgo de fracaso.

Pressman (2007) explica que: “pueden existir varias formas de evaluar una interfaz, desde la simple retroalimentación verbal de un usuario hasta el uso de cuestionarios aplicados a una población de usuarios finales para la recolección de datos y el empleo de métodos estadísticos” (pág. 377). Als et al. (2005) explican que las pruebas de usabilidad se reconocen ampliamente para identificar los principales problemas de interacción en las interfaces de usuario.

Perlman (1996) afirma que una mala interfaz de usuario puede contribuir al error humano, incluso en daños personales y financieros, y explica que quizá la evaluación de la calidad de las interfaces sea algo en lo que los fabricantes de software no estén dispuestos a invertir, pese a su gran importancia, dado que pueden llegar a representar elevados costos.

Contrario a lo anterior, Galitz (2007) expone el caso de IBM, para la cual la usabilidad representa un aspecto sumamente rentable, y lo expresa por medio de la siguiente cita: “Por cada dólar invertido en la usabilidad del sistema se devuelven de \$10 a \$100 dólares” (pág. 6).

Como ya se ha mencionado anteriormente, existen distintas alternativas para el diseño de interfaces de usuario, las cuales buscan guiar al diseñador para lograr productos con calidad. Estas alternativas intentan establecer criterios generales de diseño, que aplicados al sistema ayuden al usuario a establecer con éste una comunicación que resulte fácil y efectiva.

2.6 Teoría psicológica Gestalt

Para lograr una comunicación efectiva entre el usuario y el sistema, éste último debe contar con características especiales que permitan al usuario comprenderlo fácilmente, en torno a esto, Galitz (2007) asegura que cuando la información se presenta de manera gráfica hace más efectiva la capacidad de procesamiento y se reduce la necesidad de recodificación de la información sensorial y mental, disminuyendo la carga en memoria, además de proporcionar un atractivo estético. Asimismo, menciona que actualmente entre los principales problemas de las interfaces de usuario están la gran falta de guías de diseño y las limitaciones de comprensión humana. Como una alternativa a estos problemas se puede considerar la teoría psicológica Gestalt, la cual, de acuerdo con Gordon (2004), es catalogada como la teoría formal de la percepción visual. Esta intenta explicar por medio de leyes o principios cómo organizamos elementos individuales dentro de grupos, por lo tanto, se puede utilizar para explicar la forma en que percibimos y reconocemos patrones (Dempsey & Keith V., 2006). Dicha teoría propone que la percepción está cargada de recuerdos. Como ejemplo Guillaume (1975) explica que el tiempo necesario para leer una palabra familiar es muy inferior al que exigiría la percepción distinta del mismo número de letras agrupadas de cualquier manera.

Ginger y Ginger (1993) explican que Gestalt surge en el año de 1912 en Alemania teniendo como principales autores a Max Wertheimer, Kurt Koffka y Wolfgang Köler. Inicialmente, dicha teoría fue estudiada solamente en el campo de la psicología, pero sus conceptos se han aplicado en diversas áreas de la ciencia (Dempsey & Keith V., 2006). Por ejemplo, en Ingeniería de software en las siguientes áreas: diseño de diagramas (Lemon, Allen, Carver, & Bradshaw, 2007), patrones de lenguaje (Flieder, 2007), estética de interfaces de usuario (Youn, Stolterman, Jung, & Donaldson, 2007), entre otras.

De acuerdo con Salama (2001) la palabra Gestalt, tiene varios significados, por ejemplo: configuración, estructura, modelo, figura, forma y totalidad, entre otros. El

principio general de la teoría es el siguiente: “Una forma es otra cosa o algo más que la suma de sus partes” (Guillaume, 1975, pág. 17). Uno de los puntos principales que los autores advirtieron fue que en todo campo perceptivo, es decir, el campo sensorial abarcable, se diferencian un fondo y una forma (Sinay & Blasberg, 1997). El sujeto tiende a aislar las “formas adecuadas” o “formas fuertes o estables” que rigen las relaciones entre el organismo y el medio (Ginger & Ginger, 1993, pág. 41). La percepción de las diferentes clases de elementos y de las diferentes clases de relaciones corresponde a diferentes modos de organización de un todo, que dependen a la vez de condiciones objetivas y subjetivas (Guillaume, 1975, pág. 21).

Dempsey et al. (2001) explican que la teoría Gestalt está expresada en leyes, muchas de ellas muy similares, por lo que varios psicólogos resumen en once las correspondientes a la teoría de la forma. Éstas son: simetría, continuidad, cerramiento, figura fondo, punto focal, correspondencia isomorfa, pregnancia, proximidad, similaridad, simplicidad y armonía. A continuación se describe brevemente cada una.

Similaridad

Los elementos visuales que son similares en forma, tamaño, color, proximidad y dirección son percibidos como parte de un grupo, incluso si los elementos están separados (Graham, 2008) (Ver Figura 7).

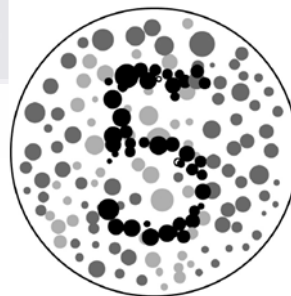


Figura 7. Similaridad (Adaptado de Graham, 2008)

Continuidad

El ojo humano está diseñado para crear relaciones entre las formas (Ver Figura 8). La continuidad se produce cuando el ojo sigue a lo largo de una línea recta, curva, o de una secuencia de formas, incluso cuando cruza sobre las formas negativas y positivas (Graham, 2008).



Figura 8. Continuidad

Punto focal

El principio del punto focal se refiere al elemento que captura la atención entre un grupo de elementos (Ver Figura 9). La gente percibe los elementos como puntos focales si los atributos de esos elementos son significativamente diferentes a los demás (Dempsey, Laurence, & Juhani, 2001).



Figura 9. Punto focal (Adaptado de Dempsey, Laurence, & Juhani, 2001)

Figura fondo

El principio ayuda a identificar los objetos (figura) a diferencia de su superficie (fondo) (Ver Figura 10). Esta ley de percepción depende del contraste. Las imágenes y el texto deben ser visibles para poder ser entendidas (Graham, 2008). Por ejemplo:

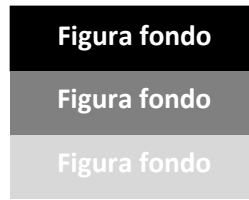


Figura 10. *Figura fondo* (Adaptado de Graham, 2008)

Unidad/Armonía

El principio de Unidad permite que un solo elemento pueda pertenecer a una sola fuente o grupo (Dempsey & Keith V., 2006) (Ver Figura 11). La unidad implica una congruencia o acuerdo entre los elementos de un grupo (Dempsey, Laurence, & Juhani, 2001).



Figura 11. *Unidad/Armonía* (Adaptado de Dempsey, Laurence, & Juhani, 2001)

Simetría

El principio de simetría también es llamado principio de balance. La gente tiende a sentirse más cómoda con un sentido de equilibrio al ver, oír o tocar (Ver Figura 12). La idea de éste principio es el de lograr una ponderación igual de los atributos cuando se muestra un elemento (Dempsey & Keith V., 2006).

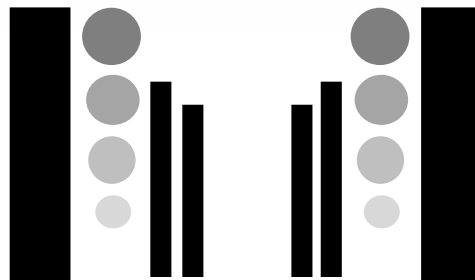


Figura 12. *Simetría*

Proximidad

Los objetos que están ubicados cerca uno del otro parecen parte de un grupo, mientras que los que están ubicados de manera distante, parecen ser de un grupo distinto (Graham, 2008) (Ver Figura 13).



Figura 13. Proximidad (Adaptado de Graham, 2008)

Cierre

Los seres humanos tienen una tendencia natural a cerrar visualmente los huecos en una forma, especialmente en formas familiares (Ver Figura 14). Cuando falta información, nos centramos en lo que está presente, se tiende a ignorar las partes que faltan y rellenar los huecos con una línea familiar. Buscamos cerrar formas para que sean estables (Graham, 2008).



Figura 14. Cierre

Correspondencia isomorfa

El principio propone que no todas las imágenes tienen el mismo significado para todos, sino que las interpretamos según nuestra experiencia (Dempsey, Laurence, & Juhani, 2001)(Ver Figura 15).



Figura 15. *Correspondencia isomorfa* (Adaptado de Dempsey, Laurence, & Juhani, 2001)

Concisión/Pregnancia

También es llamado el principio de buena forma. Una buena forma es un diseño simple o un diseño simétrico (Ver Figura 16).

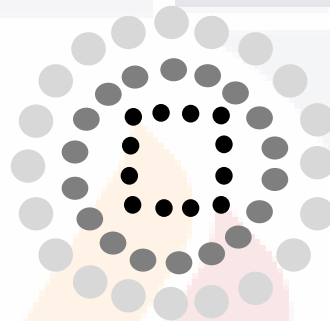


Figura 16. *Concisión/Pregnancia*

Simplicidad

La simplicidad funciona bien si el mensaje gráfico es despejado, pero si los gráficos son complejos o ambiguos el proceso de percepción del mensaje resulta confuso (Ver Figura 17).



Figura 17. *Simplicidad*

Dichas leyes o principios pueden ser interpretados como bases o reglas para el diseño de interfaces de usuario, con ellas se buscará construir productos cuyos elementos sigan una estructura que la mente reconozca y asocie de manera

natural, para con ello buscar que la interfaz resulte con un alto grado de usabilidad y en consecuencia lograr la calidad.

Galitz (2007) reporta que ya se han utilizado los principios de la teoría Gestalt en el diseño de interfaces de usuario en un estudio realizado por Tullis en 1981, que a su vez se basaba en otro estudio realizado por Zahn en 1971, en los cuales se tuvo como resultado la reducción de tiempos de localización visual de elementos cuando se utilizaba el principio de proximidad principalmente.

Dada la importancia y la aplicación que ha tenido la teoría Gestalt desde 1924 en diversas áreas de la ciencia (Dempsey, Laurence, & Juhani, 2001) y basándose en la afirmación del psicoanalista Fritz Perls: “la Gestalt es una terapia demasiado buena como para dedicársela sólo a los enfermos” (Sinay & Blasberg, 1997, pág. 27) se cree interesante enfatizar su aplicación en el diseño de interfaces gráficas usuario, campo en el cual como ya se expuso hacen falta guías de diseño basadas en aspectos cognitivos (Aberg & Chang, 2005).

2.7 Impacto en la calidad de GUIs diseñadas con Gestalt

La presente investigación evalúa el impacto que pudiera tener la teoría psicológica Gestalt en el desarrollo de GUIs desde el punto de vista de la calidad del software. Los organismos certificadores de la calidad generalmente se basan en el empleo de estándares y modelos, razón por la cual en esta investigación se hace uso del estándar de calidad ISO/IEC 9126 así como el modelo de McCall. El estudio mide el impacto tomando los siguientes factores: facilidad de uso, facilidad de aprendizaje, los cuales forman parte del aspecto de Usabilidad contemplado en el estándar de calidad ISO 9126 (1991), así como en el área de operación del producto del modelo de McCall, y grado de aceptación por parte del usuario. Como ya se ha mencionado ampliamente, las personas tienden a juzgar la calidad del software basándose principalmente en la interfaz de usuario y Somerville (2006) afirma que la calidad de ésta se evalúa en torno a su nivel de Usabilidad.

TESIS TESIS TESIS TESIS TESIS

Curtis (1984) explica que desde 1950 los psicólogos han estudiado los aspectos de comportamiento en la ingeniería de software, sin embargo, los resultados de éstas investigaciones nunca han podido organizarse dentro de una sub-área de la misma ni de la psicología, lo que puede indicar la falta de investigación en este campo, el cual trata de manera más amplia la interacción de éstas dos disciplinas: la ingeniería de software y la psicología.

Se propone el empleo de la teoría Gestalt dado que se trata de la teoría formal de la percepción visual, la cual busca explicar como la mente percibe los elementos que llegan a ella. El desarrollo de GUIs bajo los principios de la teoría psicológica Gestalt busca facilitar a los usuarios la percepción y la comprensión de los elementos presentados en pantalla, y con ello contribuir al logro de la facilidad de uso, la facilidad de aprendizaje y el nivel de aceptación del usuario.

Las GUIs construidas bajo esta teoría tendrán la ventaja de proporcionar al usuario una serie de elementos estructurados de tal manera que al llegar a sus sentidos, su mente pueda asociarlos de manera natural, y por consiguiente, procesarlos y comprenderlos más rápida y fácilmente; todo esto busca obtener productos que satisfagan al usuario y contribuir en el logro de productos con calidad.

En conclusión, la presente investigación evaluó aspectos de sistemas de información considerados como relevantes por parte de los usuarios finales de una forma estructurada y tomando como bases estándares con el fin de minimizar la subjetividad que la calidad del software tiene. Y esta es una de las principales contribuciones identificadas. Los siguientes capítulos tratan más ampliamente todo el proceso de investigación así como los resultados obtenidos.

Capítulo 3

Metodología de investigación

La evaluación de un sistema de información por parte del usuario final antes de ser entregado resulta vital para que el proyecto alcance el éxito deseado. La GUI de cualquier software representa uno de los aspectos más importantes en el logro de dicho éxito, por lo que su evaluación resulta fundamental. Con la finalidad de dar respuesta a la pregunta de investigación y confirmar o rechazar las hipótesis planteadas se realizó un experimento, el cual se detalla en las siguientes secciones.

3.1 Descripción del experimento

Para llevar a cabo el experimento, fue necesario rediseñar un sistema web ya existente, el cual pertenecía a una institución educativa de nivel medio superior, de esta manera se tuvieron dos sistemas uno diseñado sin Gestalt (sistema ya existente) y otro diseñado con Gestalt (sistema rediseñado). En el experimento participaron dos grupos de sujetos, uno se tomó como grupo de control y otro como grupo experimental.

Previo a la aplicación del experimento se realizaron las actividades descritas en la Tabla 4.

Día	Actividad	Objetivo
1	Identificar el semestre, grupo y carrera de los alumnos que colaborarán	Verificar que los sujetos tengan una formación similar.
2	Tener una plática con los alumnos acerca de su participación voluntaria en el experimento	Tratar de asegurar la participación de los sujetos, con el fin de procurar el éxito del experimento.
2	Confirmar el listado de alumnos que participarán	Reunir los participantes suficientes para el experimento.
2	Aplicación de un cuestionario de carácter demográfico a los participantes	Recabar información demográfica
3	Establecer la fecha y hora para la aplicación de cuestionarios	Fijar formalmente la fecha y la hora del experimento
4	Instalación del software	Instalar correctamente el software que se utilizará en el experimento
4	Pruebas del software	Verificar que el software funcione correctamente

Tabla 4. Actividades previas al experimento

Una vez llevadas a cabo las actividades previas se realizó el experimento con el grupo de control, el cual evaluó el sistema sin Gestalt, y con el grupo experimental, el cual avaluó el sistema con Gestalt. En la Tabla 5 se describen las actividades realizadas.

Día	Actividad	Duración
5-6	Reunión de los sujetos del grupo1 de muestras independientes	5 min
5-6	Proporcionar Instrucciones a los participantes	5 min
5-6	Exploración y uso del sistema 1 por parte de los participantes	25 min
5-6	Entrega del cuestionario y explicación de instrucciones	5 min
5-6	Contestar cuestionario	15 min
5-6	Recolección de cuestionario	3 min
5-6	Reunión de los sujetos del grupo 2 de muestras independientes	5 min
5-6	Proporcionar Instrucciones a los participantes	5 min
5-6	Exploración y uso del sistema 2 por parte de los participantes	25 min
5-6	Entrega del cuestionario y explicación de instrucciones	5 min
5-6	Contestar cuestionario	15 min
5-6	Recolección de cuestionario	3 min

Tabla 5. Actividades realizadas en el experimento con el grupo experimental y de control

3.1.1 Población objetivo

De acuerdo a cifras reportadas por el INEGI (2005) en México existen 7,921,850 jóvenes de entre 16 y 19 años, de los cuales el 49.2% son hombres y el 50.8% son mujeres. De esta cifra, el 47.8% (3,782,493) asiste a la escuela, siendo el 49.4% de hombres y 50.6 mujeres.

El sistema que se diseñó estuvo dirigido hacia los alumnos que cursan el nivel medio superior, concretamente, bachillerato cuyas edades oscilan entre los 16 y 19 años. Alumnos que requieren utilizar sistemas de información para realizar consultas o tramites referentes a la institución a la cual pertenecen. Para efectos del presente estudio, se contemplan solo los alumnos que cursan el bachillerato en el estado de Zacatecas.

Para el estado de Zacatecas se reporta que existen 113,874 jóvenes de entre 16 y 19 años de los cuales 47.9% son hombres y 52.1% son mujeres; de estos, solo el 39.3% cursa el nivel medio superior distribuidos de la siguiente forma: el 46.2% son hombres y 53.8% son mujeres.

3.1.2 Selección de la muestra

La institución en la cual se llevó a cabo el experimento se encuentra ubicada en el estado de Zacatecas, dicha institución cuenta con una población total de 788 alumnos, de los cuales 49.1% (387) son hombres y 50.9 % (401) son mujeres. Porcentajes muy similares a los reportados en el censo (INEGI 2005), por lo cual los resultados obtenidos pueden ser extrapolados a la población estudiada. Para la realización del experimento, se tomó una muestra por conveniencia. Los participantes pertenecen a la carrera de Técnico en informática; dicha carrera está dividida en 3 grupos, de los cuales se eligieron 2, así como se asignó la tecnología de desarrollo de sistema de manera aleatoria. Los seleccionados son el grupo A (grupo de control) y el grupo C (grupo experimental), los cuales tenían 27 y 24 alumnos respectivamente. A continuación se describe la información demográfica de ambos grupos.

Edad

Respecto a la edad, en la Figura 18 se puede apreciar que la mayoría de los participantes en ambos grupos tenían una edad muy similar. El 83.3% del grupo con Gestalt contaba con 16 años, mientras que el 88.9% del grupo sin Gestalt también. De manera que respecto a la edad se puede afirmar que los grupos son muy similares.

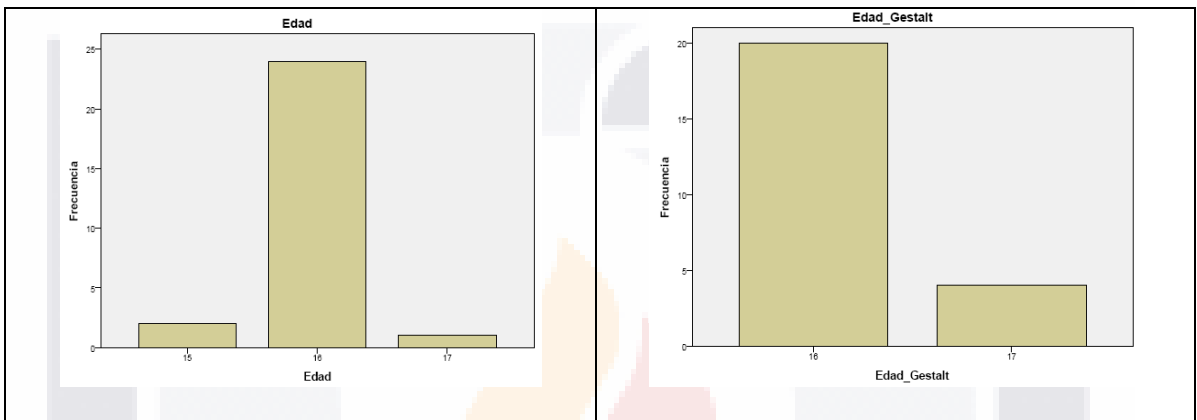


Figura 18. *Histogramas. Edad de los usuarios*

Género

En relación al género de los participantes, en el grupo experimental se tuvo una participación del 58% de hombres y 42% de mujeres; en el grupo de control participaron en una proporción de 48% hombres y 52% mujeres. En la Figura 19 se puede observar gráficamente la similitud entre ambos grupos.

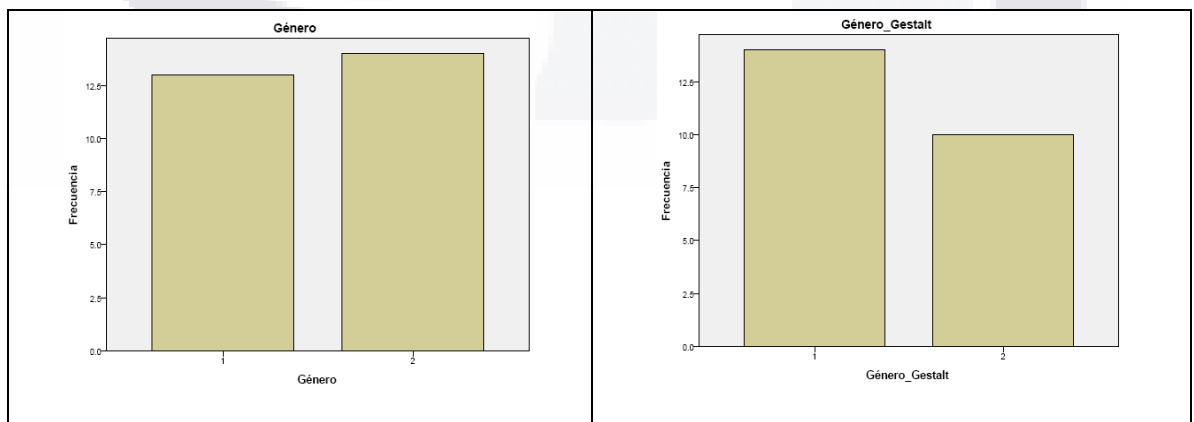


Figura 19. *Histogramas. Género de los usuarios*

Años de experiencia utilizando computadora

Respecto a la experiencia en el uso de computadora con la que contaban los participantes, en la Figura 20 se observa gráficamente que en ambos grupos, la mayoría de los sujetos manifestó tener entre 4 y 6 años aproximadamente usando computadora; 51.9 % en el grupo de control y 54.2% en el grupo experimental, de manera que en este aspecto los grupos se comportan de manera similar.

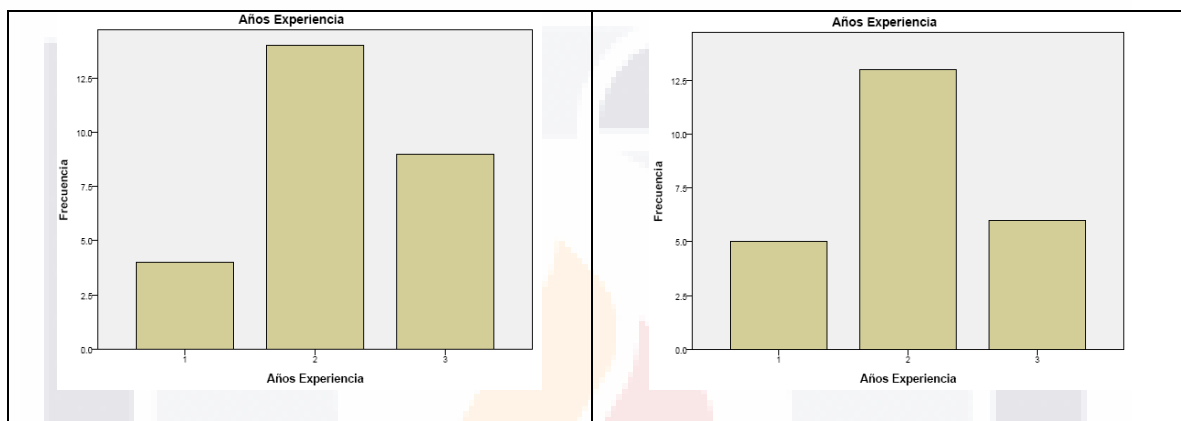


Figura 20. *Histogramas. Años de experiencia de los usuarios*

Uso de computadora

Respecto al uso de computadora expresado en horas a la semana, la Figura 21 muestra gráficamente la similitud entre los grupos. En ambos, el de control y el experimental, la mayoría manifestó usar la computadora entre 6 y 10 horas a la semana; 37% y 33.3 % respectivamente.

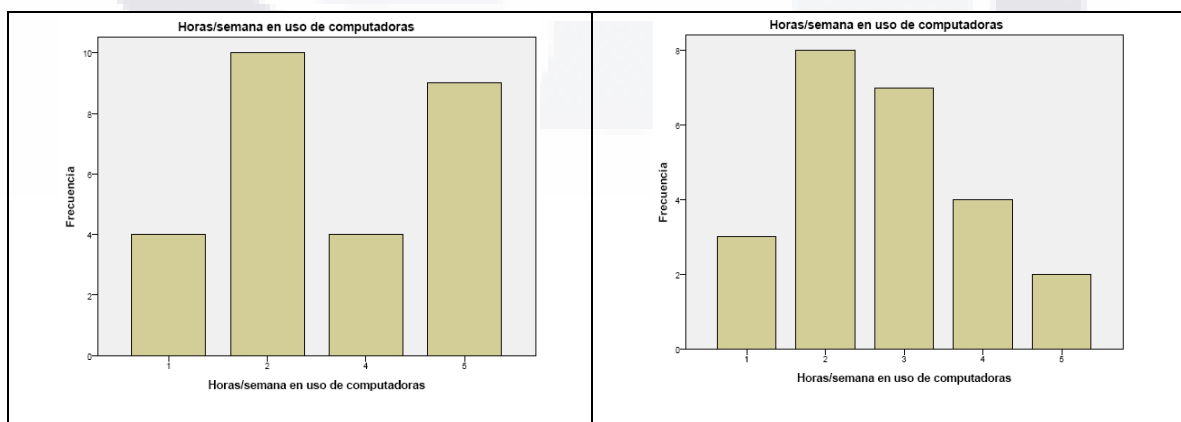


Figura 21. *Histogramas. Horas a la semana de uso de computadora*

Uso de Internet

En torno al uso de Internet, el grupo experimental la mayoría de los resultados se centraron en la opción 1 y 2 (62.5%), similar al grupo de control (59.3%). Ambas opciones 1 y 2 equivalen a 5 o menos horas y 6 a 10 horas respectivamente. De manera que se puede decir que el comportamiento en ambos grupos es muy similar. En la figura 22 se muestra gráficamente dicha similitud.

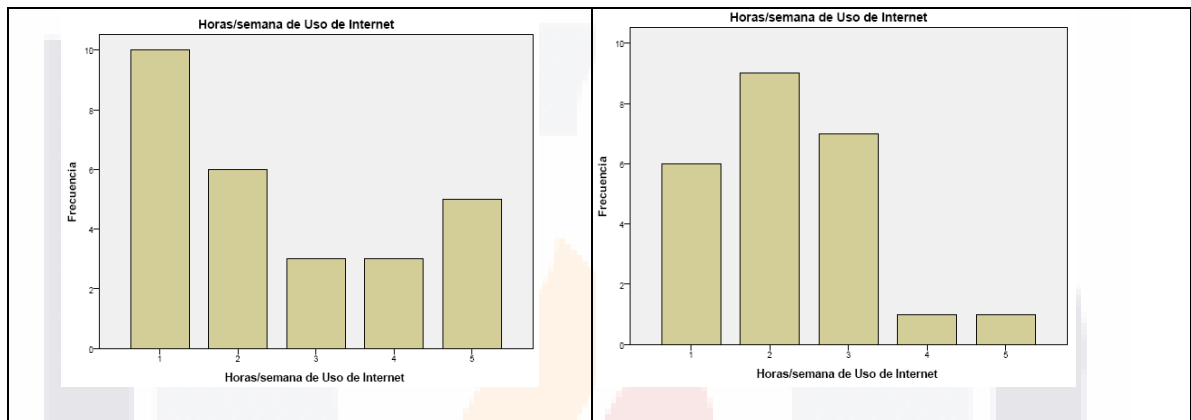


Figura 22. Histogramas. Horas a la semana de uso de Internet

Búsqueda de información

Respecto al uso principal de Internet, el 100% del grupo experimental manifestó usarlo para la búsqueda de información, mientras que solo el 63% del grupo de control expreso utilizarlo para esta actividad. En la Figura 23 se observan gráficamente las estadísticas.

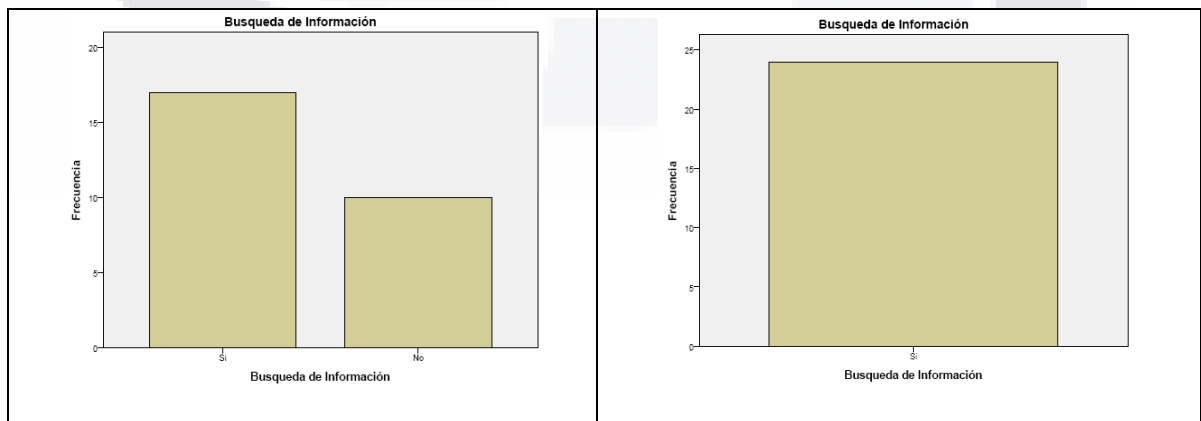


Figura 23. Histogramas. Uso de internet para búsqueda de información

Trabajo

En la Figura 24 se muestra que en el grupo de experimental el 75% de los sujetos expresa que uno de los usos principales de internet es la realización de trabajo, mientras que en el grupo de control solo el 48.1% manifestó utilizar el Internet para la realización de esta actividad.

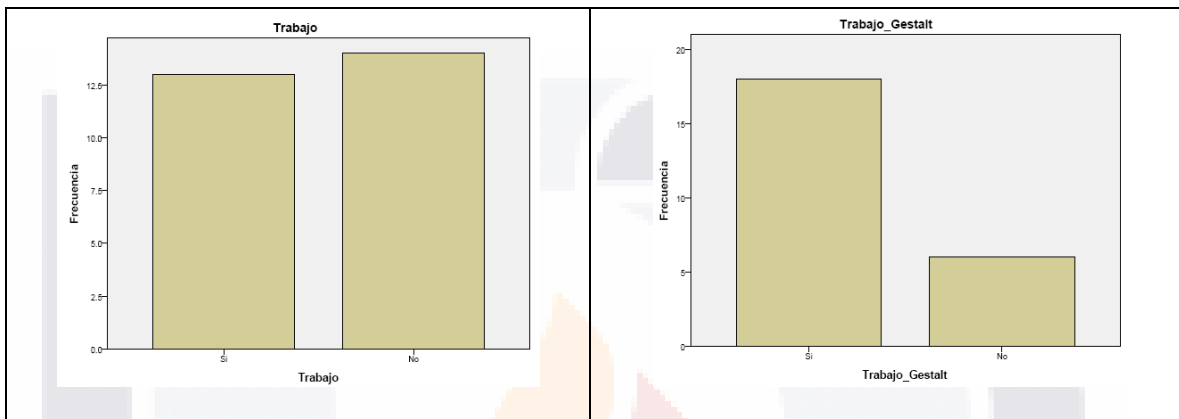


Figura 24. Histogramas. Uso de internet para realizar trabajo

Juegos

Respecto al uso principal de internet, el 29.2% del grupo experimental manifestó usarlo para juegos, de manera similar, el 22.2% del grupo de control expreso utilizarlo para esta actividad. En la Figura 25 se observa gráficamente el comportamiento de los datos

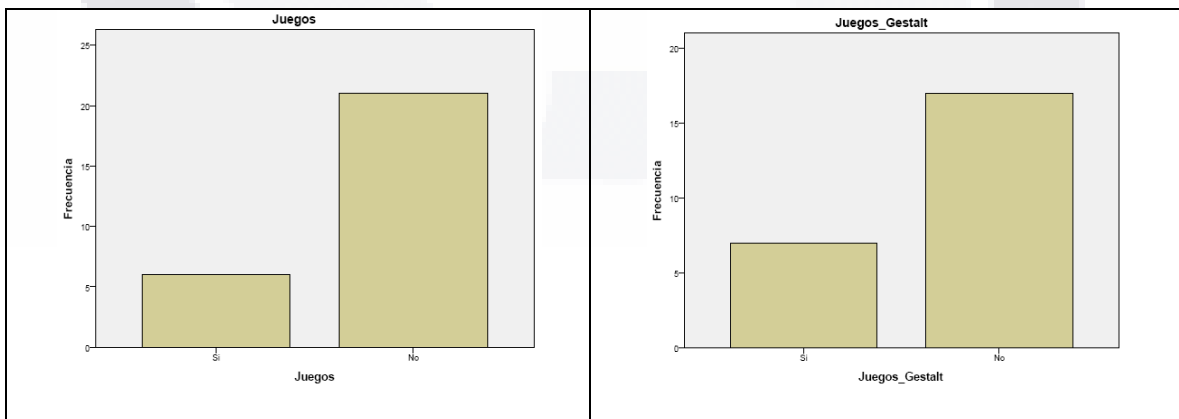


Figura 25. Histogramas. Uso de internet para juegos

Bajar música

Respecto al uso principal de internet, el 79.2% del grupo experimental manifestó usarlo para bajar música, asimismo, el 48.1% del grupo de control expreso utilizarlo para esta actividad. En la Figura 26 se observan gráficamente las estadísticas.

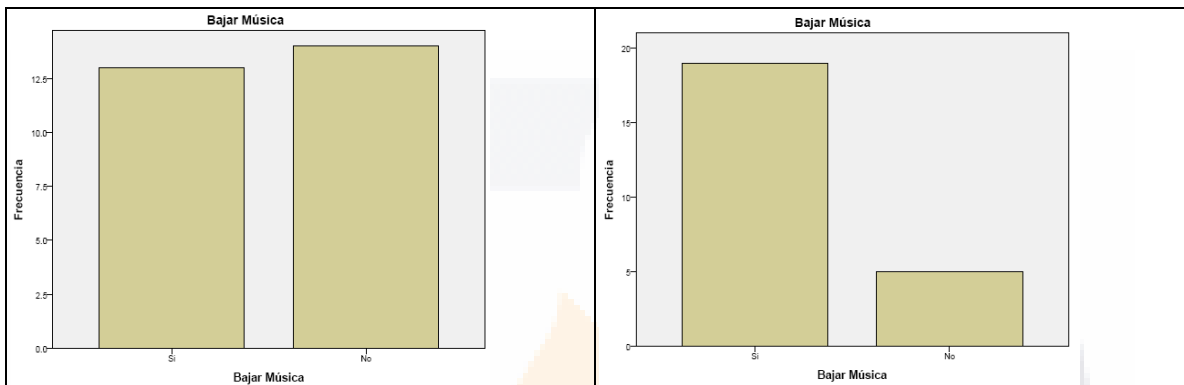


Figura 26. Histogramas. Uso se internet para descargar música

Ver videos

En la Figura 27 se muestra que en el grupo experimental el 79.2% de los sujetos expresa que uno de los usos principales de internet es ver videos, mientras que en el grupo de control solo el 44.4% manifestó utilizar el Internet para la realización de esta actividad.

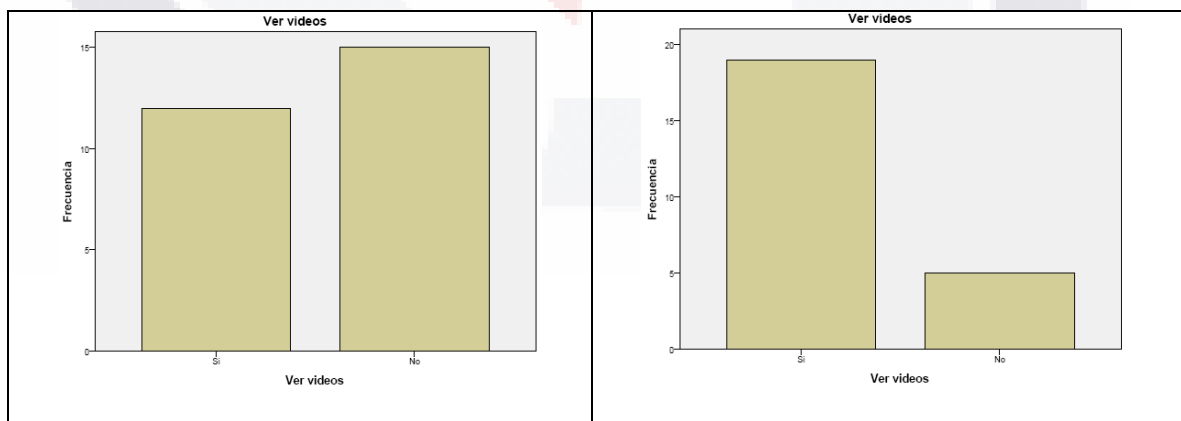


Figura 27. Histogramas. Uso de internet para ver videos

Otros usos

Respecto a otros usos de internet, el 66.7% del grupo experimental manifestó no usar Internet para otras actividades, de manera similar, el 59.3% del grupo de control también expreso no utilizarlo para otras actividades. En la Figura 28 se observan gráficamente las similitudes entre ambos grupos.

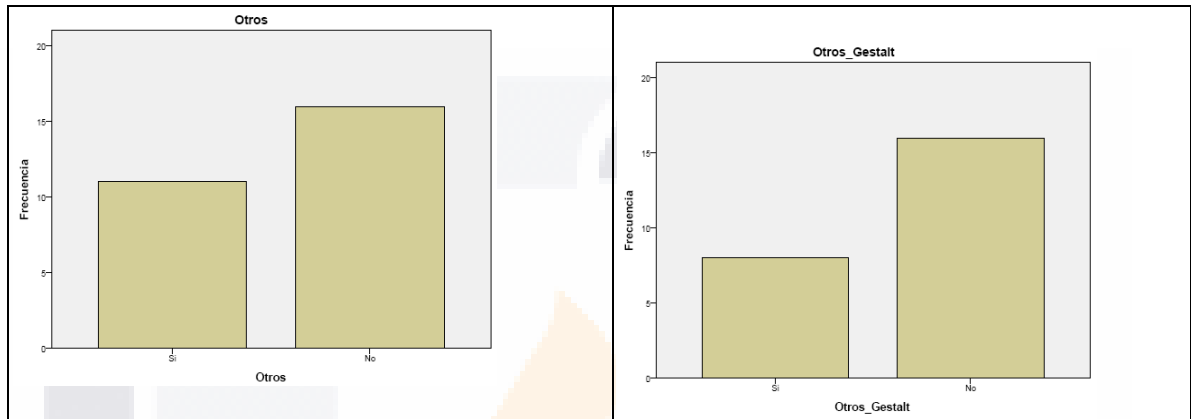


Figura 28. Histogramas. Otros usos de internet

Nivel de experiencia

La mayoría de los participantes del experimento en ambos grupos se autoevaluaron con un nivel de experiencia de 3 en una escala de 1 a 5. 54.2% del grupo experimental y 81.5% del grupo de control. Por lo que se puede afirmar que el nivel de experiencia en ambos grupos se comporta de manera similar. La Figura 29 muestra gráficamente la similitud entre los grupos.

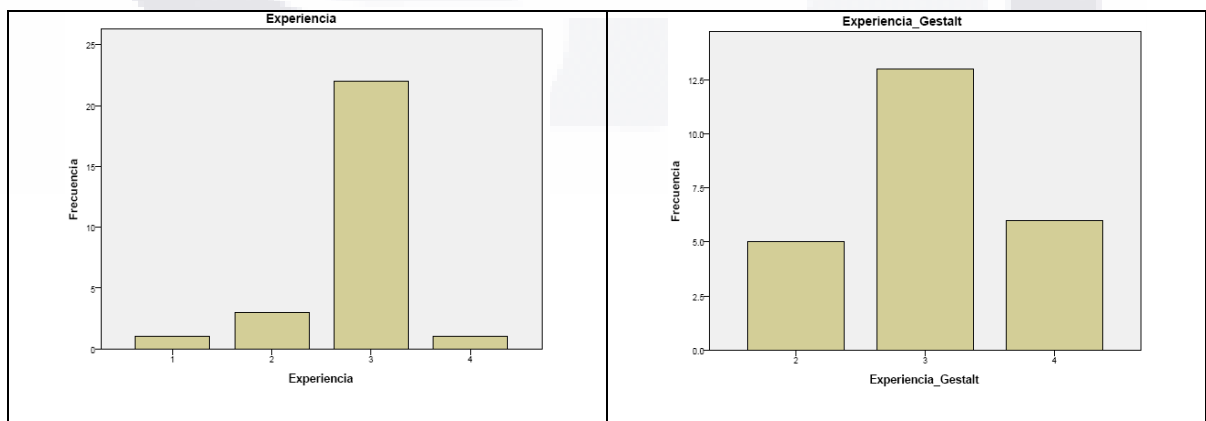


Figura 29. Histogramas. Nivel de experiencia de los usuarios

3.2 Diseño del instrumento de evaluación

La presente investigación evaluó la calidad del software tomando como elementos de medición la facilidad de uso, la facilidad de aprendizaje y el nivel de aceptación por parte del usuario. El cuestionario que se construyó está basado en los elementos previamente mencionados. La Tabla 6 muestra la relación entre las hipótesis planteadas en la investigación y los elementos de medición.

Elemento de medición	Hipótesis
Facilidad de uso	H₁: Las GUI construidas bajo los principios de la teoría Gestalt son más fáciles de usar comparadas con las GUI que se construyen sin estos principios.
Facilidad de aprendizaje	H₂: Las GUI construidas bajo los principios de la teoría Gestalt son más rápidas de aprender comparadas con las GUI que se construyen sin estos principios.
Nivel de Aceptación del usuario	H₃: Las GUI construidas bajo los principios de la teoría Gestalt son más aceptadas por los usuarios comparadas con las GUI que se construyen sin estos principios.

Tabla 6. Hipótesis de la investigación

Los elementos de medición planteados están fuertemente relacionados y definidos por los estándares de calidad ISO/IEC 9126 y el modelo de McCall, dentro de un solo factor: Facilidad de uso, a continuación en la Tabla 7 se muestra la relación entre ambos modelos y dicho factor.

<p>ISO/IEC 9126 -1</p> <p>McCall</p> <p>Facilidad de uso: El esfuerzo necesario para aprender, operar y preparar los datos de entrada de un programa e interpretar la salida.</p>	<p>Facilidad de uso: La capacidad de un producto de software para ser comprendido, aprendido y atractivo para el usuario en condiciones específicas de uso</p>	<p>Facilidad de comprensión (understability)</p>	<p>Facilidad de aprendizaje (learnability)</p>	<p>Operabilidad (Operability)</p>	<p>Atractividad (Attractiveness)</p>	
	<p>La capacidad del producto software para permitir al usuario entender si el software es adecuado, y cómo puede ser usado para tareas particulares y condiciones de uso</p>	<p>La capacidad del producto software para permitir al usuario aprender su aplicación</p>	<p>La capacidad de un producto de software para permitir al usuario operarlo y controlarlo.</p>	<p>La capacidad del producto software para ser atractivo para el usuario</p>		
<p>Familiaridad (Training)</p>	<p>Los atributos del software que proporcionan transición de la operación en curso o de familiarización inicial</p>			<p>✓</p>	<p>✓</p>	
<p>Comunicación (Communicativeness)</p>	<p>Los atributos del software que proveen las entradas y salidas útiles que pueden ser asimiladas.</p>	<p>✓</p>	<p>✓</p>			
<p>Operabilidad (Operability)</p>	<p>Los atributos del software que determinan el funcionamiento y procedimientos relacionados con el funcionamiento del software.</p>		<p>✓</p>	<p>✓</p>		

Tabla 7. Factor Facilidad de uso, ISO/IEC 9126 - McCall

De ésta manera, se tienen identificados los siguientes 5 factores: Facilidad de comprensión, Operabilidad, Atractividad, Facilidad de aprendizaje y Aceptación del usuario. Para determinar cómo evaluar cada factor se recurrió a los principios de Diseño de Interacción de Tognazzini (2001), donde se describen un conjunto de características fundamentales que deben presentar todas las interfaces, las diez heurísticas de usabilidad de Nielsen (2005) y las Reglas de Oro de Theo Mandel (1997). En la Tabla 8 se pueden observar cada uno de los atributos que componen cada factor, de acuerdo a los principios mencionados.

Elemento		Descripción		Factor
Consistencia	Organización	Toda la información visual debe estar organizada de acuerdo con un estándar de diseño que se mantenga en todas las presentaciones de pantalla.		Facilidad de comprensión Atractividad Facilidad de aprendizaje
		La interfaz debe acentuar un diseño simple e intuitivo que organice el contenido y la funcionalidad en categorías obvias para el usuario.		
	Operaciones	Siempre que sea posible, la interfaz debe ser consistente en el sentido de que las operaciones comparables se activan de la misma forma.		Operabilidad Facilidad de comprensión Facilidad de aprendizaje
	Color	Uniformidad y combinación de colores		Atractividad Facilidad de aprendizaje Facilidad de comprensión
	Lenguaje	En situaciones similares, se debe utilizar terminología idéntica en avisos, menús y pantallas de ayuda, y se deben usar comandos coherentes en todas partes. El sistema debe hablar el lenguaje de los usuarios, con palabras, frases y conceptos familiares para el usuario, los términos orientados al sistema. Sigue las convenciones del mundo real, haciendo que la información aparezca en un orden natural y lógico.		Facilidad de comprensión Facilidad de aprendizaje
Comunicación	Legibilidad De Texto	Tamaño Tipografía Color	Facilidad del texto para poder leerse	Facilidad de comprensión
	Claridad de redacción	Facilidad para comprender la redacción del texto		
	Lenguaje	Uso de términos y palabras familiares para el usuario		
	Símbolos	Uso de símbolos e íconos que representen cierta funcionalidad		Facilidad de comprensión Atractividad
Dar Control al usuario	Modos de Interacción	Debe darse al usuario una opción fácil de interactuar entre diferentes funcionalidades del sistema. Ejemplo: entrar y salir sin esfuerzo		Operabilidad

	<i>Manipulación de objetos</i>	El usuario siente que tiene el control cuando manipula los objetos necesarios para realizar una tarea de manera parecida a como lo haría con un objeto material.		
Carga de Memoria en el usuario	<i>Pistas visuales</i>	Proporcionar al usuario pistas visuales que le permitan reconocer acciones anteriores sin tener que recordarlas.		Facilidad de aprendizaje
	<i>Valores estándar</i>	Permitir al usuario regresar a los valores dados por defecto por el sistema		
	<i>Intuitividad</i>	Facultad para comprender las cosas instantáneamente, sin necesidad de razonamiento		
	<i>Metáforas</i>	Permitir al usuario depender de pistas visuales que comprende bien, ya que las asocia con el mundo real, en lugar de memorizar una misteriosa secuencia de interacciones		
	<i>Organización jerárquica</i>	La información sobre una tarea, un objeto o algún comportamiento debe presentarse primero en un grado alto de abstracción. Después de que el usuario se interese por seleccionar algo, deben presentarse más detalles		Facilidad de aprendizaje Operabilidad
Control de errores	<i>Recuperación de errores</i>	La interfaz debe incluir mecanismos para permitir a los usuarios recuperarse de los errores.		Operabilidad
	<i>Claridad del mensaje de error</i>	El mensaje debe describir claramente el problema en un lenguaje que el usuario entienda.		
	<i>Utilidad del mensaje de error</i>	El mensaje debe proporcionar consejos constructivos sobre la manera de recuperarse del error.		
Navegabilidad		Navegación simple, consistente y obvia. Debe estar diseñada de modo que sea intuitiva y predecible.		Operabilidad
Apariencia Visual	<i>Color</i>	<i>Fondo Combinación</i>	Uso adecuado del color	Atractividad
	<i>Equilibrio de elementos visuales</i>	Ecuanimidad de texto, tablas, gráficos, imágenes.		Atractividad
	<i>Sobrecarga</i>	El amontonamiento dificulta que el usuario identifique la información o características necesarias y crea un caos visual desagradable		Atractividad Facilidad de comprensión
	<i>Resaltar contenido</i>	Destacar la información de mayor relevancia		Atractividad Facilidad de comprensión
	<i>Simplicidad</i>	Evitar efectos visuales extremos, animaciones entrometidas, etc. Buscar la moderación y simplicidad		Atractividad Facilidad de aprendizaje Facilidad de comprensión

Tabla 8. Características fundamentales de una GUI y factores de medición

Finalmente, en la Tabla 9 se definen los elementos que componen el cuestionario final, también se consultaron otros cuestionarios como MUM (Multidimensional Usability Model Questionnaire Template), (Cole Ian, 2008) el cual fue utilizado

para la evaluación de entornos virtuales de aprendizaje y está basado en el QUIs (The Questionnaire for User Interaction Satisfaction), el cual fue diseñado para evaluar aspectos subjetivos de la satisfacción de usuarios.

Factor	Dimensión	Indicador	Pregunta
Facilidad de comprensión Definición según ISO/IEC 9126: La capacidad de un producto de software para permitir al usuario entender si el software es adecuado, y cómo puede ser usado para tareas particulares en condiciones específicas de uso.	Lenguaje	Grado en el que el uso de términos y palabras resultan familiares para el usuario	Las palabras y términos utilizados en la interfaz me resultan familiares
	Legibilidad del texto (tamaño de letra)	Grado en el que el texto se puede leer	No tengo problemas para leer el texto presentado en la interfaz debido a que el tamaño de la letra es adecuado
	Legibilidad del texto (tipo de letra)	Grado en el que el texto se puede leer	No tengo problemas para leer el texto presentado en la interfaz debido a que el tipo de letra es adecuado
	Legibilidad del texto (color de letra)	Grado en el que el texto se puede leer	No tengo problemas para leer el texto presentado en la interfaz debido a que el contraste entre el color de fondo y el color del texto es adecuado
	Redacción del texto	Grado de comprensión del texto en torno a la redacción	No tengo problemas para comprender el texto presentado en la interfaz debido a que la redacción es adecuada
	Significado de los íconos	Grado de entendibilidad del significado de los íconos	Me resulta entendible el significado de la funcionalidad que representan los iconos que contiene la interfaz
Facilidad de operación Definición según ISO/IEC 9126: La capacidad de un producto de software para permitir al usuario operarlo y controlarlo.	Consistencia en operaciones	Grado en el que las operaciones comparables se activan de la misma forma	Realizar tareas comparables me resulta fácil, debido a que se activan de manera similar
	Dar Control al usuario	Grado en el que el usuario se siente en control de la situación	Considero que el diseño de la pantalla me proporciona el control de las tareas que realizo
	Control de errores (recuperación)	Grado en el que la interfaz se recupera de los errores	Creo que la interfaz maneja fácilmente los errores que puedo cometer
	Control de errores (Claridad de los mensajes)	Grado de claridad de los mensajes de error	Considero que los mensajes que se presentan en la interfaz son muy claros
	Control de errores (Utilidad del mensaje)	Grado de ayuda de los mensajes de error	Me parece que la información que me muestran los mensajes es de gran ayuda

	Navegabilidad	Grado de facilidad con el que se puede navegar en el sistema	La forma de navegación en el sistema me resulta muy fácil
	Movimientos	Grado en el que la cantidad de movimientos en la interfaz, para realizar una tarea parecen suficientes	Creo que la cantidad de movimientos a hacer para ejecutar una acción en la pantalla es adecuada
Atractividad Definición según ISO/IEC 9126: La capacidad del producto software para ser atractivo para el usuario	Organización de elementos	Grado en el que los elementos de la interfaz están organizados de manera simple	Considero que los elementos de la interfaz están organizados de manera simple
	Combinación de Color	Grado de gusto de la combinación de colores en la interfaz	Me parece muy agradable la combinación de colores de la interfaz
	Color de fondo	Grado de adecuación del color de fondo	El color del fondo me parece adecuado
	Símbolos e íconos	Grado de Atractividad de los símbolos e íconos	Me parece que los símbolos e íconos de la interfaz son muy atractivos
Facilidad de aprendizaje Definición según ISO/IEC 9126: La capacidad del producto software para permitir al usuario aprender su aplicación	Pistas visuales	Grado en el que las pistas visuales contribuyen a recordar acciones anteriores	Considero que los elementos visuales que contiene la interfaz que me ayudan a reconocer acciones anteriores
	Intuitividad	Grado en que se comprenden las cosas instantáneamente, sin necesidad de razonamiento	Me parece fácil comprender cómo realizar las tareas a través la interfaz
		Grado en que se puede recordar como acceder a las tareas del sistema	Me parece fácil recordar cómo acceder a las tareas a través de la interfaz
	Organización jerárquica	Grado en el que la información se presenta de manera jerárquica	Considero que la información presentada en la interfaz se encuentra ordenada de manera jerárquica
Equilibrio de elementos	Grado en el que la cantidad de elementos contenidos en la interfaz es suficiente	Considero que la cantidad de elementos que contiene la interfaz es suficiente	
Aceptación del usuario	Beneficios percibidos	Grado en el que el usuario considera benéfica la interfaz	Me parece que el diseño de la interfaz me ayuda a realizar la tarea que deseo
	Confianza	Grado de confianza que inspira la interfaz en el usuario	Considero que el diseño de la interfaz me inspira confianza
	Satisfacción de necesidades	Grado en el que la interfaz satisface las necesidades del usuario	Me parece que el diseño satisface mis necesidades

Tabla 9. Desarrollo de preguntas

3.3 Modelo de Investigación

De acuerdo a los factores identificados, que intervienen en la calidad de las GUIs, se clasificaron las principales características que deben contener éstas, según los principios de diseño mencionados anteriormente, en la Figura 30 se observa dicha clasificación, la cual constituye el modelo utilizado en la presente investigación.

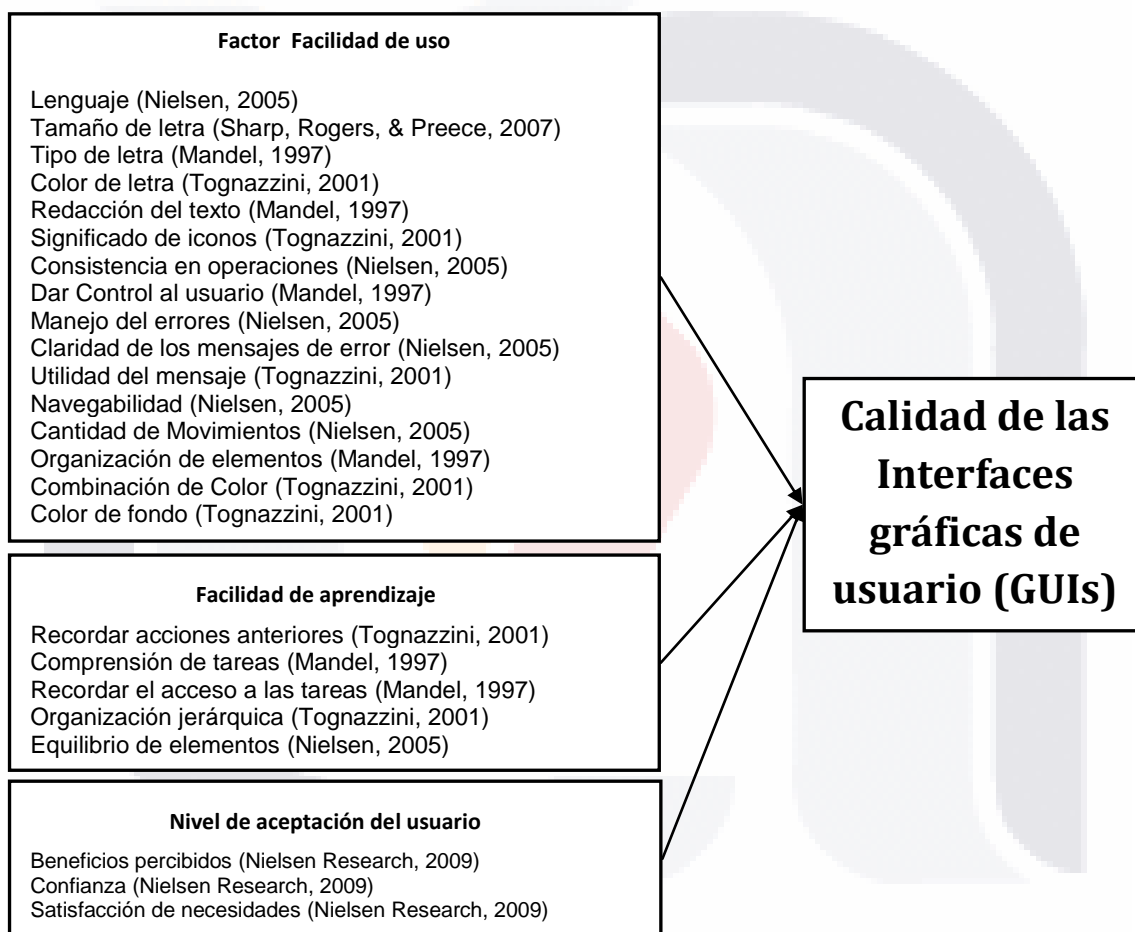


Figura 30. Modelo Propuesto para la presente investigación

3.4 Estudio piloto

Con la finalidad de evaluar los aspectos de interés que se exponen y dar respuesta a las hipótesis planteadas en la presente investigación, se diseñó un

instrumento de evaluación, el cual se validó por medio de un estudio piloto con el objetivo de analizar si dicho instrumento está correctamente diseñado.

El estudio piloto se realizó en la Universidad Autónoma de Aguascalientes con alumnos del Centro de Ciencias Económicas y Administrativas de la carrera de Licenciado en Administración de empresas de 5º semestre. El grupo estuvo constituido por un total de 33 alumnos, de los cuales el día de la aplicación solo asistieron 28 (84.38% de participación).

El estudio se realizó en un salón de clases de la misma universidad. Antes de entregar el cuestionario a los alumnos, se les explicó cuál era el objetivo de la evaluación, la relevancia que éste tenía para la investigación que se lleva a cabo y la importancia de su colaboración. Al entregar el cuestionario se procedió a leer las instrucciones en voz alta de manera grupal, al terminar se les cuestionó si tenían alguna duda acerca de las instrucciones o si alguna de éstas les resultaba poco clara o confusa, a lo que se tuvo una respuesta negativa, de lo cual se puede deducir que las instrucciones resultaron adecuadas. Se les resaltó la importancia de responder con absoluta sinceridad y que el cuestionario era de naturaleza anónima.

Se les dio un tiempo aproximado de 30 minutos para contestar el cuestionario, el cual estaba integrado por seis secciones. La primera parte tiene el fin de recabar información demográfica del participante, la segunda sección permite recolectar información relacionada con los cinco factores de calidad de software a evaluar, los cuales son: Facilidad de comprensión, Facilidad de aprendizaje, Facilidad de operación, Atractividad y Aceptación del usuario. En total el cuestionario estuvo conformado por 34 reactivos, de los cuáles 9 pertenecían a información demográfica y los 25 restantes estuvieron distribuidos entre los 5 diferentes factores. Todas las preguntas, excepto las demográficas, tuvieron 7 opciones de respuesta, basadas en la escala Likert. Al final de cada sección se agregó un espacio para comentarios.

Al entregar cada participante el cuestionario que respondió, se le preguntó si tuvo algún tipo de inconveniente o problema para responderlo, o si tenía algún comentario que agregar, de cual solo se obtuvo un comentario acerca de las instrucciones; el participante mencionó que “las instrucciones resultaban demasiado obvias.”

Al realizar la revisión de los cuestionarios se encontró que se respondieron los 28 entregados, solo se presentó un caso en el que una pregunta de tipo demográfico no fue contestada.

3.4.1 Análisis de reactivos del instrumento de evaluación

Para verificar la validez de cada pregunta relacionada con los cinco factores de calidad se probó que cada una de ellas tuviera un comportamiento normal. Para tal fin, se realizó un análisis de dispersión de las respuestas mediante técnicas estadísticas. A continuación se presentan los resultados de cada pregunta agrupadas para cada factor de calidad.

3.4.1.1 Factor Facilidad de comprensión

La Tabla 10 muestra la estadística descriptiva de todas las sub-características que conforman al factor Facilidad de comprensión. En ella se puede identificar que los datos de cada una de ellas se comportan siguiendo una distribución normal, por lo que las fluctuaciones en los valores son las esperadas. Asimismo, los participantes no tuvieron problemas con la forma ni el fondo de cada una de las preguntas correspondientes. Por lo tanto, con respecto a esto el instrumento está bien diseñado. Para corroborar estos valores se puede analizar la Figura 31, la cual muestra los distintos histogramas de cada sub-característica.

	Lenguaje Familiar	Tamaño Texto	Tipo Letra	Contraste Color	Redacción	Significado Símbolos
Válidos	28	28	28	28	28	28
Faltantes	0	0	0	0	0	0
Media	2.18	2.25	1.79	2.21	2.11	2.50
Error Estándar de la Media	.179	.234	.157	.202	.195	.233
Mediana	2.00	2.00	2.00	2.00	2.00	2.00
Moda	2	2	2	2	2	2
Desviación Estándar	.945	1.236	.833	1.067	1.031	1.232
Varianza	.893	1.528	.693	1.138	1.062	1.519
Asimetría	.469	1.513	1.265	.920	1.302	.703
Error Estándar de Asimetría	.441	.441	.441	.441	.441	.441
Curtosis	-.505	2.420	1.882	.479	1.544	-.179
Error Estándar de Curtosis	.858	.858	.858	.858	.858	.858
Rango	3	5	3	4	4	4
Mínimo	1	1	1	1	1	1
Máximo	4	6	4	5	5	5

Tabla 10. Estadística descriptiva. Factor: Facilidad de comprensión

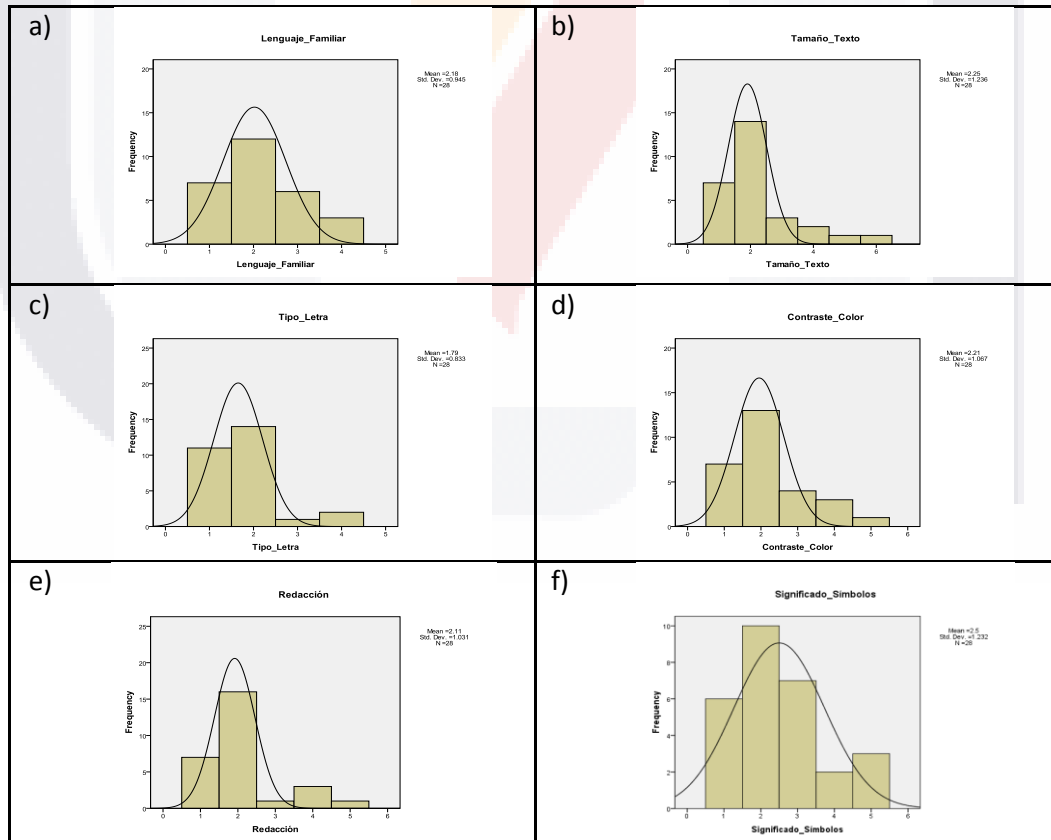


Figura 31. Histogramas. Factor: Facilidad de comprensión

3.4.1.2 Factor Facilidad de operación

En la Tabla 11 se muestra la estadística descriptiva de cada sub-característica que compone el factor Facilidad de operación. En ella se puede observar que los datos tienen un comportamiento normal, las fluctuaciones que se presentan son las esperadas. De la misma manera, los participantes no tuvieron problemas con el fondo y la forma de cada una de las preguntas planteadas. Por lo anterior se puede decir que el instrumento está bien diseñado. En la Figura 32 se observa cada histograma perteneciente a cada sub-característica, en ellos se puede corroborar de manera gráfica el comportamiento de los datos.

	Tareas Similares	Ctrl Usuario	Ctrl Error	Claridad Msje	Ayuda Msje	Navegación	Mov. Navegación
Válidos	28	28	28	28	28	28	28
Faltantes	0	0	0	0	0	0	0
Media	2.29	2.29	2.61	2.21	2.39	2.21	2.50
Error Estándar de la Media	.198	.191	.243	.188	.208	.202	.174
Mediana	2.00	2.00	2.00	2.00	3.00	2.00	3.00
Moda	2	2	2	2	3	2	3
Desviación Estándar	1.049	1.013	1.286	.995	1.100	1.067	.923
Varianza	1.101	1.026	1.655	.989	1.210	1.138	.852
Asimetría	.825	.982	.925	.752	.203	.920	-.304
Error Estándar de Asimetría	.441	.441	.441	.441	.441	.441	.441
Curtosis	.378	.807	.699	.827	-.478	.479	-.703
Error Estándar de Curtosis	.858	.858	.858	.858	.858	.858	.858
Rango	4	4	5	4	4	4	3
Mínimo	1	1	1	1	1	1	1
Máximo	5	5	6	5	5	5	4

Tabla 11. Estadística descriptiva. Factor: Facilidad de operación

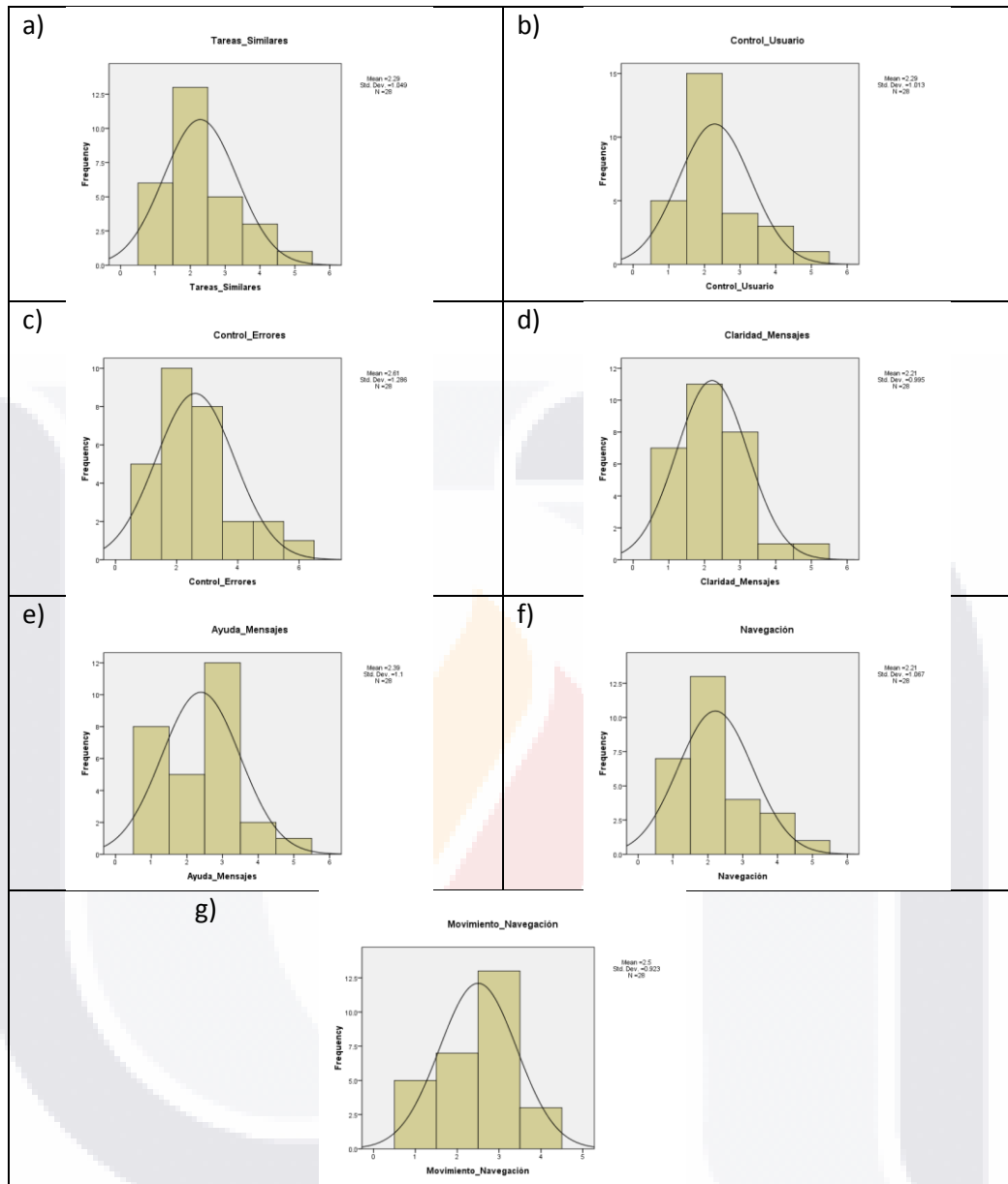


Figura 32. *Histogramas. Factor: Facilidad de operación*

3.4.1.3 Factor Atractividad

A continuación en la Tabla 12 se presenta la estadística descriptiva de las subcaracterísticas que conforman el factor Atractividad. En dicha tabla se puede apreciar que los datos de cada una de ellas tienen un comportamiento normal, por lo que las fluctuaciones en los valores son las esperadas. Asimismo, los participantes no manifestaron problema alguno con la forma ni el fondo de cada

una de las preguntas correspondientes. De ésta manera, con respecto a este punto el instrumento de evaluación está diseñado de manera correcta. Los histogramas mostrados en la Figura 33 corroboran de manera gráfica el comportamiento de los datos de cada sub-característica.

	Organización Elementos	Combinación colores	Color Fondo	Atractividad de símbolos
Válidos	28	28	28	28
Faltantes	0	0	0	0
Media	2.18	2.50	2.46	2.61
Error Estándar de la Media	.163	.221	.189	.208
Mediana	2.00	2.00	2.00	2.50
Moda	2	2	2	2
Desviación Estándar	.863	1.171	.999	1.100
Varianza	.745	1.370	.999	1.210
Asimetría	.749	1.268	.825	.516
Error Estándar de Asimetría	.441	.441	.441	.441
Curtosis	.348	2.074	.277	-.044
Error Estándar de Curtosis	.858	.858	.858	.858
Rango	3	5	4	4
Mínimo	1	1	1	1
Máximo	4	6	5	5

Tabla 12. Estadística descriptiva. Factor: Atractividad

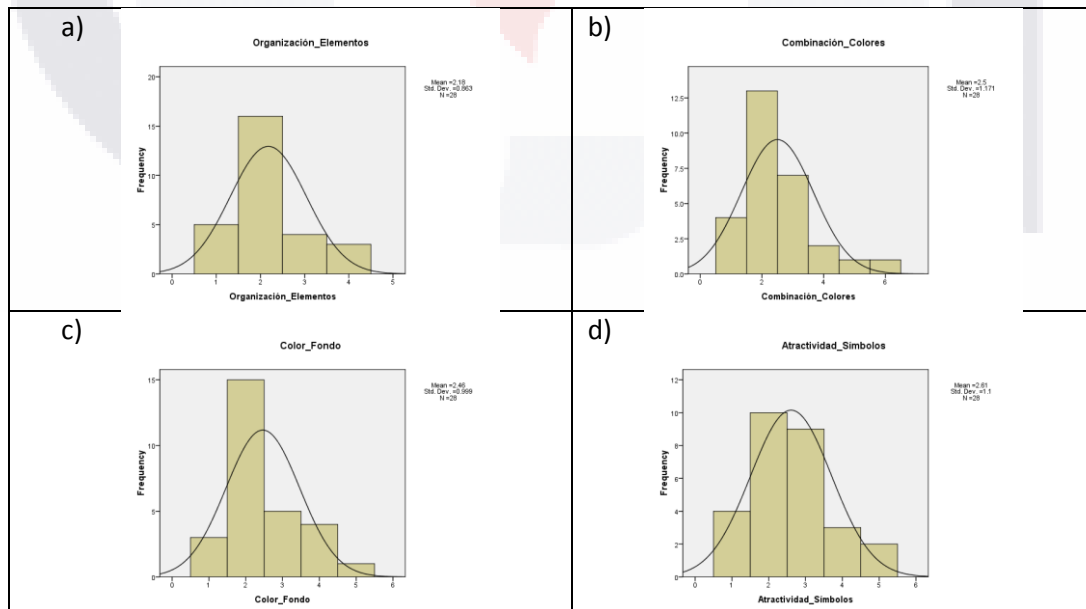


Figura 33. Histogramas. Factor: Atractividad

3.4.1.4 Factor Facilidad de aprendizaje

A continuación en la Tabla 13 se presenta la estadística descriptiva correspondiente a las sub-características que componen el factor Facilidad de aprendizaje. Se puede apreciar en dicha tabla que los datos se distribuyen de manera normal, por lo que las variaciones en los valores son las que se esperaban. En la aplicación de la prueba piloto los participantes no manifestaron problema alguno para responder las preguntas planteadas. Por esto se puede decir que el instrumento está diseñado de manera correcta. En la Figura 34 se pueden observar los histogramas correspondientes a cada sub-característica, los cuáles muestran la distribución de los datos.

	Recordar Acciones	Tareas Fáciles de comprender	Tareas Fáciles de recordar	Información Jerárquica	Número de elementos
Válidos	28	28	28	28	28
Faltantes	0	0	0	0	0
Media	2.43	2.36	2.29	2.39	2.25
Error Estándar de la Media	.181	.258	.246	.259	.216
Mediana	2.00	2.00	2.00	2.00	2.00
Moda	2	2	2	1	2
Desviación Estándar	.959	1.367	1.301	1.370	1.143
Varianza	.921	1.868	1.693	1.877	1.306
Asimetría	.354	1.355	1.379	.893	1.394
Error Estándar de Asimetría	.441	.441	.441	.441	.441
Curtosis	-.732	1.876	1.839	.272	3.029
Error Estándar de Curtosis	.858	.858	.858	.858	.858
Rango	3	5	5	5	5
Mínimo	1	1	1	1	1
Máximo	4	6	6	6	6

Tabla 13. Estadística descriptiva. Factor: Facilidad de aprendizaje

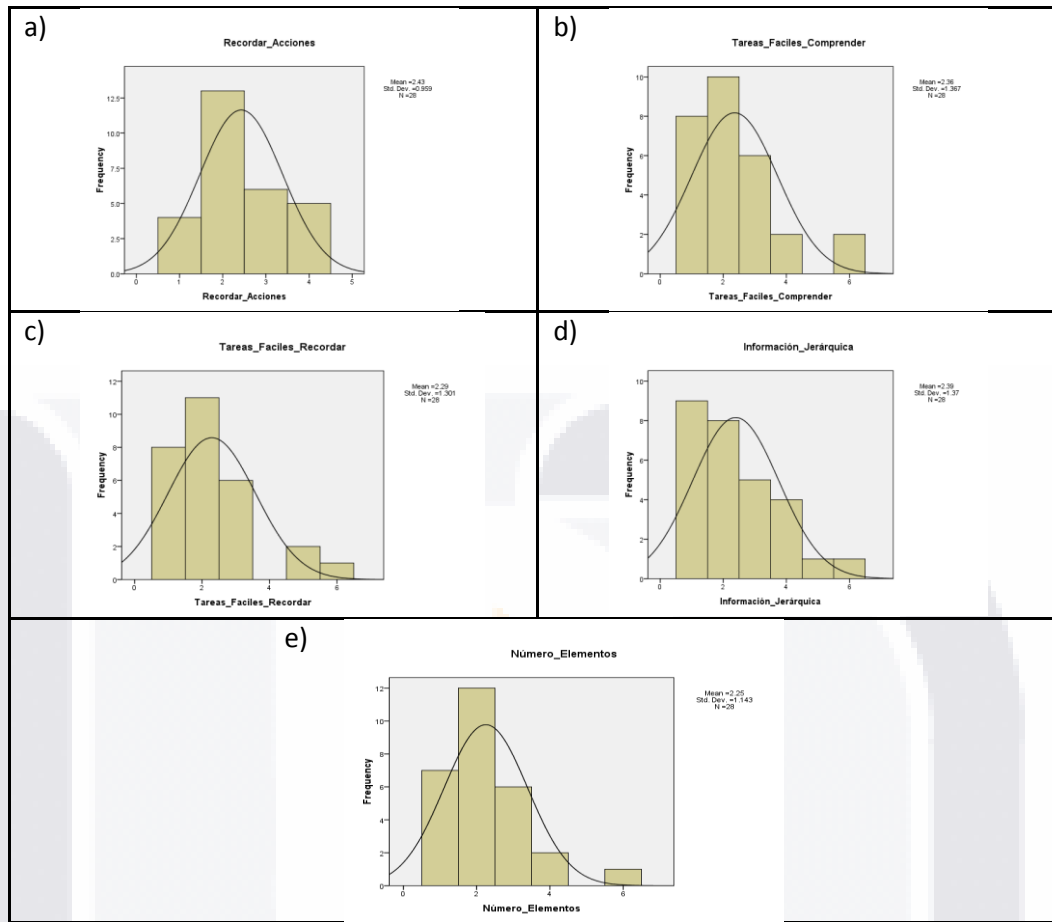


Figura 34. Histogramas. Factor: Facilidad de aprendizaje

3.4.1.5 Factor Aceptación del usuario

En la Tabla 14 se muestra la estadística descriptiva de cada sub-característica que compone el factor Aceptación del usuario. En ella se puede observar que los datos tienen un comportamiento normal, las fluctuaciones que se presentan son las esperadas. De la misma manera, los participantes no tuvieron problemas con el fondo y la forma de cada una de las preguntas planteadas. Por lo anterior se puede decir que el instrumento está bien diseñado. En la Figura 35 se observa cada histograma perteneciente a cada sub-característica, en ellos se puede corroborar de manera gráfica el comportamiento de los datos.

	Beneficio percibido	Confianza	Satisfacción de necesidades
Válidos	28	28	28
Faltantes	0	0	0
Media	2.46	2.39	2.46
Error Estándar de la Media	.182	.201	.174
Mediana	2.00	2.00	2.00
Moda	2	2	2
Desviación Estándar	.962	1.066	.922
Varianza	.925	1.136	.851
Asimetría	.513	1.092	.112
Error Estándar de Asimetría	.441	.441	.441
Curtosis	.542	1.002	-.685
Error Estándar de Curtosis	.858	.858	.858
Rango	4	4	3
Mínimo	1	1	1
Máximo	5	5	4

Tabla 14. Estadística descriptiva. Factor: Aceptación de usuario

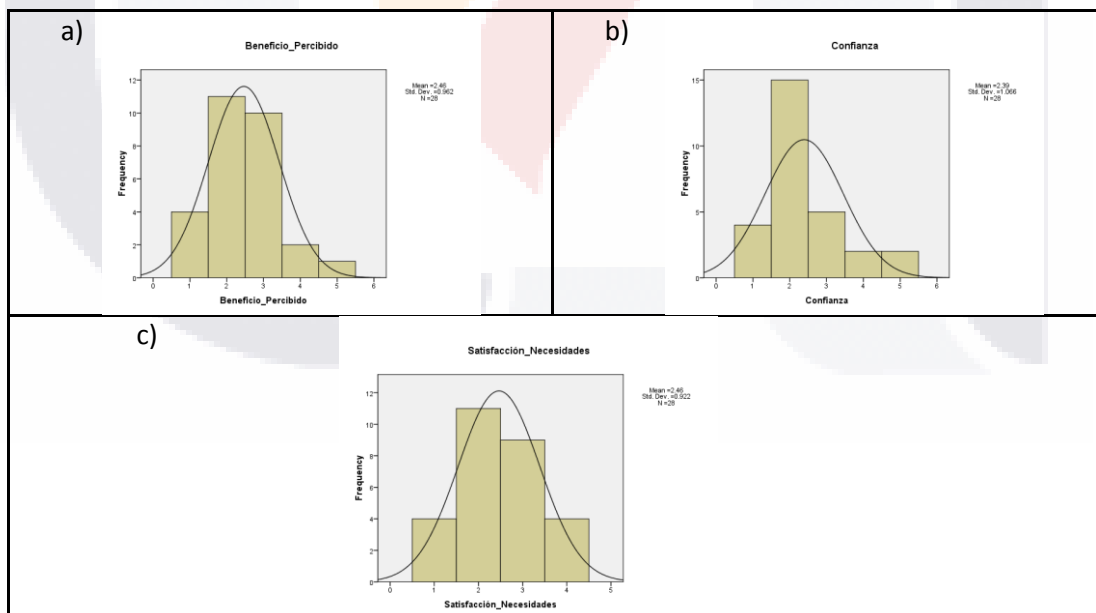


Figura 35. Histogramas. Factor: Aceptación de usuario

3.5 Resumen de validez de los reactivos del instrumento

Adicionalmente, se realizó un análisis de confiabilidad por medio del Alfa de Cronbach. Por medio de este se puede comprobar la cohesividad interna que tienen los elementos de un mismo factor. Si los elementos de un factor no son cohesivos, el instrumento tienen un problema y, por lo tanto, dicho factor debe ser revisado. Para este fin, se utilizó como valor mínimo del Alfa el sugerido en la literatura (Hair, Anderson, Tatham, & Black, 1998) (Kerlinger & Lee, 2000) el cual es de .7. La Tabla 15 muestra los valores obtenidos de Alfa. En ella se puede observar que todos los factores sobrepasan el mínimo establecido. En consecuencia, el instrumento si mide lo que se intenta evaluar a través de él.

Factor	Valor de Alfa	Número de elementos
Facilidad de comprensión	.890	6
Facilidad de aprendizaje	.864	5
Facilidad de operación	.918	7
Atractividad	.858	4
Aceptación	.815	3

Tabla 15. Valores de Alfa de Cronbach

Finalmente, una vez hecho estos análisis se tiene la certeza de que el instrumento utilizado en el experimento final está correctamente diseñado.

Capítulo 4

Análisis de resultados

4.1 Prueba de medias

Con el fin de probar las hipótesis planteadas para el presente estudio (ver Capítulo 1 sección 1.6) se utilizó una serie de pruebas mediante la *t* de Student con muestras independientes. La prueba se aplicó con un intervalo de confianza del 95%. El instrumento de evaluación está dividido en 5 secciones, cada una corresponde a un sub-factor identificado en la literatura. Los resultados de las pruebas se describen a continuación.

4.1.1 Factor Facilidad de comprensión

En el sub-factor *Lenguaje familiar* se encontró una diferencia **no significativa** ($p=0.748$) entre ambas formas de diseño de GUIs. Además, se encontró que Gestalt (calidad promedio = 1.63) genera una mejor percepción por parte de los usuarios comparada sin Gestalt (calidad promedio = 2.04), es decir, existe una diferencia de 0.412 (6.9%). Por lo tanto Gestalt produce mejores resultados que el diseño sin Gestalt pero no de manera significativa.

En la Figura 36 se muestran los histogramas correspondientes a las evaluaciones, del lado izquierdo (sistema diseñado sin Gestalt) se puede observar que la mayoría de los valores se concentran en 2 (51.9%); del lado derecho se observa la evaluación del sistema diseñado con Gestalt, donde la mayoría de los valores se centran en 1 (50%). De esta manera, se muestra gráficamente la diferencia entre

ambos diseños, la cual favorece al sistema diseñado con Gestalt, pero no de manera significativa estadísticamente hablando. Por lo tanto, la hipótesis $H_{1,A1}$ es rechazada.

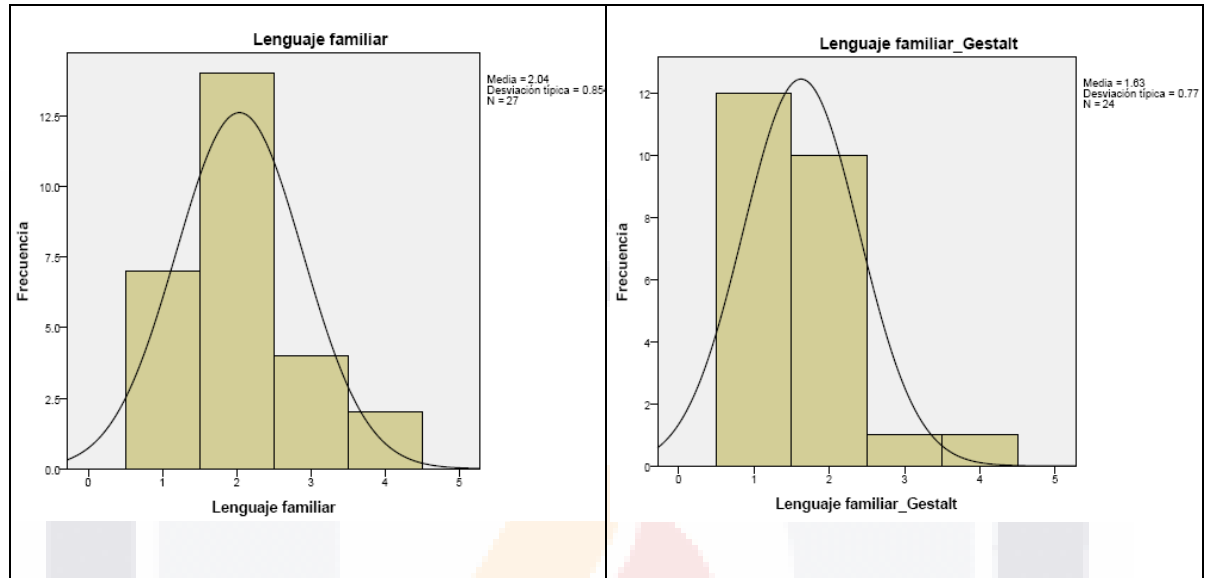


Figura 36. Histogramas. Sub-factor: Lenguaje familiar

En el sub-factor **Tamaño del texto** se observó una diferencia **significativa** ($p=0.004$) entre las dos formas de diseño de GUIs. Asimismo, se encontró que las GUIs con Gestalt (calidad promedio = 1.63) fueron evaluadas de manera más favorable en comparación con las GUIs sin Gestalt (calidad promedio = 2.37), es decir, se tiene una diferencia significativa del 12.4% aproximadamente. Por lo anterior se puede deducir que Gestalt genera mejores resultados comparado con el diseño sin Gestalt de manera significativa.

En la Figura 37 se puede observar del lado izquierdo el histograma de los valores del sistema sin Gestalt, los cuales se concentran en el valor 1 (40.7%), también se observa una mayor dispersión entre los demás valores, que indica que las opiniones fueron más variadas, comparado con el histograma del lado derecho que pertenece al sistema diseñado con Gestalt, en el cual se muestra que la mayor concentración de datos se encuentra en 1 y 2 (91.7%), lo que demuestra que dicho sistema causó una mejor opinión a los usuarios. Por lo tanto, la hipótesis $H_{1,A2}$ es aceptada.

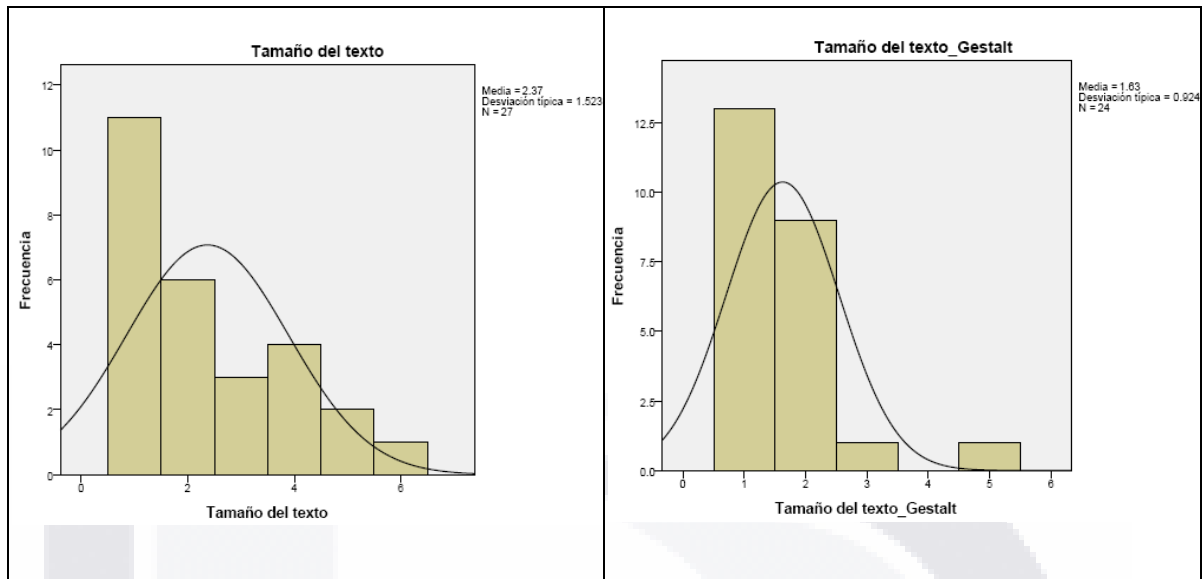


Figura 37. Histogramas. Sub-factor: Tamaño del texto

En torno al sub-factor **Tipo de letra** se encontró que las GUIs diseñadas bajo los principios de Gestalt (calidad promedio = 1.42) generan una mejor percepción por parte de los usuarios finales en comparación con el diseño sin Gestalt (calidad promedio = 1.78), pese a esto, se observa que dicha diferencia **no resulta significativa** ($p=0.149$), la cual es de 0.361 (6%). Por lo tanto se puede deducir que el empleo de Gestalt no influye significativamente en la calidad de dicho sub-factor.

La Figura 38 muestra que en el sistema diseñado sin Gestalt (histograma del lado izquierdo), la mayoría de los valores se sitúan entre 1 y 2 (70.4%), similar a la cantidad de datos ubicados en 1 del lado derecho (75%) (Sistema con Gestalt). La diferencia refleja que el sistema diseñado con Gestalt recibió una evaluación levemente más favorable respecto al factor tipo de letra, dicha diferencia no resulta significativa. Por lo tanto, la hipótesis $H_{1,A3}$ es rechazada.

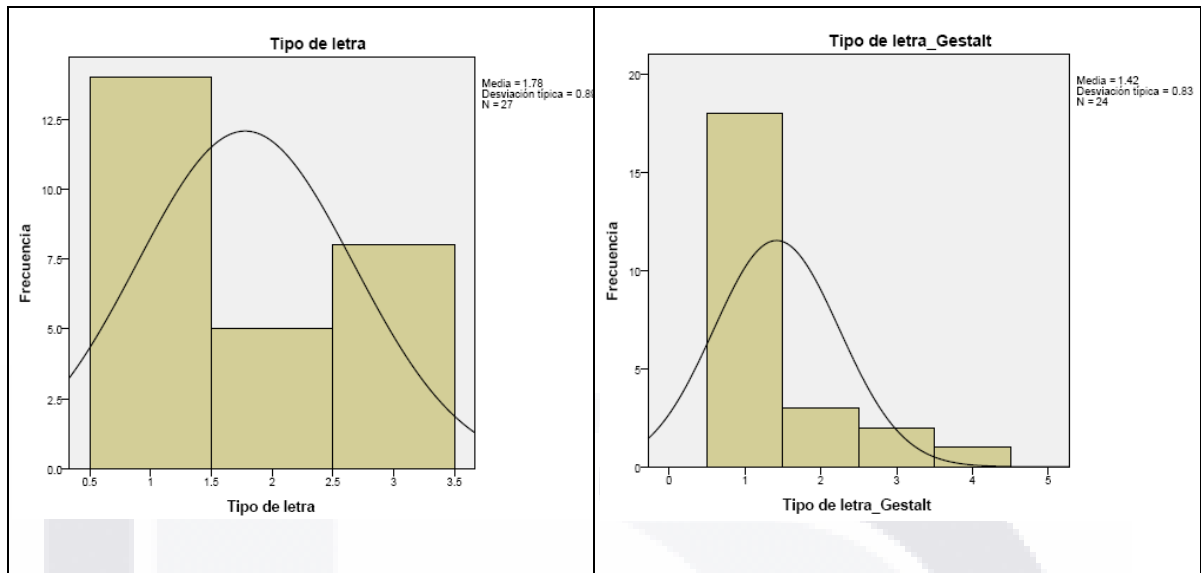


Figura 38. Histogramas. Sub-factor: Tipo de letra

En el sub-factor **Contraste de color de fondo y texto** se encontró una diferencia **significativa** ($p=0.002$) entre ambas formas de diseño de GUIs. Además, se encontró que Gestalt (calidad promedio = 1.38) genera una mejor percepción por parte de los usuarios comparada con el diseño sin Gestalt (calidad promedio = 2.85), es decir, existe una diferencia de 1.477 (24.6%). Por lo anterior se puede deducir que Gestalt genera mejores resultados comparado con el diseño sin Gestalt de manera significativa.

En la Figura 39 se puede apreciar una diferencia significativa entre los histogramas que muestran los datos del sistema sin Gestalt (lado izquierdo) y el sistema con Gestalt (lado derecho). En el primero se aprecia que las opiniones se encuentran más dispersas, a diferencia del segundo donde el 95.8 % de las opiniones se centró en los valores 1 y 2. Por lo tanto, la hipótesis $H_{1.A4}$ es aceptada.

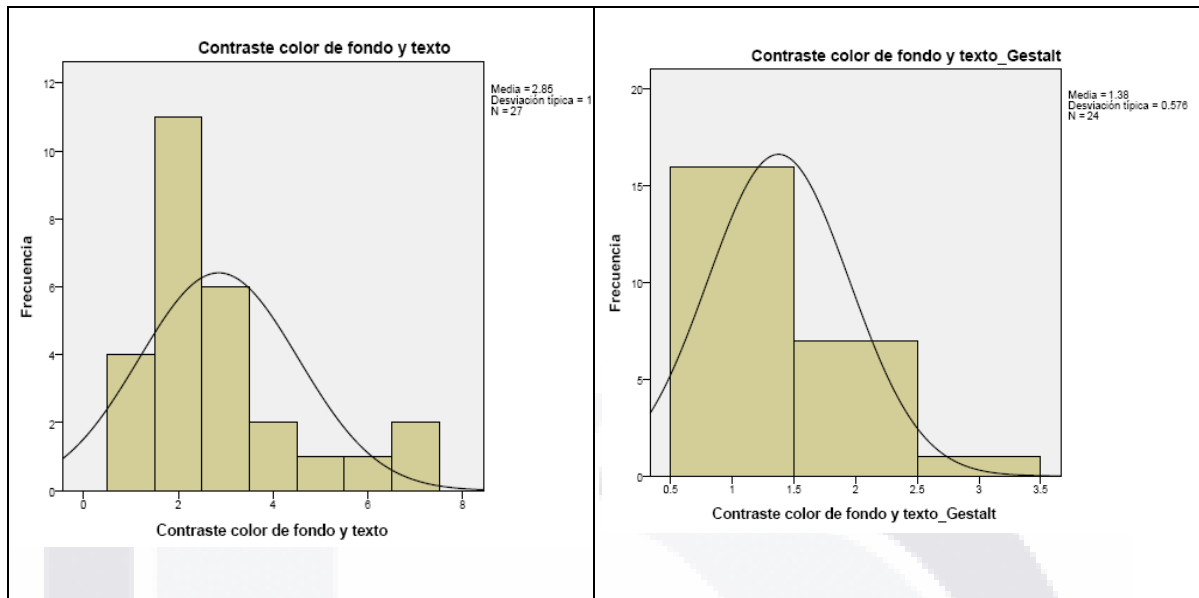


Figura 39. Histogramas. Sub-factor: Contraste de color de fondo y texto

Respecto al sub-factor **redacción del texto** se observó una diferencia **no significativa** ($p=0.312$) entre ambas formas de diseño de GUIs. Asimismo, se encontró que Gestalt (calidad promedio = 1.38) produce una mejor percepción por parte de los usuarios comparada con las que están diseñadas sin Gestalt (calidad promedio = 2.07), es decir, existe una diferencia de 0.699 (11.7%). Por lo tanto Gestalt no influye de manera significativa en la calidad de las GUIs en torno a la redacción del texto, respecto a las GUIs diseñadas sin Gestalt.

En la Figura 40 se muestra de manera grafica la evaluación de ambos sistemas (a la izquierda sistema sin Gestalt y derecha sistema con Gestalt). En el primero los datos de encuentran distribuidos de manera normal entre 1 y 4, a diferencia del grafico de la derecha, donde los valores se concentran del lado izquierdo entre 1 y 2. La evaluación resulta levemente más favorable para el sistema construido con Gestalt, sin que esta diferencia resulte significativa. Por lo tanto, la hipótesis $H_{1.A5}$ es rechazada.

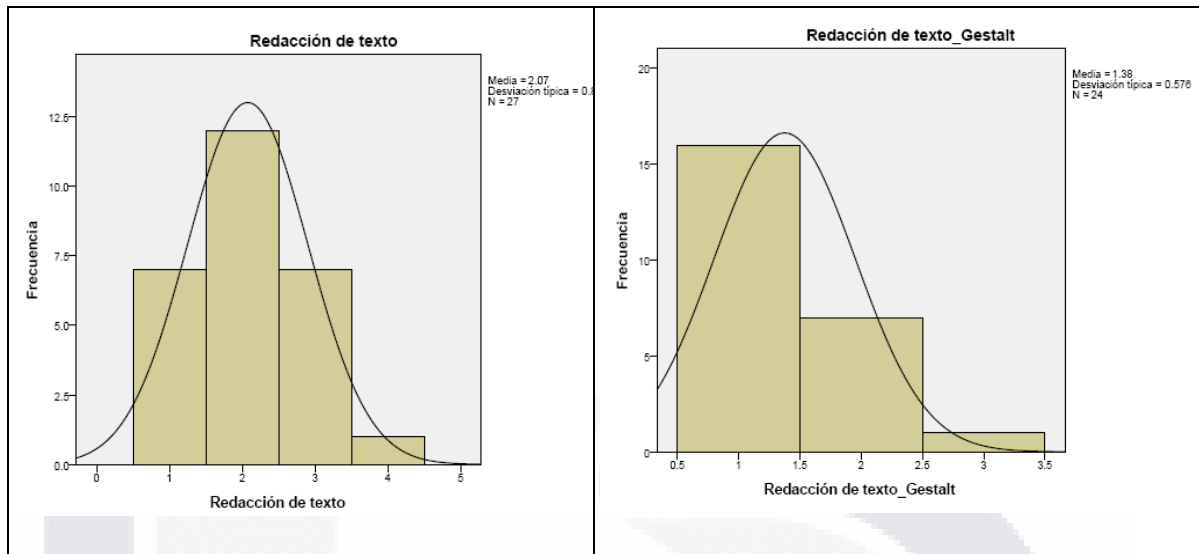


Figura 40. Histogramas. Sub-factor: Redacción de texto

En el sub-factor **significado de símbolos** se encontró una diferencia **no significativa** ($p=0.683$) entre las dos formas de diseño de GUIs. Además, se observó que Gestalt (calidad promedio = 1.79) produce una mejor opinión de los usuarios comparada con las que están diseñadas sin Gestalt (calidad promedio = 1.81), es decir, existe una diferencia de 0.023 (0.4%). Por lo tanto se puede deducir que el empleo de Gestalt no influye significativamente en la calidad de dicho sub-factor.

En la Figura 41 se muestran los histogramas correspondientes a la evaluación de ambos sistemas, del lado izquierdo (sistema sin Gestalt) se puede observar que la mayoría de los valores se concentran en 1 y 2 (77.8%), del lado derecho se observa la evaluación del sistema diseñado con Gestalt, donde los valores se centran igualmente en 1 y 2 (87.5%). La diferencia entre ambas evaluaciones favorece levemente al sistema diseñado con Gestalt pero no de manera significativa. Por lo tanto, la hipótesis $H_{1,A6}$ es rechazada.

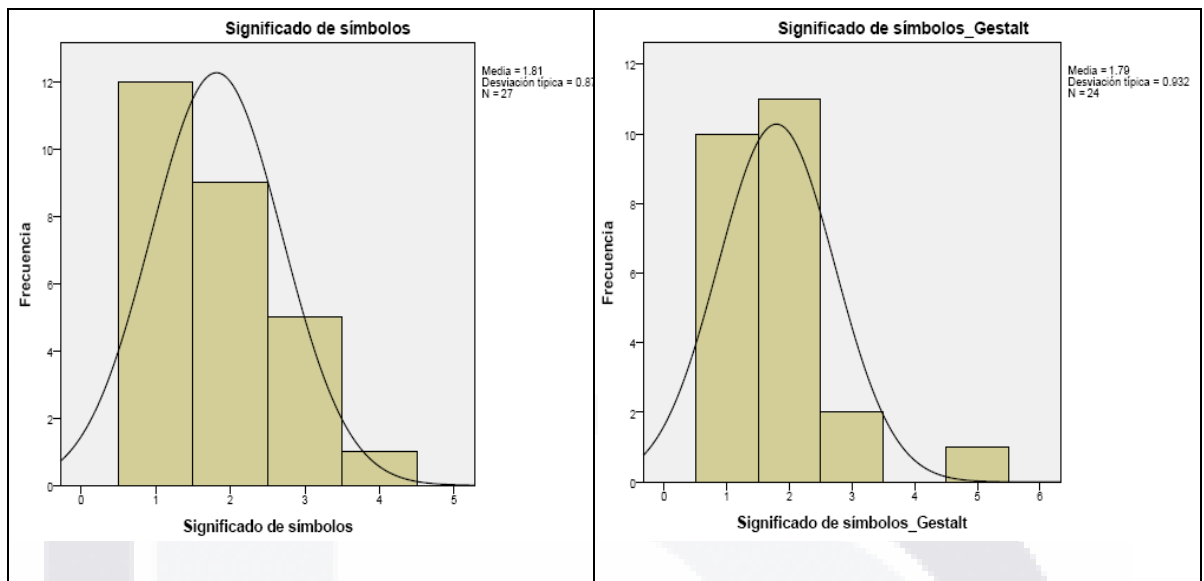


Figura 41. Histogramas. Sub-factor: Significado de símbolos

En resumen respecto al factor Facilidad de comprensión, la Figura 42 muestra la evaluación de cada uno de los subfactores, en la cual solo los subfactores Tamaño del texto y Contraste de color de fondo y texto resultaron tener una diferencia significativa en la calidad, favorable para el sistema diseñado con Gestalt. Las subfactores restantes también presentaron una diferencia favorable al sistema con Gestalt, pero éstas no resultaron ser significativas.

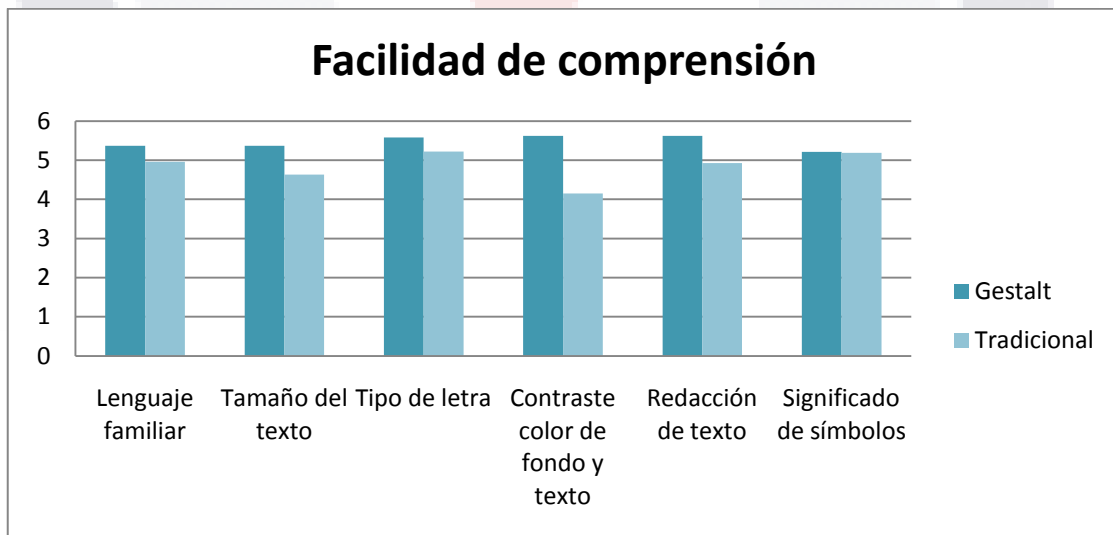


Figura 42. Gráfico de barras para comparar las medias de los sub-factores de Facilidad de comprensión.

4.1.2 Factor Facilidad de operación

Respecto al sub-factor **Tareas similares** se observó una diferencia **no significativa** ($p=0.309$) entre las dos formas de diseño de GUIs. Además, se observó que Gestalt (calidad promedio = 1.71) produce una mejor percepción en los usuarios comparada con las que están diseñadas sin Gestalt (calidad promedio = 2.15), es decir, existe una diferencia de 0.44 (7.3%). Por lo tanto Gestalt produce mejores resultados que el diseño sin Gestalt pero no de manera significativa.

La Figura 43 muestra en la parte izquierda el histograma correspondiente a la evaluación realizada para el sistema sin Gestalt, en dicho gráfico se observa la mayor cantidad de datos en el centro, a diferencia del histograma de la derecha correspondiente a la evaluación realizada para el sistema con Gestalt, en el cual la mayoría de los datos se sitúan del lado izquierdo, lo que significa que éste último recibió una evaluación más favorable. Por lo que la hipótesis $H_{1,B1}$ es rechazada.

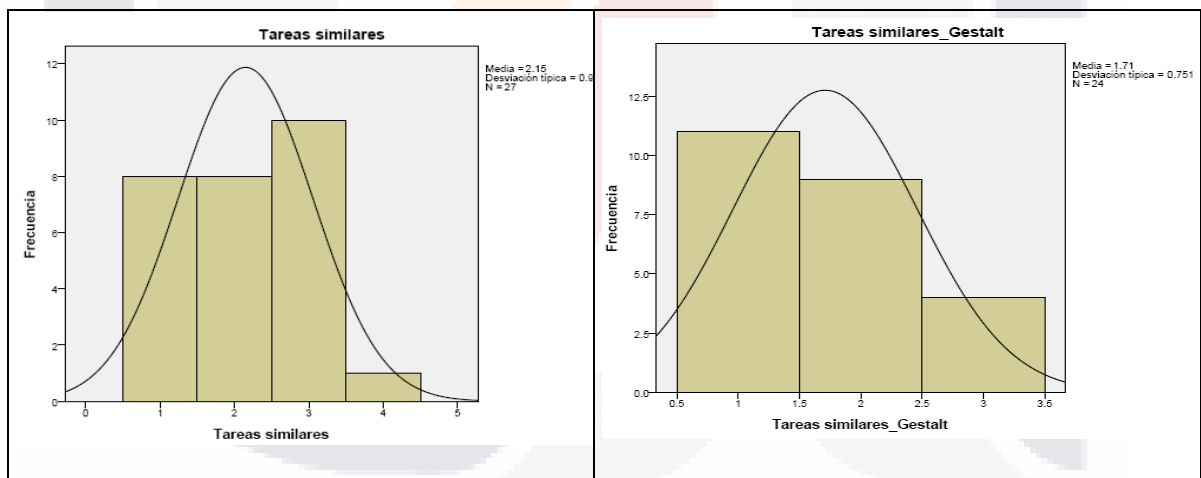


Figura 43. Histogramas. Sub-factor: Tareas similares

En el sub-factor **Control al usuario** se observó una diferencia **significativa** ($p < 0.001$) entre las dos formas de diseño de GUIs. Asimismo, se encontró que Gestalt (calidad promedio = 1.5) tiene una mejor apreciación por parte de los usuarios comparada sin Gestalt (calidad promedio = 2.85), es decir, se tiene una diferencia significativa de 1.352 (22.5%). Por lo anterior se puede deducir que Gestalt genera mejores resultados comparado con el diseño sin Gestalt de manera significativa.

En la Figura 44 se muestra de manera grafica la evaluación de ambos sistemas (a la izquierda sistema sin Gestalt y derecha sistema con Gestalt). En el primer histograma los datos se encuentran distribuidos entre los valores 1 y 7 a diferencia del segundo, donde los valores se encuentran a la derecha del grafico en 1 y 2 (95.8%), lo que representa una evaluación más favorable para el sistema diseñado con Gestalt. Por lo tanto, la hipótesis $H_{1.B2}$ es aceptada.

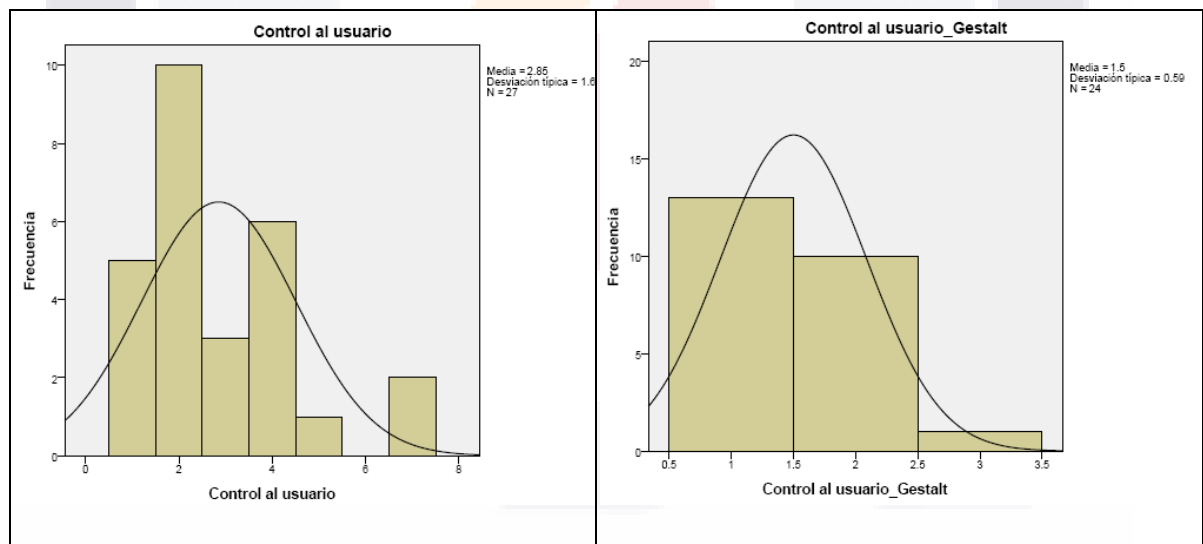


Figura 44. Histogramas. Sub-factor: Control al usuario

Respecto al sub-factor **Control de errores** se encontró una diferencia **significativa** ($p = 0.016$) entre ambas formas de diseño de GUIs. Además, se encontró que Gestalt (calidad promedio = 1.67) genera una mejor percepción por parte de los usuarios comparada con el diseño sin Gestalt (calidad promedio = 2.78), es decir, existe una diferencia de 1.111 (18.5%). Por lo anterior se puede

deducir que Gestalt genera mejores resultados comparado con el diseño sin Gestalt de manera significativa.

En la Figura 45 se puede observar del lado izquierdo el histograma de la evaluación del sistema sin Gestalt, en dicho grafico los datos se centran en el valor 2 (44.4%), también se observa una mayor dispersión entre los demás valores, que indica que las opiniones fueron más variadas, comparado con el histograma del lado derecho que pertenece al sistema diseñado con Gestalt, en el cual se muestra que la mayor concentración de datos se encuentra en 1 y 2 (83.3%), lo que demuestra que dicho sistema causó una mejor opinión a los usuarios. Por lo que la hipótesis $H_{1.B3}$ es Aceptada.

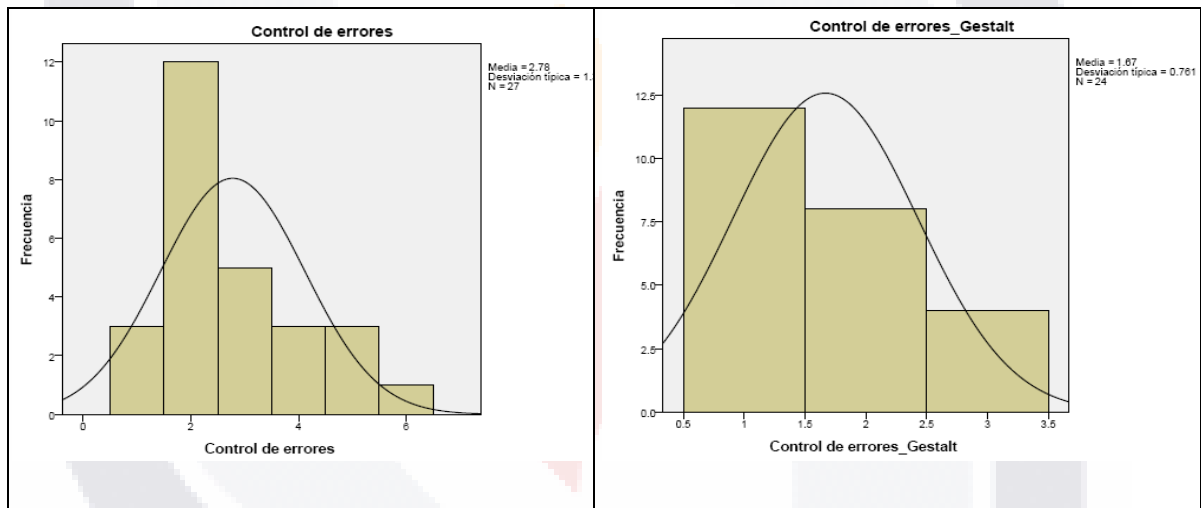


Figura 45. Histogramas. Sub-factor: Control de errores

En el sub-factor **Claridad de mensajes** se observó una diferencia **no significativa** ($p=0.866$) entre las dos formas de diseño de GUIs. Asimismo, se encontró que Gestalt (calidad promedio = 1.58) tiene una mejor apreciación por parte de los usuarios comparada con el diseño sin Gestalt (calidad promedio = 1.63), es decir, se tiene una diferencia significativa de 0.046 (0.8%). Por lo anterior se puede deducir que Gestalt genera mejores resultados comparado con el diseño sin Gestalt pero no de manera significativa.

En la Figura 46 se muestra de manera grafica la evaluación de ambos sistemas (a la izquierda sistema sin Gestalt y derecha sistema con Gestalt). En el primer

gráfico los datos de encuentran distribuidos entre 1 y 3 principalmente, de manera similar en el segundo se distribuyen entre 1 y 3. Existe una leve diferencia respecto al grafico de la derecha, lo cual favorece al sistema diseñado con Gestalt. Por lo que la hipótesis $H_{1,B4}$ es rechazada.

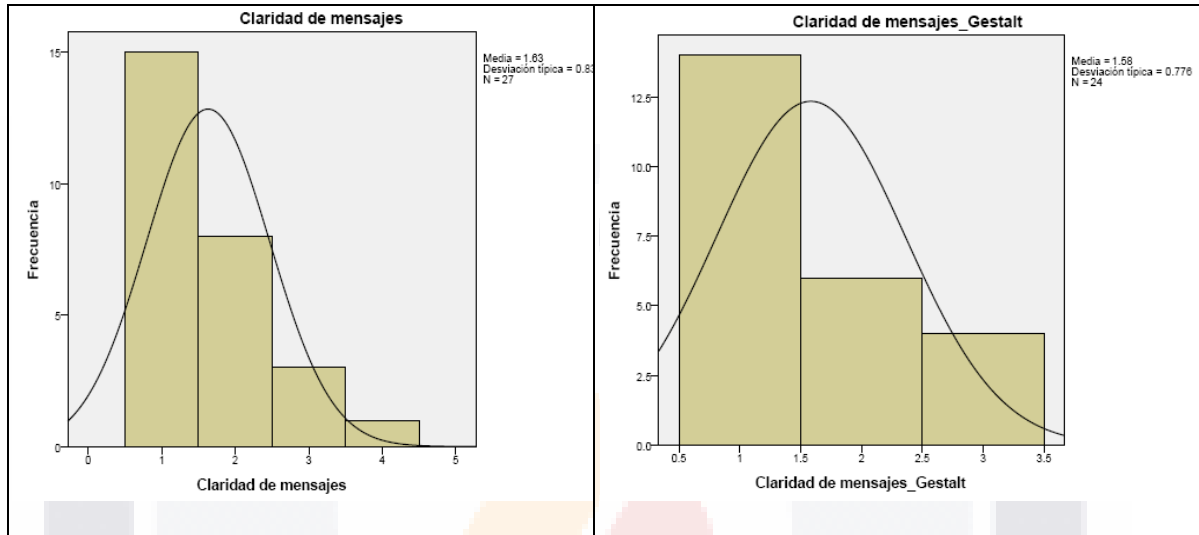


Figura 46. Histogramas. Sub-factor: Claridad de mensajes

En el sub-factor **Ayuda de mensajes** se observo una diferencia **no significativa** ($p=0.096$) entre ambas formas de diseño de GUIs. Además se encontró que el diseño sin Gestalt (calidad promedio = 1.44) produce una mejor percepción en los usuarios comparada con las que están diseñadas con Gestalt (calidad promedio =1.5), es decir, existe una diferencia de 0.056 (0.9%). Por lo tanto se puede deducir que el empleo de Gestalt no influye en la calidad de dicho sub-factor.

En la Figura 47 se muestran los histogramas correspondientes a las evaluaciones, del lado izquierdo se puede observar que la mayoría de los valores se concentran en 1 y 2 (96.3%), del lado derecho se observa la evaluación del sistema diseñado con Gestalt, donde la mayoría de los valores se centran en 1 (70.8%). De esta manera se muestra gráficamente la diferencia entre ambos diseños, la cual favorece al sistema diseñado con Gestalt, pero no resulta significativa. Por lo que la hipótesis $H_{1,B5}$ es rechazada.

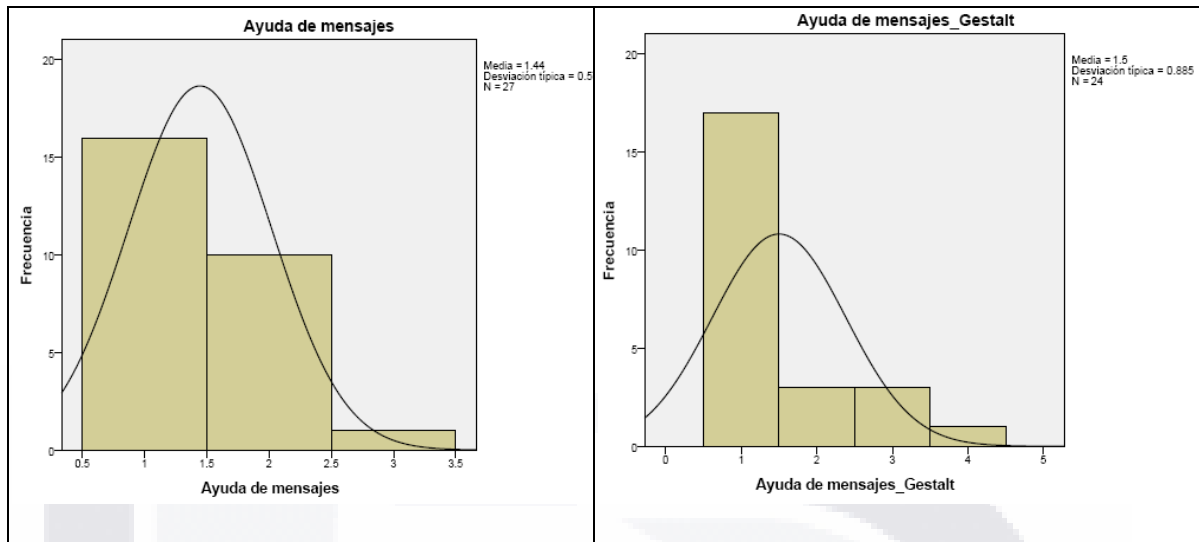


Figura 47. Histogramas. Sub-factor: Ayuda de mensajes

Respecto al sub-factor **Navegación** se observó una diferencia **no significativa** ($p=0.514$) entre las dos formas de diseño de GUIs. Además, se observó que Gestalt (calidad promedio = 1.63) produce una mejor percepción en los usuarios comparada con las que están diseñadas sin Gestalt (calidad promedio = 1.89), es decir, existe una diferencia de 0.264 (4.4%). Por lo tanto Gestalt produce mejores resultados que el diseño sin Gestalt pero no de manera significativa.

En la Figura 48 se muestran los histogramas correspondientes a las evaluaciones, del lado izquierdo se puede observar que la mayoría de los valores se concentran en 1 y 2 (77.8%), además se observa que los datos se encuentran más dispersos, a diferencia de los representados en el gráfico de la derecha (sistema diseñado con Gestalt), donde la mayoría de los datos se centran en 1 y 2 (91.7%). De esta manera se muestra gráficamente la diferencia entre ambos diseños, la cual favorece al sistema diseñado con Gestalt, pero no resulta significativa. Por lo que la hipótesis $H_{1.B6}$ es rechazada.

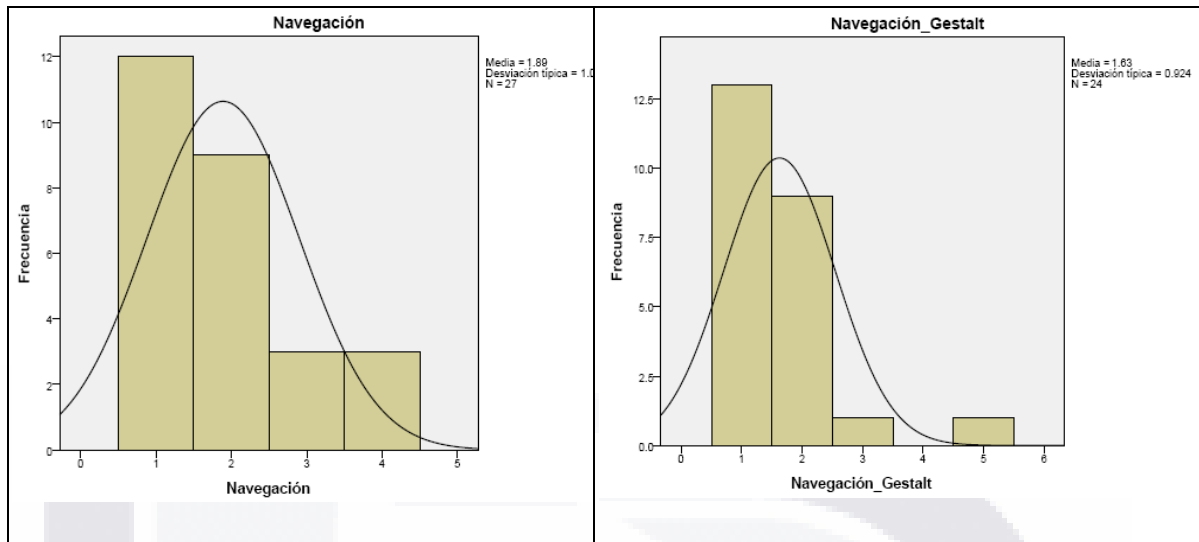


Figura 48. Histogramas. Sub-factor: Navegación

En el sub-factor **Movimientos de navegación** se observó una diferencia **no significativa** ($p=0.564$) entre las dos formas de diseño de GUIs. Asimismo, se encontró que Gestalt (calidad promedio = 1.63) tiene una mejor apreciación por parte de los usuarios comparada con el diseño sin Gestalt (calidad promedio = 1.96), es decir, se tiene una diferencia significativa de 0.338 (5.6%). Por lo anterior se puede deducir que Gestalt genera mejores resultados comparado con el diseño sin Gestalt pero no de manera significativa.

La Figura 49 muestra gráficamente la evaluación de ambos sistemas. En el histograma de la izquierda (sistema sin Gestalt) se puede observar que los datos se encuentran más dispersos en comparación al gráfico de la derecha (sistema con Gestalt) donde se aprecia que los datos se encuentran orientados al lado izquierdo, lo que representa una mejor opinión del sistema con Gestalt. Por lo que la hipótesis $H_{1,B7}$ es rechazada.

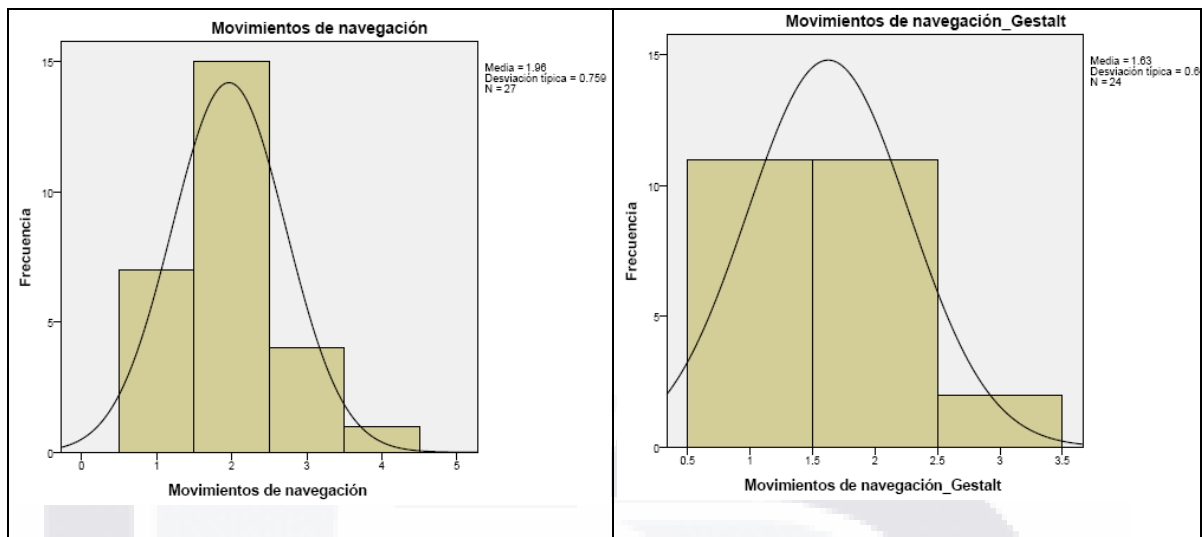


Figura 49. Histogramas. Sub-factor: Movimientos de navegación

En resumen respecto al factor Facilidad de operación, la Figura 50 muestra la evaluación de cada uno de los subfactores, en la cual solo dos de los subfactores Control al usuario y control de errores tuvieron una diferencia significativa en la calidad, favorable para el sistema diseñado sin Gestalt. Los subfactores restantes presentaron una diferencia favorable al sistema con Gestalt, pero éstas no resultaron ser significativas.

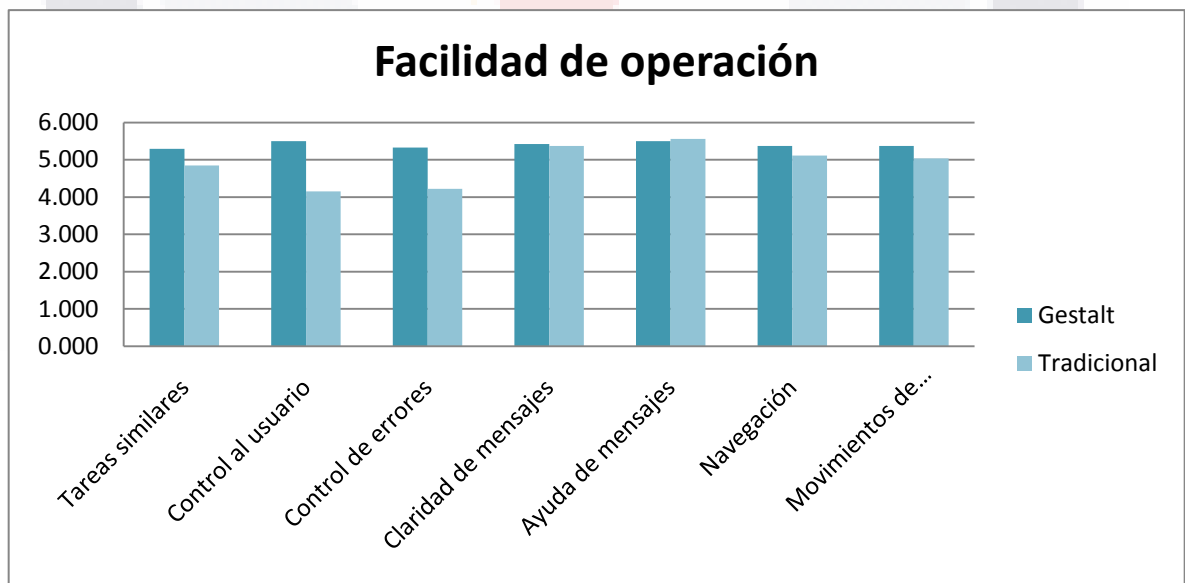


Figura 50. Gráfico de barras para comparar las medias de los sub-factores de Facilidad de operación.

4.1.3 Factor Atractividad

En el sub-factor **Organización de elementos** se observó una diferencia **significativa** ($p=0.026$) entre las dos formas de diseño de GUIs. Asimismo, se encontró que Gestalt (calidad promedio = 1.75) tiene una mejor apreciación por parte de los usuarios comparada con el diseño sin Gestalt (calidad promedio = 2.44), es decir, se tiene una diferencia significativa de 0.694 (11.6%). Por lo anterior se puede deducir que Gestalt genera mejores resultados comparado con el diseño sin Gestalt de manera significativa.

En la Figura 51 se puede apreciar una diferencia significativa entre los histogramas que muestran los datos del sistema sin Gestalt (lado izquierdo) y el sistema con Gestalt (lado derecho). En el primero se aprecia que las opiniones se encuentran más dispersas, a diferencia del segundo donde la mayoría de los datos se encuentra del lado izquierdo en los valores 1 y 2 (91.7 %). Esto significa que el sistema con Gestalt recibió una evaluación significativamente más favorable. Por lo que la hipótesis $H_{1,c1}$ es aceptada.

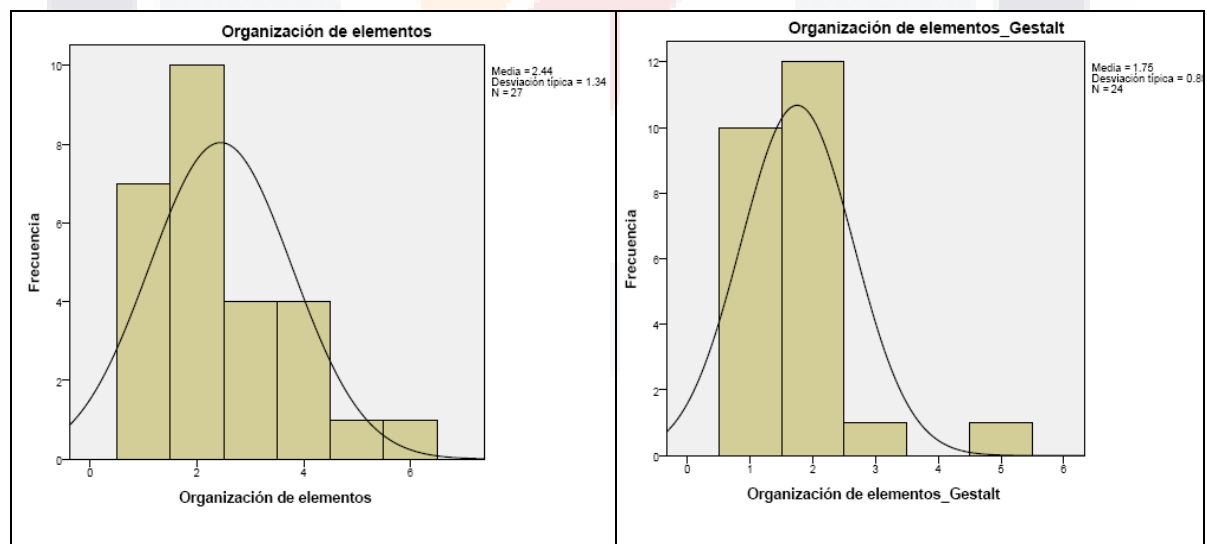


Figura 51. Histogramas. Sub-factor: Organización de elementos

Respecto al sub-factor **Combinación de colores** se encontró una diferencia **significativa** ($p \leq 0.001$) entre ambas formas de diseño de GUIs. Además, se encontró que Gestalt (calidad promedio = 1.58) genera una mejor percepción por parte de los usuarios comparada con el diseño sin Gestalt (calidad promedio = 3.63), es decir, existe una diferencia de 2.046 (34.1%). Por lo anterior se puede deducir que Gestalt genera mejores resultados comparado con el diseño sin Gestalt de manera significativa.

En la Figura 52 se puede observar del lado izquierdo el histograma de los valores del sistema sin Gestalt, donde se observa una mayor dispersión entre los valores, que indica que las opiniones fueron más variadas, comparado con el histograma del lado derecho que pertenece al sistema diseñado con Gestalt, en el cual se muestra que la mayor concentración de datos se encuentra ubicada en el lado izquierdo en el valor 1 (66.7%), lo que demuestra que dicho sistema causó una mejor opinión a los usuarios. Por lo que la hipótesis $H_{1,c2}$ es Aceptada.

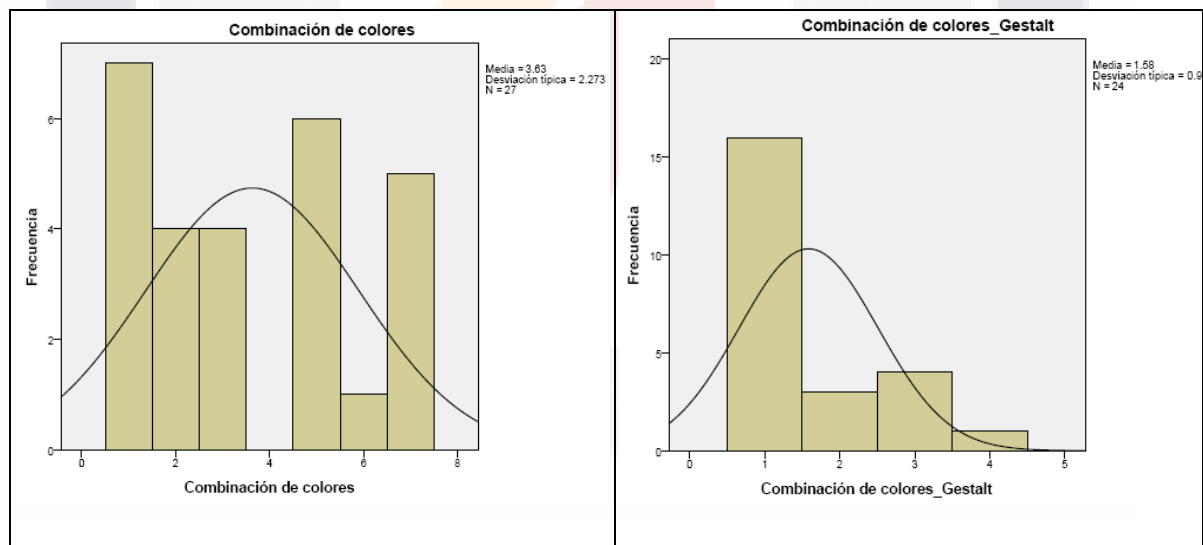


Figura 52. Histogramas. Sub-factor: Combinación de colores

En el sub-factor **Color de fondo** se observó una diferencia **significativa** ($p=0.002$) entre las dos formas de diseño de GUIs. Asimismo, se encontró que Gestalt (calidad promedio = 1.75) tiene una mejor apreciación por parte de los usuarios comparada con el diseño sin Gestalt (calidad promedio = 3.52), es decir, se tiene

una diferencia significativa de 1.769 (29.5%). Por lo anterior se puede deducir que Gestalt genera mejores resultados comparado con el diseño sin Gestalt de manera significativa.

En la Figura 53 se muestran los histogramas correspondientes a la evaluación de ambos sistemas, del lado izquierdo (sistema sin Gestalt) se puede observar que la mayoría de los valores se encuentran muy dispersos, del lado derecho se observan la evaluación del sistema diseñado con Gestalt, donde los valores se centran en 1 y 2 (87.5%). La diferencia entre ambas evaluaciones favorece significativamente al sistema diseñado con Gestalt. Por lo que la hipótesis $H_{1,c3}$ es aceptada.

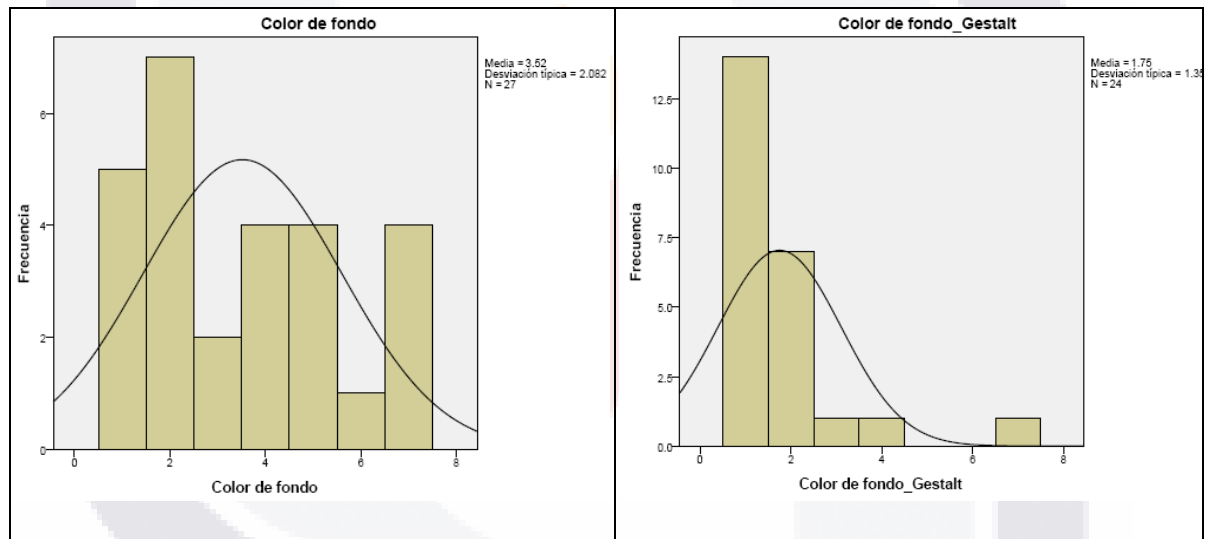


Figura 53. Histogramas. Sub-factor: Color de fondo

Respecto al sub-factor **Atractividad de símbolos** se encontró una diferencia **significativa** ($p=0.001$) entre ambas formas de diseño de GUIs. Además, se encontró que Gestalt (calidad promedio = 1.46) genera una mejor percepción por parte de los usuarios comparada con el diseño sin Gestalt (calidad promedio = 2.81), es decir, existe una diferencia de 1.356 (22.6%). Por lo anterior se puede deducir que Gestalt genera mejores resultados comparado con el diseño sin Gestalt de manera significativa.

En la Figura 54 se observan los histogramas correspondientes a los datos de las evaluaciones de ambos sistemas (izquierda sistema sin Gestalt, derecha sistema con Gestalt), en el primero se puede apreciar que los datos se encuentran distribuidos entre los diferentes valores, mientras que del lado derecho la concentración de datos se ubican en el lado izquierdo 95.8%. Esto significa que el sistema con Gestalt fue evaluado notablemente de manera más satisfactoria en comparación al sistema sin Gestalt, en el cual las opiniones fueron más variadas. Por lo que la hipótesis $H_{1,c4}$ es aceptada.

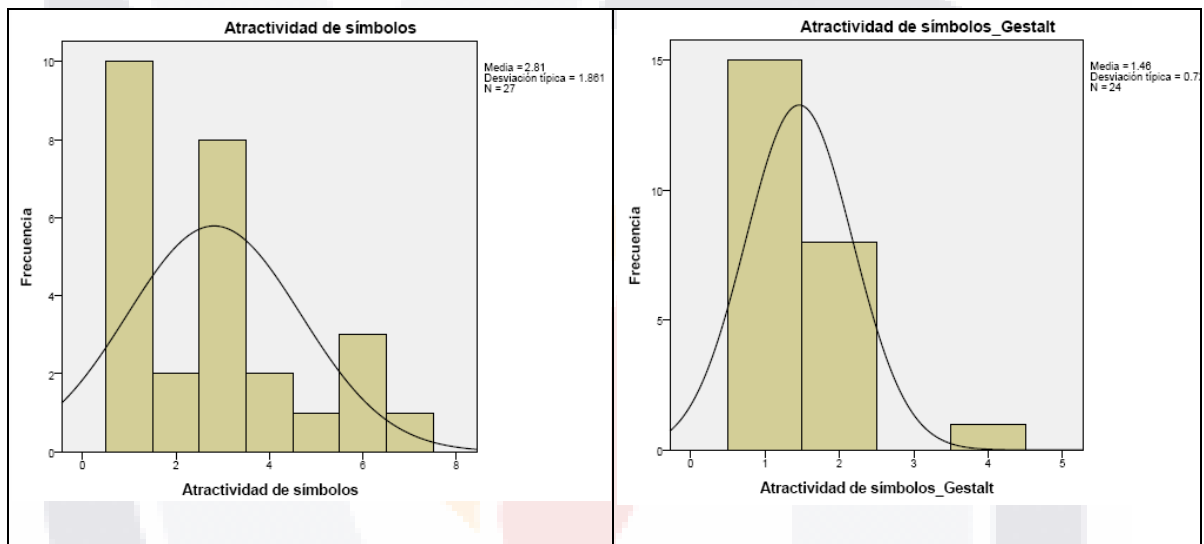


Figura 54. Histogramas. Sub-factor: Atractividad de símbolos

En resumen respecto al factor Atractividad, la Figura 55 muestra la evaluación de cada uno de los subfactores, en la cual los cuatro subfactores que componen el factor tuvieron una diferencia significativa en la calidad, favorable para el sistema diseñado con Gestalt.

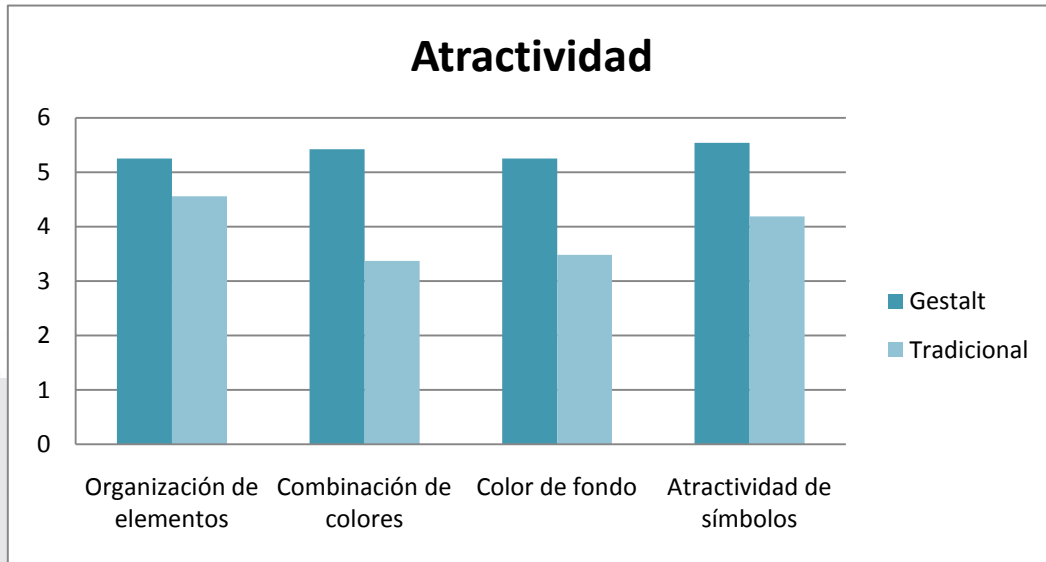


Figura 55. Gráfico de barras para comparar las medias de los sub-factores de Atractividad

4.1.4 Factor Facilidad de aprendizaje

Respecto al sub-factor **Recordar acciones anteriores** se observó una diferencia **no significativa** ($p=0.266$) entre las dos formas de diseño de GUIs. Además, se observó que Gestalt (calidad promedio = 1.54) produce una mejor percepción en los usuarios comparada con las que están diseñadas sin Gestalt (calidad promedio = 2.33), es decir, existe una diferencia de 0.792 (13.2%). Por lo tanto Gestalt produce mejores resultados que el diseño sin Gestalt pero no de manera significativa.

En la Figura 56 se puede observar del lado izquierdo el histograma de los valores del sistema sin Gestalt, los cuales se concentran en el valor 2 (48.1%), también se observa una mayor dispersión entre los demás valores, que indica que las opiniones fueron más variadas, comparado con el histograma del lado derecho que pertenece al sistema diseñado con Gestalt, en el cual se muestra que la mayor concentración de datos se encuentra en 1 y 2 (91.7%), lo que demuestra que dicho sistema causó una mejor opinión a los usuarios. Por lo tanto, la hipótesis $H_{2,A}$ es rechazada.

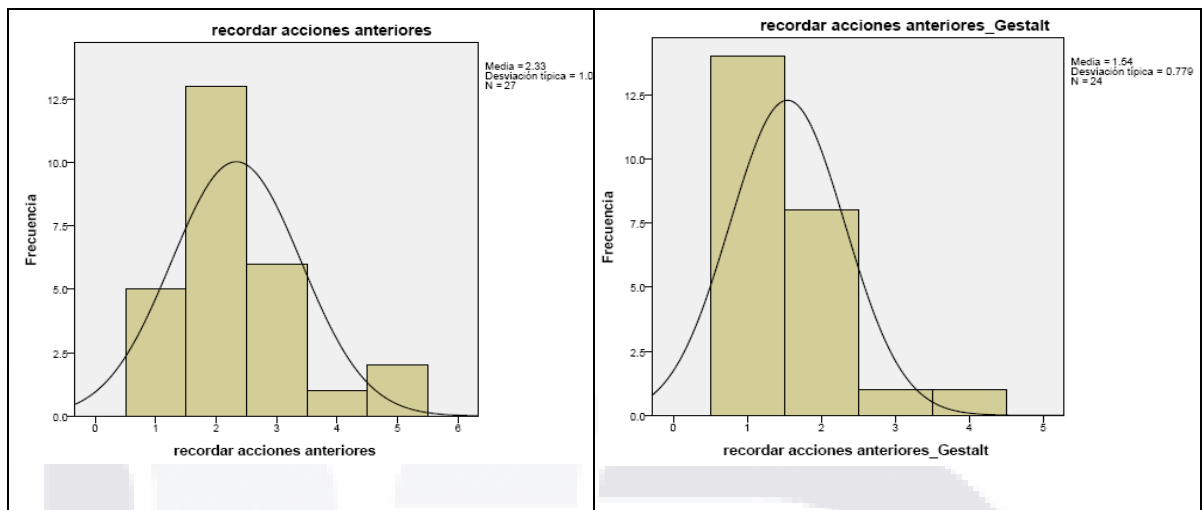


Figura 56. Histogramas. Sub-factor: Recordar acciones anteriores

En el sub-factor **Tareas fáciles de comprender** se encontró una diferencia **no significativa** ($p=0.990$) entre ambas formas de diseño de GUIs. Además se observó que Gestalt (calidad promedio = 1.71) produce una mejor percepción en los usuarios comparada con las que están diseñadas sin Gestalt (calidad promedio = 2.04), es decir, existe una diferencia de 0.329 (5.5%). Por lo tanto Gestalt produce mejores resultados que el diseño sin Gestalt pero no de manera significativa.

En la Figura 57 se muestra de manera gráfica la evaluación de ambos sistemas (a la izquierda sistema sin Gestalt y derecha sistema con Gestalt). En el primero los datos se encuentran distribuidos de manera normal entre 1 y 4, a diferencia del gráfico de la derecha, donde los valores se concentran del lado izquierdo entre 1 y 3. La evaluación resulta levemente más favorable para el sistema construido con Gestalt, sin que esta diferencia resulte significativa. Por lo tanto, la hipótesis $H_{2,B}$ es rechazada.

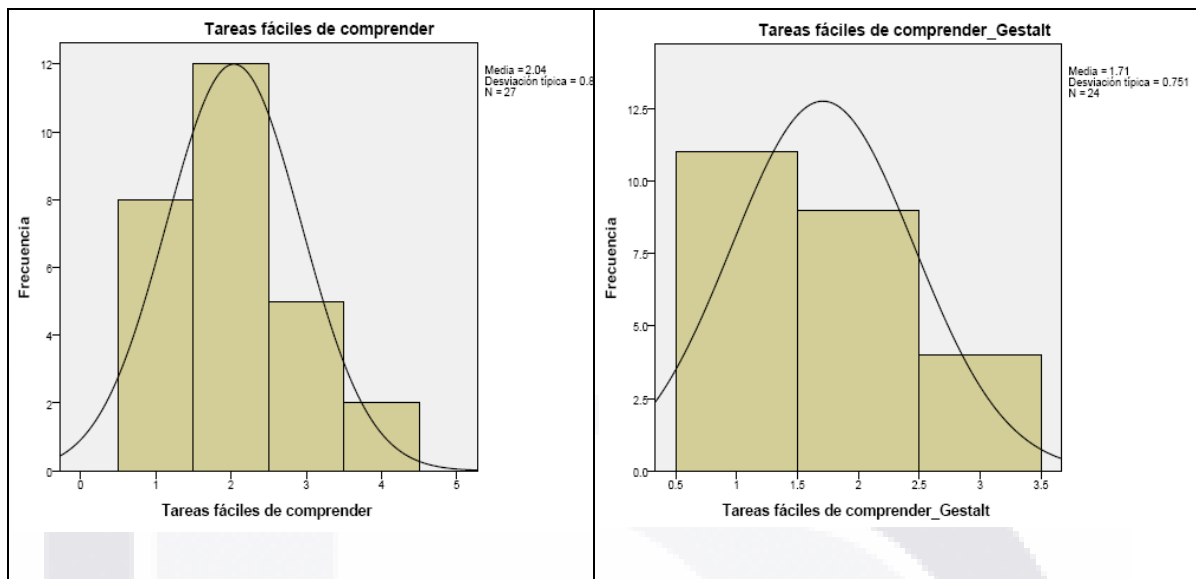


Figura 57. Histogramas. Sub-factor: Tareas fáciles de comprender

En el sub-factor **Tareas fáciles de recordar** se observó una diferencia **significativa** ($p=0.044$) entre ambas formas de diseño de GUIs. Además se encontró que el diseño sin Gestalt (calidad promedio = 1.67) produce una mejor percepción en los usuarios comparada con las que están diseñadas con Gestalt (calidad promedio = 1.79), es decir, existe una diferencia de 0.125 (2.1%). Por lo tanto, se puede deducir que el empleo de Gestalt influye en la calidad de dicho sub-factor más no en el sentido planteado para la presente investigación.

En la Figura 58 se puede apreciar una diferencia significativa entre los histogramas que muestran los datos del sistema sin Gestalt (lado izquierdo) y el sistema con Gestalt (lado derecho). En el segundo se aprecia que las opiniones se encuentran más dispersas, a diferencia del primero donde el 92.6% de las opiniones se centró en los valores 1 y 2. La hipótesis $H_{2,c}$ es aceptada desde el punto de vista estadístico, pero esto contradice con lo planteado en la presente investigación ya que la calidad es mejor en sistemas con interfaces desarrolladas sin Gestalt que desarrolladas con esta teoría. En consecuencia, no tiene significancia práctica.

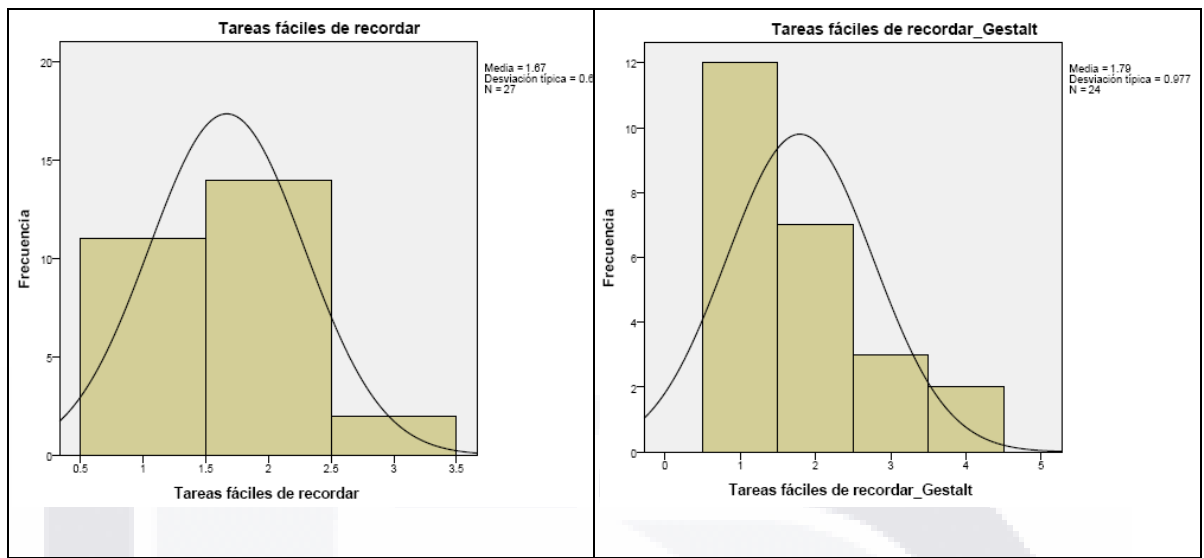


Figura 58. Histogramas. Sub-factor: Tareas fáciles de recordar

En el sub-factor **Información ordenada de manera jerárquica** se encontró una diferencia **no significativa** ($p=0.138$) entre las dos formas de diseño de GUIs. Además, se observó que Gestalt (calidad promedio = 1.58) produce una mejor opinión de los usuarios comparada con las que están diseñadas sin Gestalt (calidad promedio = 2.15), es decir, existe una diferencia de 0.565 (9.4%). Por lo tanto se puede deducir que el empleo de Gestalt no influye significativamente en la calidad de dicho sub-factor.

En la Figura 59 se muestra de manera gráfica la evaluación de ambos sistemas (a la izquierda sistema sin Gestalt y derecha sistema con Gestalt). En el primero los datos se encuentran distribuidos entre 1 y 5, a diferencia del gráfico de la derecha, donde los valores se concentran del lado izquierdo entre 1 y 2 (91.7%). La evaluación resulta levemente más favorable para el sistema construido con Gestalt, sin que esta diferencia resulte significativa. Por lo que la hipótesis $H_{2,D}$ es rechazada.

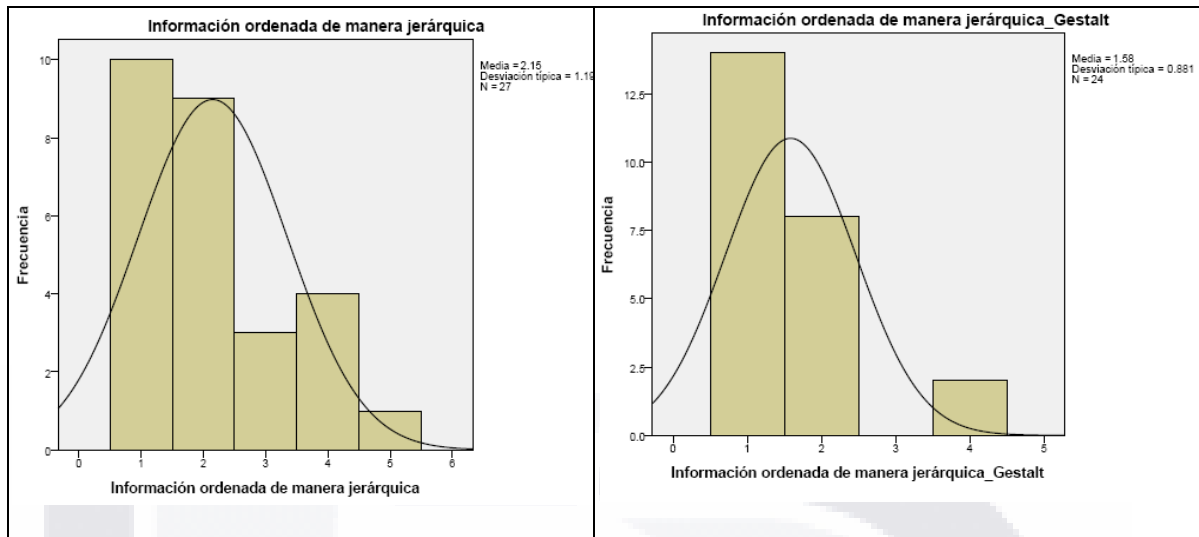


Figura 59. Histogramas. Sub-factor: Información ordenada de manera jerárquica

En el sub-factor **Cantidad de elementos** se encontró una diferencia **no significativa** ($p=0.173$) entre ambas formas de diseño de GUIs. Asimismo, se encontró que Gestalt (calidad promedio = 1.92) genera una mejor percepción por parte de los usuarios comparada sin Gestalt (calidad promedio = 2.63), es decir, existe una diferencia de 0.713 (11.9%). Por lo tanto Gestalt produce mejores resultados que el diseño sin Gestalt pero no de manera significativa.

En la Figura 60 se observan los resultados de ambas evaluaciones, en el gráfico de la izquierda (sistema sin Gestalt) se observa que los valores se encuentran más dispersos a diferencia del gráfico de la derecha (sistema con Gestalt) donde la mayoría de los datos se concentran del lado izquierdo entre los valores 1 y 3 (95.8%). La opinión de los usuarios favorece levemente al sistema diseñado con Gestalt. Por lo tanto, la hipótesis $H_{2,E}$ es rechazada.

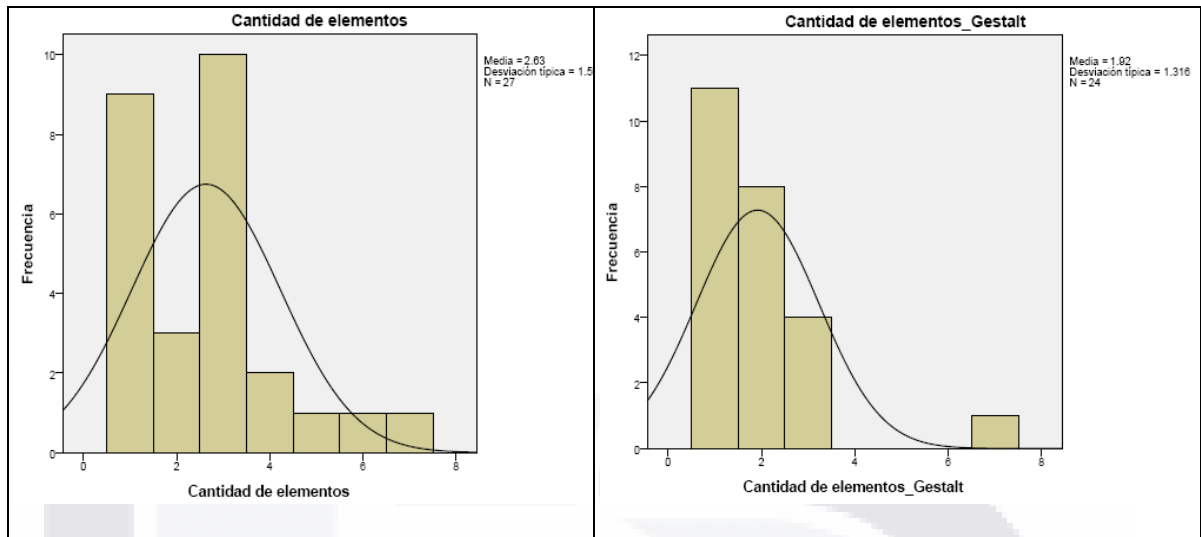


Figura 60. Histogramas. Sub-factor: Cantidad de elementos

En resumen respecto al factor Facilidad de aprendizaje, la Figura 61 muestra la evaluación de cada uno de los subfactores, en la cual solo uno de los subfactores Tareas fáciles de recordar resultó tener una diferencia significativa en la calidad, favorable para el sistema diseñado sin Gestalt. Las subfactores restantes presentaron una diferencia favorable al sistema con Gestalt, pero éstas no resultaron ser significativas.

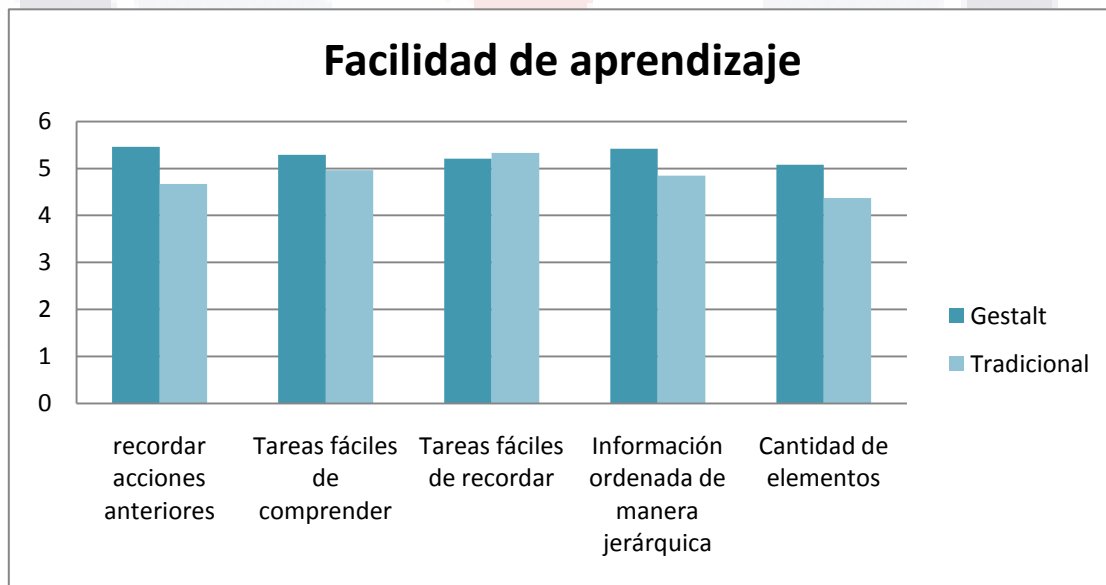


Figura 61. Gráfico de barras para comparar las medias de los sub-factores de Facilidad de aprendizaje

4.1.5 Factor Aceptación del usuario

En el sub-factor **Beneficio percibido** se observó una diferencia **significativa** ($p=0.014$) entre las dos formas de diseño de GUIs. Asimismo, se encontró que Gestalt (calidad promedio = 1.33) tiene una mejor apreciación por parte de los usuarios comparada con el diseño sin Gestalt (calidad promedio = 2.44), es decir, se tiene una diferencia significativa de 1.111 (18.5%). Por lo anterior se puede deducir que Gestalt genera mejores resultados comparado con el diseño sin Gestalt de manera significativa.

En la Figura 62 se puede apreciar una diferencia significativa entre los histogramas que muestran los datos del sistema sin Gestalt (lado izquierdo) y el sistema con Gestalt (lado derecho). En el primero se aprecia que las opiniones se encuentran más dispersas, a diferencia del segundo donde el 95.8 % de las opiniones se centró en los valores 1 y 2. Por lo que la hipótesis $H_{3,A}$ es aceptada.

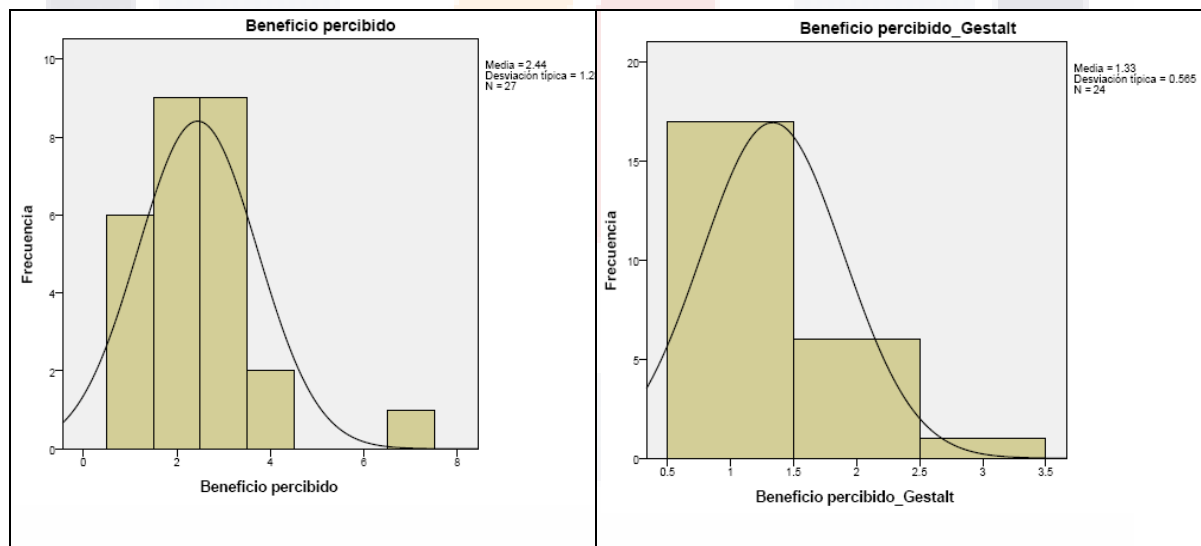


Figura 62. Histogramas. Sub-factor: **Beneficio percibido**

Respecto al sub-factor **Confianza** se encontró una diferencia **significativa** ($p=0.041$) entre ambas formas de diseño de GUIs. Además, se encontró que Gestalt (calidad promedio = 1.63) genera una mejor percepción por parte de los usuarios comparada con el diseño sin Gestalt (calidad promedio = 3.04), es decir, existe una diferencia de 1.412 (23.5%). Por lo anterior se puede deducir que

Gestalt genera mejores resultados comparado con el diseño sin Gestalt de manera significativa.

En la Figura 63 se puede observar del lado izquierdo el histograma de los valores del sistema sin Gestalt, en el cual se observa una mayor dispersión entre los demás valores, que indica que las opiniones fueron más variadas, comparado con el histograma del lado derecho que pertenece al sistema diseñado con Gestalt, en el cual se muestra que la mayor concentración de datos se encuentra en 1 y 2 (87.5%), lo que demuestra que dicho sistema causó una mejor opinión a los usuarios. Por lo que la hipótesis $H_{3,B}$ es aceptada.

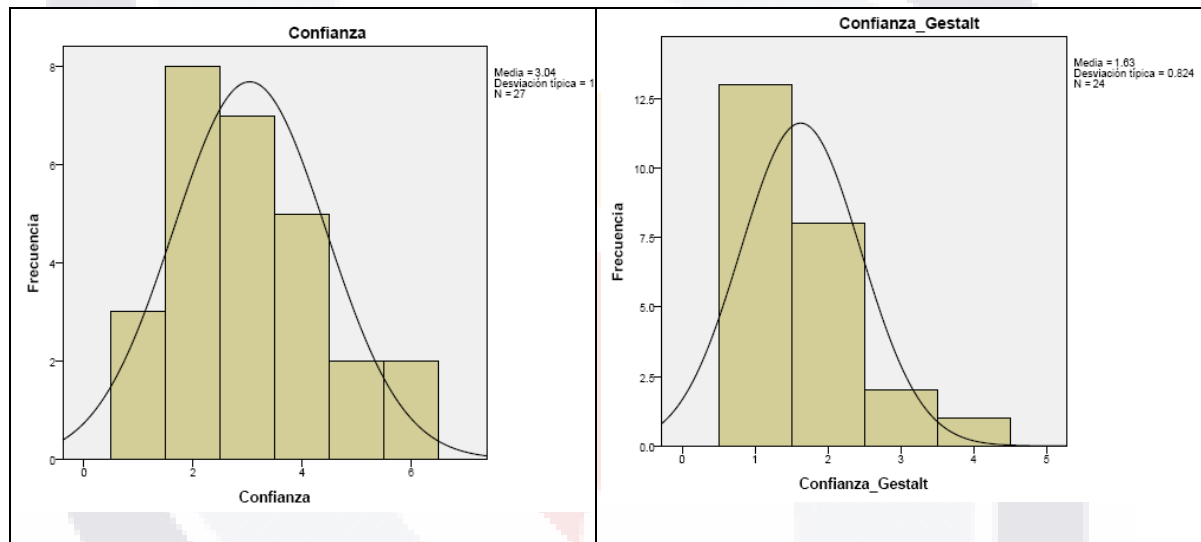


Figura 63. Histogramas. Sub-factor: Confianza

En el sub-factor **Beneficio percibido** se observó una diferencia **significativa** ($p=0.008$) entre las dos formas de diseño de GUIs. Asimismo, se encontró que Gestalt (calidad promedio = 1.63) tiene una mejor apreciación por parte de los usuarios comparada con el diseño sin Gestalt (calidad promedio = 2.85), es decir, se tiene una diferencia significativa de 1.227 (20.5%). Por lo anterior se puede deducir que Gestalt genera mejores resultados comparado con el diseño sin Gestalt de manera significativa. En la Figura 64 se puede apreciar una diferencia significativa entre los histogramas que muestran los datos del sistema sin Gestalt (lado izquierdo) y el sistema con Gestalt (lado derecho). En el primero se aprecia

que las opiniones se encuentran más dispersas, a diferencia del segundo donde el 991.7% de las opiniones se centró en los valores 1 y 2. Por lo que la hipótesis $H_{3,c}$ es aceptada.

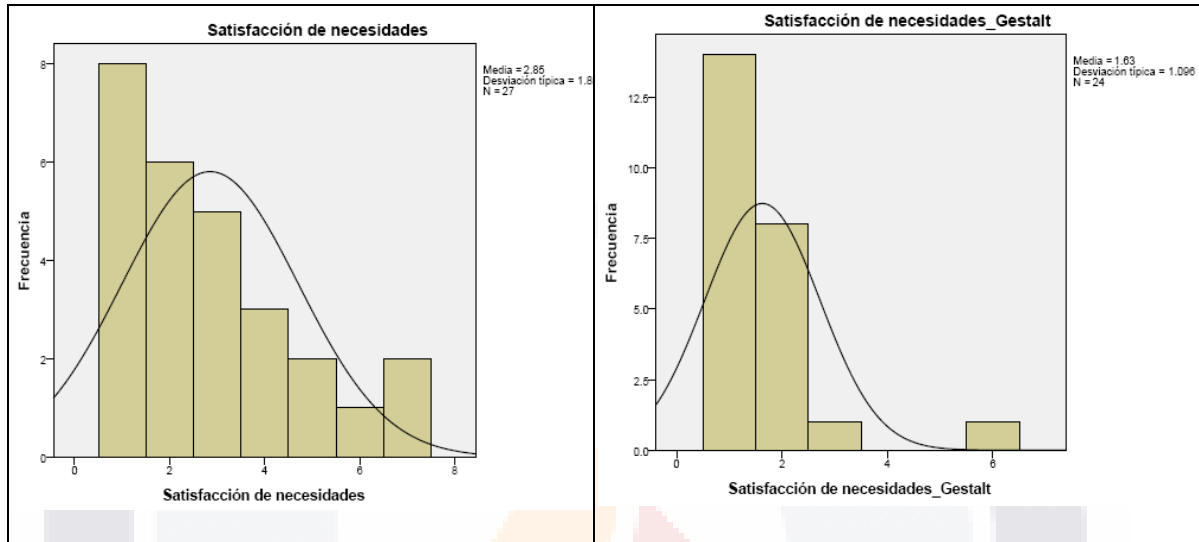


Figura 64. Histogramas. Sub-factor: Satisfacción de necesidades

En resumen respecto al factor Nivel de aceptación del usuario, la Figura 65 muestra la evaluación de cada uno de los subfactores, en la cual los tres subfactores que componen el factor tuvieron una diferencia significativa en la calidad, favorable para el sistema diseñado con Gestalt.

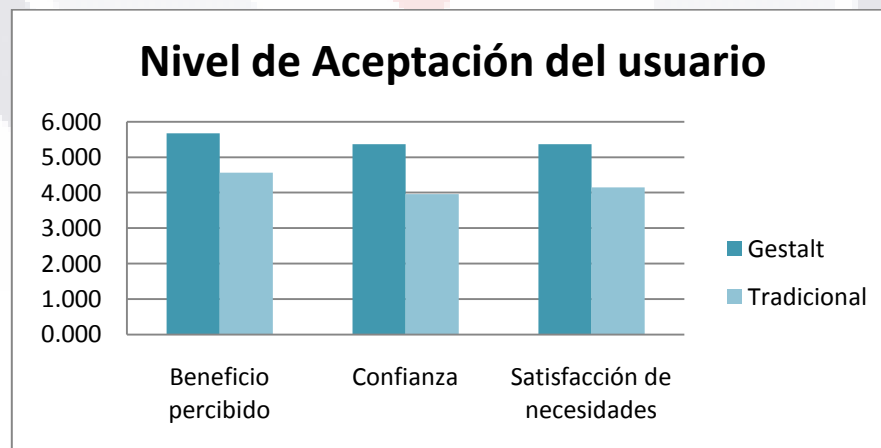


Figura 65. Gráfico de barras para comparar las medias de los sub-factores de Nivel de aceptación del usuario

A continuación en la Tabla 16 se muestran los resultados para cada hipótesis planteada.

Hipótesis	Resultado	Prob.
H _{1.A1} : Las GUIs construidas bajo los principios de la teoría Gestalt tienen un lenguaje más familiar comparadas con las GUIs que se construyen sin estos principios.	Rechazada	.748
H _{1.A2} : Las GUIs construidas bajo los principios de la teoría Gestalt tienen un tamaño de texto más adecuado comparadas con las GUIs que se construyen sin estos principios.	Aceptada	.004
H _{1.A3} : Las GUIs construidas bajo los principios de la teoría Gestalt tienen un tipo de texto más adecuado comparadas con las GUIs que se construyen sin estos principios.	Rechazada	.149
H _{1.A4} : Las GUIs construidas bajo los principios de la teoría Gestalt tienen un texto más fácil de leer debido al contraste entre el color del texto y el color de fondo comparadas con las GUIs que se construyen sin estos principios.	Aceptada	.002
H _{1.A5} : Las GUIs construidas bajo los principios de la teoría Gestalt tienen un texto más fácil de comprender debido a que la redacción es más adecuada comparadas con las GUIs que se construyen sin estos principios.	Rechazada	.312
H _{1.A6} : Las GUIs construidas bajo los principios de la teoría Gestalt tienen símbolos más fáciles de comprender comparadas con las GUIs que se construyen sin estos principios.	Rechazada	.683
H _{1.B1} : Las GUIs construidas bajo los principios de la teoría Gestalt tienen operaciones más consistentes comparadas con las GUIs que se construyen sin estos principios.	Rechazada	.309
H _{1.B2} : Las GUIs construidas bajo los principios de la teoría Gestalt proporcionan al usuario más control sobre las tareas que realiza comparadas con las GUIs que se construyen sin estos principios.	Aceptada	.001
H _{1.B3} : Las GUIs construidas bajo los principios de la teoría Gestalt manejan más fácilmente los errores que puede cometer el usuario comparadas con las GUI que se construyen sin estos principios.	Aceptada	.016
H _{1.B4} : Las GUIs construidas bajo los principios de la teoría Gestalt tienen mensajes más claros comparadas con las GUIs que se construyen sin estos principios.	Rechazada	.866
H _{1.B5} : Las GUIs construidas bajo los principios de la teoría Gestalt tienen mensajes más útiles comparadas con las GUIs que se construyen sin estos principios.	Rechazada	.096
H _{1.B6} : Las GUIs construidas bajo los principios de la teoría Gestalt resultan más fáciles de navegar comparadas con las GUIs que se construyen sin estos principios.	Rechazada	.514
H _{1.B7} : Las GUIs construidas bajo los principios de la teoría Gestalt tienen tareas más fáciles de acceder debido a que la cantidad de movimientos es más adecuada, comparadas con las GUIs que se construyen sin estos principios.	Rechazada	.564
H _{1.C1} : Las GUIs construidas bajo los principios de la teoría Gestalt están organizadas de manera más simple comparadas con las GUIs que se construyen sin estos principios.	Aceptada	.026
H _{1.C2} : Las GUIs construidas bajo los principios de la teoría Gestalt resultan más agradables debido a la combinación de colores utilizados, comparadas con las GUIs que se construyen sin estos principios.	Aceptada	.001
H _{1.C3} : Las GUIs construidas bajo los principios de la teoría Gestalt resultan más agradables debido al color de fondo utilizado, comparadas con las GUIs que se construyen sin estos principios.	Aceptada	.002
H _{1.C4} : Las GUIs construidas bajo los principios de la teoría Gestalt tienen símbolos e iconos más agradables , comparadas con las GUIs que se construyen sin estos principios.	Aceptada	.001
H _{2.A} : Las GUIs construidas bajo los principios de la teoría Gestalt ayudan a recordar más fácilmente acciones anteriores comparadas con las GUIs que se construyen sin estos principios.	Rechazada	.266
H _{2.B} : Las GUIs construidas bajo los principios de la teoría Gestalt ayudan a comprender como realizar las tareas más fácilmente comparadas con las GUIs que se construyen sin estos principios.	Rechazada	.990
H _{2.C} : Las GUIs construidas bajo los principios de la teoría Gestalt ayudan a recordar más fácilmente como acceder a las tareas comparadas con las GUI que se construyen sin estos principios.	Aceptada	.044
H _{2.D} : En las GUIs construidas bajo los principios de la teoría Gestalt la información se encuentra más ordenada de manera jerárquica , comparadas con las GUIs que se construyen sin estos principios.	Rechazada	.138
H _{2.E} : En las GUIs construidas bajo los principios de la teoría Gestalt la cantidad de elementos utilizados resulta más adecuada, comparadas con las GUIs que se construyen sin estos principios.	Rechazada	.173
H _{3.A} : Las GUIs construidas bajo los principios de la teoría Gestalt proporcionan más beneficios al usuario comparadas con las GUIs que se construyen sin estos principios	Aceptada	.014
H _{3.B} : Las GUIs construidas bajo los principios de la teoría Gestalt inspiran más confianza al usuario comparadas con las GUIs que se construyen sin estos principios	Aceptada	.041
H _{3.C} : Las GUIs construidas bajo los principios de la teoría Gestalt satisfacen mejor las necesidades de los usuarios comparadas con las GUIs que se construyen sin estos principios	Aceptada	.008

Tabla 16. Resultado de las hipótesis planteadas

TESIS TESIS TESIS TESIS TESIS

Capítulo 5

Conclusiones

5.1 Conclusiones generales

En el presente trabajo de investigación se resalta la importancia de la evaluación de la calidad del software desde el punto de vista del producto y no del proceso como generalmente se realiza. Dicha evaluación efectuada desde la perspectiva del usuario final no del desarrollador, lo cual busca asegurar plenamente que se satisfacen sus necesidades así como incrementar la aceptación del nuevo software. De la misma manera, se hace notar que la evaluación de la calidad del software a partir de la interface de usuario podría representar la diferencia entre el éxito o fracaso de una aplicación. La evaluación de una GUI por parte del usuario final busca obtener retroalimentación, de manera que se determine si a éste le gusta y se siente cómodo tanto con la estética así como con la interacción con el software.

Con el fin de poder obtener la opinión del usuario, para la presente investigación se diseñó un instrumento de evaluación, el cual busca reducir el nivel de subjetividad del concepto de calidad en las GUIs. Dicho cuestionario fue validado estadísticamente, obteniéndose alto grado de confiabilidad, por lo que se tiene la certeza de que en realidad mide lo que se intenta medir a través de él. Dicho instrumento fue fundamental en la evaluación del impacto que tiene la teoría Gestalt en la calidad del software, particularmente en las GUIs. La evaluación se desarrolló en torno a 25 sub-factores identificados en la literatura (ISO/IEC 9126;

Mandel, 1997; McCall A., Richards, & Walters, 1977; Nielsen, 2005; Tognazzini, 2001), de los cuáles 11 mostraron un impacto significativo en el presente estudio. En consecuencia, se puede afirmar que desarrollar GUIs tomando en consideración los principios Gestalt realmente incremental la calidad del software final de acuerdo a la percepción del usuario final. A continuación se describe el impacto identificado de cada uno de los 25 sub-factores.

El sub-factor **lenguaje familiar** resultó no ser estadísticamente significativo, esto pudo deberse a que ninguno principio se refiere de forma concreta y específica a la forma de redacción o utilización del lenguaje en la construcción de GUIs, en consecuencia, la redacción fue prácticamente la misma en ambos diseños. Aunque la calidad en las GUIs con Gestalt resultó ser mayor en éste aspecto, no fue lo suficiente para ser considerada significativa. No obstante, es importante que para construir GUIs con calidad se recomienda que el lenguaje utilizado esté basado en el conocimiento del usuario, es decir, conocer los aspectos de lenguaje de las personas a las que está dirigido el sistema. En general, el lenguaje debe ser claro, sencillo, y consistente, utilizar términos con los que el usuario esté familiarizado, de manera que sea comprensible para éste, además de cuidar aspectos tales como la sintáctica, por ejemplo: la ortografía, la puntuación, y la gramática; y la semántica, por ejemplo: la exactitud de la información presentada, la consistencia y la ambigüedad. Finalmente, es importante recordar que la redacción también debe incluir la información mínima necesaria dentro del ámbito de la aplicación del sistema para el cual las GUIs fueron desarrolladas.

En relación al sub-factor **tamaño del texto** se encontró una diferencia significativa. Dicha diferencia pudo deberse al empleo del principio de la teoría Gestalt Punto Focal (REF), el cual se refiere al elemento que captura la atención entre un grupo de elementos, ya que en el diseño de las GUIs con Gestalt el tamaño del texto de los encabezados, etiquetas y mensajes se establece de acuerdo a la relevancia que tienen, es decir, si se busca que sean más llamativos al usuario, se recomienda elegir un tamaño significativamente diferente al del texto del cual se busca variar, por lo que el texto diferenciado se convierte en un elemento central

que atrae la atención del usuario. Posiblemente esto provoque que en opinión de los usuarios el tamaño del texto resulte ser más adecuado. También se recomienda tener en cuenta las características personales del usuario tales como edad y capacidad visual para decidir el tamaño del texto que se utilizará en una GUI.

El sub-factor **tipo de letra** resultó no ser significativo, esto se pudiera deber a que la teoría Gestalt no contempla ningún principio específicamente relacionado con los tipos de letra o con las características que debiera tener la fuente utilizada en una GUI, además de que en ambos diseños se utilizó una fuente muy similar. Aunque la calidad en las GUIs con Gestalt resultó ser mayor en éste aspecto, no fue lo suficiente para ser considerada estadísticamente significativa. No obstante, se recomienda para construir GUIs con calidad no utilizar demasiadas fuentes, ya que podrían afectar la consistencia, para textos largos se recomienda utilizar tipografía clara como: Times New Roman, Arial o Verdana.

Respecto al sub-factor **Contraste entre color de fondo y color de texto** se encontró una diferencia significativa, esto pudiera deberse al empleo del principio de la teoría Gestalt Figura-Fondo, el cual promueve el contraste en la visibilidad de los elementos para poder ser apreciados de manera más clara (Densey, Lawrence & Juhani, 2001). También es importante resaltar que para construir GUIs con calidad se recomienda no utilizar colores fluorescentes en el fondo ni en texto, utilizar color para distinguir secciones y que éste sea consistente en toda la aplicación.

En el sub-factor **Redacción del texto**, se encontró una diferencia no significativa. Esto se pudiera deberse a que la redacción en ambas GUIs fue la misma y a que la teoría Gestalt no contempla específica y concretamente este aspecto como parte de sus principios. No obstante, es importante que para construir GUIs con calidad se recomienda utilizar una redacción sencilla y clara, evitar el lenguaje técnico lo más posible y utilizar frases cortas y comprensibles.

En relación al sub-factor **Significado de símbolos** resultó haber una diferencia no significativa. Pese a que la Teoría Gestalt sugiere el principio de correspondencia isomorfa, el cual se refiere al significado de las imágenes o símbolos de acuerdo a la experiencia del usuario, la calidad en las GUIs con Gestalt resultó ser mayor en éste aspecto, pero no lo suficiente para ser considerada estadísticamente significativa, esto pudiera deberse a que en ambas GUIs se utilizó simbología similar. También es importante resaltar que para construir GUIs con calidad se recomienda utilizar simbología sencilla que le sea familiar al usuario, de ésta manera le será más fácil asociar el símbolo con su significado.

Respecto al sub-factor **Tareas similares** se encontró una diferencia no significativa. Esto pudo deberse a que en ambas GUIs con Gestalt y sin Gestalt la manera de realizar tareas es consistente, es decir, las operaciones relacionadas se activan y funcionan de manera similar. Aunque la calidad en las GUIs con Gestalt resultó ser mayor en éste aspecto, no fue lo suficiente para ser considerada estadísticamente significativa. No obstante, es importante que para construir GUIs con calidad se recomienda mantener la consistencia en las acciones o tareas a realizar, de manera que al usuario le sea simple asociarlas e identificarlas.

En el sub-factor **Control al usuario** se encontró una diferencia significativa, esto se puede deber a que un usuario siente que tiene el control cuando se siente seguro y a gusto en el sistema que opera, esto se logra a través de los principios de Gestalt de Armonía y Simetría (Densey, Lawrence & Juhani, 2001), los cuales implican congruencia entre los elementos de la interface, lo que proporciona al usuario una sensación de equilibrio en lo que percibe y manipula; de la misma manera, el principio de Simplicidad (Densey, Lawrence & Juhani, 2001) busca proporcionar al usuario mensajes gráficos despejados, de manera que no se sienta abrumado y confundido, sino por el contrario, que se sienta seguro de la situación que maneja. También es importante resaltar que para construir GUIs con calidad se recomienda para dar el control al usuario mediante el diseño de GUIs flexibles y con una interacción sencilla.

En relación al sub-factor **control de errores** se encontró una diferencia significativa, esto se puede deber al empleo del principio de la teoría Gestalt Simplicidad (Densey, Lawrence & Juhani, 2001), el cual busca reducir la probabilidad de que el usuario cometa errores por medio de un diseño simple. No obstante, es importante que para construir GUIs con calidad se recomienda que el diseño se anticipe a los errores que pueda cometer el usuario, por medio de una organización e interacción sencilla (Pressman, 2007; Sommerville, 2006).

Respecto al sub-factor **Claridad de mensajes**, este resultó existir una diferencia no significativa, esto pudiera deberse a que la teoría Gestalt no contempla concretamente ningún principio que guíe en la redacción de mensajes, por lo que para el diseño de éstos Gestalt no tiene influencia alguna. Aunque la calidad en las GUIs con Gestalt resultó ser mayor en éste aspecto, no fue lo suficiente para ser considerada estadísticamente significativa. También es importante resaltar que para construir GUIs con calidad se recomienda que los mensajes sean claros, y sencillos, de manera que proporcionen al usuario información que le sea comprensible y fácil de entender (Pressman, 2007; Sommerville, 2006).

En el sub-factor **Ayuda de mensajes** se encontró una diferencia no significativa, esto se puede deber a que la teoría Gestalt no contempla guías que proporcionen ayuda para el diseño del contenido de mensajes, por lo que su aplicación en éste aspecto resulta irrelevante. No obstante, es importante que para construir GUIs con calidad se recomienda que los mensajes dirigidos a los usuarios contengan información suficiente sin resultar demasiado complejos, así como que sean informativos y constructivos, de forma que causen en el usuario un impacto positivo, ayudándolos a realizar la acción que desean (Pressman, 2007; Sommerville, 2006).

En relación al sub-factor **Navegación** la diferencia resultó no ser significativa. La teoría Gestalt propone principios como continuidad y concisión que buscan guiar al usuario en sus movimientos dentro del sistema, pese a esto, su aplicación en el diseño de GUIs no representa una diferencia significativa estadísticamente.

También es importante resaltar que para construir GUIs con calidad se recomienda diseñar una navegación guiada de manera que el usuario sepa dónde está, hacia donde puede dirigirse y de donde procede proporcionándole pistas visuales, distinguir claramente entre la navegación principal o global y permitir un acceso claro y rápido al inicio de la aplicación.

Respecto al sub-factor **Movimientos de navegación** resultó existir una diferencia no significativa, esto pudiera deberse a que en ambas aplicaciones probadas, la cantidad de movimientos para realizar una tarea es adecuada de acuerdo a la percepción de los usuarios. Aunque la calidad en las GUIs con Gestalt resultó ser mayor en éste aspecto, no fue lo suficiente para ser considerada estadísticamente significativa. No obstante, es importante que para construir GUIs con calidad se recomienda diseñar accesos sencillos, sin demasiados movimientos, de manera que no se dificulte al usuario la interacción con las tareas que desea realizar.

En el sub-factor **Organización de elementos** se encontró una diferencia significativa, esto se puede deber principalmente a la aplicación del principio de Gestalt Proximidad (Densey, Lawrence & Juhani, 2001), el cual sugiere como organizar los diferentes elementos de acuerdo a sus características; así como al principio de simplicidad y similaridad. También es importante resaltar que para construir GUIs con calidad se recomienda organizar los elementos de la interfaz de forma balanceada, clara y concisa, de forma que se guíe al usuario visualmente; así como la utilización de espacios en blanco como elementos de diseño.

En relación al sub-factor **Combinación de colores** resultó existir una diferencia significativa, lo cual pudiera deberse a la aplicación del principio de la teoría Gestalt Figura-Fondo (Densey, Lawrence & Juhani, 2001). Además, se utilizaron los colores propios de la institución, por lo que los usuarios pudieron sentirse más identificados. No obstante, es importante que para construir GUIs con calidad se recomienda evitar el uso de colores muy oscuros, emplear colores de la misma gama o tono y utilizarlos de manera consistente en toda la aplicación de software.

En el sub-factor **Color de fondo** se encontró una diferencia significativa, esto se puede deber principalmente a la aplicación del principio Figura-Fondo (Densey, Lawrence & Juhani, 2001). También es importante resaltar que para construir GUIs con calidad se recomienda utilizar para el fondo colores claros y armónicos en los que el texto sea más legible, de manera que resulte cómodo para el usuario

Respecto al sub-factor **Atractividad de símbolos** resultó existir una diferencia significativa, esto se puede deber al empleo de los principios de Punto Focal, Correspondencia Isomorfa y Simplicidad (Densey, Lawrence & Juhani, 2001). No obstante, es importante que para construir GUIs con calidad se recomienda utilizar símbolos e íconos que llamen la atención del usuario sin distraerlo, que sean agradables a la vista y tengan un significado coherente.

En relación al sub-factor **Recordar acciones anteriores** se encontró una diferencia no significativa, esto se puede deber a que no se utilizaron pistas visuales en la interfaz, ya que Gestalt no involucra concretamente ningún principio que se refiera a éste aspecto. Aunque la calidad en las GUIs con Gestalt resultó ser mayor, no fue lo suficiente para ser considerada estadísticamente significativa. También es importante resaltar que para construir GUIs con calidad se recomienda incluir pistas visuales que ayuden al usuario a recordar cómo realizar tareas que ya realizó anteriormente.

Respecto al sub-factor **Tareas fáciles de comprender** resultó existir una diferencia no significativa. La teoría Gestalt propone principios que buscan hacer los elementos visuales más intuitivos, y por lo tanto, más fáciles de comprender. Aunque la calidad en las GUIs con Gestalt resultó ser mayor en éste aspecto, no fue lo suficiente para ser considerada estadísticamente significativa. No obstante, es importante que para construir GUIs con calidad se recomienda organizar los elementos intuitivamente, utilizar metáforas que le sean familiares al usuario, esto para contribuir con la comprensión de las tareas que se deseen realizar.

En el sub-factor **Tareas fáciles de recordar** se encontró una diferencia significativa, esto pudiera deberse a que los usuarios usaron en meses anteriores

una aplicación similar, por lo que ya estaban más familiarizados con las tareas a realizar. También es importante resaltar que para construir GUIs con calidad se recomienda proporcionar pistas visuales al usuario, definir accesos intuitivos, así como utilizar metáforas tomadas de la realidad.

En relación al sub-factor **Información ordenada de manera jerárquica** resultó existir una diferencia significativa, esto puede deberse a que la aplicación del principio de la teoría Gestalt Proximidad (Densey, Lawrence & Juhani, 2001), el cual contribuye a organizar la información y los elementos de la interfaz de manera jerárquica. No obstante, es importante que para construir GUIs con calidad se recomienda desglosar la información de manera progresiva, las tareas deben presentarse en un alto grado de abstracción, y posteriormente mostrar paulatinamente los detalles.

Respecto al sub-factor **Cantidad de elementos** resultó existir una diferencia no significativa, lo cual pudiera deberse a que en ambas aplicaciones no existían demasiados elementos en la interfaz, por lo que no hubo una gran diferencia. Aunque la calidad en las GUIs con Gestalt resultó ser mayor en éste aspecto, no fue lo suficiente para ser considerada estadísticamente significativa. También es importante resaltar que para construir GUIs con calidad se recomienda reducir la carga de memoria en el usuario, es decir, mantener la simplicidad en el diseño de las GUIs, ya que cuantas más cosas tenga que recordar el usuario, más probabilidades habrá de cometer errores al interactuar con el sistema.

En el sub-factor **Beneficio percibido** se encontró una diferencia significativa, esto pudiera deberse a que la implementación de los principios de Gestalt (Densey, Lawrence & Juhani, 2001) logran que la interfaz sea más estética y amigable al usuario, de manera que realizar las tareas resulta más agradable y sencillo, de esta manera se percibe un beneficio. Un usuario percibirá que un software le proporciona beneficios si le ayuda a cumplir un objetivo, es decir, a realizar las tareas que desea. No obstante, es importante que para construir GUIs con calidad

se recomienda construir un diseño que guíe y ayude al usuario a realizar sus tareas de forma que cumpla sus objetivos y satisfaga sus necesidades.

En relación al sub-factor **Confianza** resultó existir una diferencia significativa, esto pudiera deberse a que la aplicación de los principios de Gestalt (Densey, Lawrence & Juhani, 2001) ayuda a construir GUIs armónicas y equilibradas de manera que estos atributos son agradables a la vista del usuario, y se cree que pueden inspirar confianza. Así mismo se recomienda para construir GUIs con calidad basarse en diseños amigables, con elementos sencillos que inspiren al usuario una sensación de confianza y equilibrio.

Respecto al sub-factor **Satisfacción de necesidades** resultó existir una diferencia significativa, esto puede deberse a que los principios de la teoría Gestalt (Densey, Lawrence & Juhani, 2001) ayudan a construir GUIs simples que facilitan la interacción entre el usuario y la aplicación, de manera que colaboran con la satisfacción de necesidades del usuario. Un usuario se sentirá satisfecho con un producto dependiendo de la interacción que tenga con éste. Además se recomienda para construir GUIs con calidad verificar que los requisitos expuestos por los usuarios desde etapas iniciales estén cubiertos, esto con la finalidad de satisfacer sus necesidades.

5.2 Limitaciones de la investigación

Resulta importante resaltar que la investigación realizada se llevó a cabo solo en el estado de Zacatecas con estudiantes de una institución de nivel medio superior, de un solo semestre, por lo cual, los resultados que se reportan solo son válidos en éste contexto y no se pueden generalizar para toda la población. Los resultados pudieran variar si el estudio se reproduce en poblaciones diferentes a la aquí mencionada.

Otra limitante identificada es el tiempo de exposición de los usuarios al sistema, ya que pudiera no haber sido suficiente para familiarizarse con su uso. En

consecuencia, si se amplía el tiempo de exposición a los participantes los resultados podrían ser diferentes.

Por otra parte, otro aspecto que pudiera representar una limitación es que Gestalt no pretende ser una guía de diseño, sino que solo proporciona una serie de principios acerca de cómo la mente percibe los elementos que llegan a ella por medio de los sentidos, por lo tanto su aplicación como principios de diseño depende de cada persona, lo que puede representar una variación en la aplicación de dichos principios, pudiendo ser más amplia o diferente. Para efectos de ésta investigación la aplicación se basó en ejemplos de investigaciones realizadas previamente y en la interpretación personal.

Los resultados obtenidos pueden ser verdaderos únicamente para aplicaciones Web tal como la utilizada en la presente investigación, en consecuencia, si se desarrolla un sistema en otra tecnología los resultados podrían variar.

Pueden existir algunas otras limitaciones y/o variables relacionadas, así como estudios similares no identificadas al momento de conducir la presente investigación, las cuales podrían tener un efecto, y en consecuencia, modificar los resultados obtenidos.

5.3 Recomendaciones

Con la finalidad de mejorar y ampliar el presente trabajo de investigación se recomienda reproducir el presente estudio en más escuelas de Zacatecas, así como en más estados de la República Mexicana, de manera que los resultados pudieran ser representativos de toda la población y así generalizarse.

Adicionalmente, se recomienda aumentar el tiempo de exposición de los usuarios al sistema, con la finalidad de aumentar la familiaridad de los usuarios conduciendo el estudio de manera longitudinal. Es decir, evaluar al tener la exposición, en uno o más puntos a la que los participantes estén expuestos, y al final de la misma. De esta manera, se podría evaluar si cambia la opinión de cada uno, así como del grupo de una medición a otra.

También podría ser interesante realizar el estudio con diferentes segmentos de la población, por ejemplo adultos mayores, niños o personas con capacidades diferentes. Al hacerlo, se pueden hacer comparaciones entre distintos grupos de edad o capacidades. Logrando un estudio que permite obtener directrices más amplias para los desarrolladores.

Se recomienda desarrollar un sistema diferente en tecnología diferente de forma tal que se pueda identificar si la teoría Gestalt tiene el mismo efecto que lo aquí identificado.

Por último, se recomienda repetir el presente estudio incorporando nuevas investigaciones relacionadas al tema de forma tal que se identifiquen todas las variables que podrían haber sido ignoradas involuntariamente o no identificadas para obtener un conjunto de factores críticos en el desarrollo de sistemas que son de vital importancia para la aceptación por el usuario final.

Anexos

Productividad





ESTUDIO DEL IMPACTO DE LAS TEORÍAS GESTALT EN LA CALIDAD DEL SOFTWARE

Echavarría Álvarez Erika Guadalupe,
Gómez Reynoso Juan Manuel.

INTRODUCCIÓN: Una de las mayores preocupaciones de la industria del software es el desarrollo de productos con calidad. El software de computadora, es la tecnología más importante en el ámbito mundial (2007). A pesar de su importancia, existen muchos ejemplos de productos con mala calidad, lo que resulta en la insatisfacción de los usuarios y en pérdidas económicas [2]. La calidad del software, es juzgada de acuerdo a su interfaz de usuario (Pressman, 2007), una interfaz deficiente puede llevar al fracaso una aplicación bien diseñada. “Es posible afirmar que la interfaz de usuario es el elemento más importante de un sistema o producto de cómputo” (Pressman, Ingeniería del Software Un Enfoque Práctico, 2007, pág. 378). Es de vital importancia prestar atención a la calidad del diseño de la interfaz de usuario. En el área de diseño de interfaces existe una falta de directrices basadas en la ciencia cognitiva [3]. En el área de la psicología está la teoría Gestalt, la cual está constituida por una serie de leyes que se aplican para sugerir como presentar los elementos visuales, a fin de lograr resultados efectivos [4]. Éste trabajo propone aplicar las leyes de la teoría Gestalt al diseño de las interfaces gráficas de usuario (GUIs) y medir el impacto que dicha teoría pudiera tener en la calidad del software. La literatura no reporta un estudio

en el cual se analice dicho impacto desde el punto de vista de calidad.

OBJETIVO: Medir el impacto del diseño de las interfaces gráficas de usuario utilizando las teorías Gestalt en la calidad del software, tomando como elementos de medición la facilidad de uso, la facilidad de aprendizaje y el grado de aceptación por parte del usuario.

METODOLOGÍA: Se tiene 1 grupo de sujetos, de los cuales se obtienen 2 mediciones: 1. Evaluación de un sistema con GUIs desarrolladas de forma tradicional y 2. Evaluación de un sistema con GUIs desarrolladas usando los principios de la Teoría Gestalt. Ambos sistemas se evalúan utilizando un cuestionario cuyas respuestas están diseñadas en una escala de likert de 7 puntos; las preguntas están basadas en los estándares de producción de software con calidad, establecidos por ISO 9126 y la teoría de McCall. La confiabilidad del instrumento está validada por medio de un estudio piloto al que se le realizó una prueba de normalidad y alfa de Cronbach.

RESULTADOS: Al terminar la investigación se espera haber evaluado el impacto de las teorías Gestalt en la calidad del software. Se espera que los sistemas bajo Gestalt sean más aceptados con respecto a los que no están construidos con dicha teoría.

CONCLUSIONES: Se considera importante evaluar la calidad del software desde el punto de vista del producto, ya que es éste con el que el usuario está en contacto. Es de relevante importancia

Autónoma de Aguascalientes. UAA
Av. Universidad # 740 C.P. 20100, Aguascalientes México
Maestría en Ciencias Exactas, Sistemas y de la Información,
Área: Ingeniería de software
lerik_g@hotmail.com
Teléfono: 449 1122047



realizar estudios en torno a la calidad de la interfaz de usuario dado que por medio de ésta se juzga la calidad de un sistema en su totalidad. Al terminar la investigación se espera contribuir con conocimiento acerca de la evaluación de la calidad del software y el diseño de interfaces gráficas de usuario.

BIBLIOGRAFÍA:

- [1] R. S. Pressman, Ingeniería del Software Un Enfoque Práctico. México: McGrawHill, 2007.
- [2] T. Howles, "Widespread Effects of Deffects," Wilson Web, pp. 58-63, 2003.
- [3] G. Aberg and J. Chang, "Applying Cognitive Science Research in Graphical User Interface (GUI)," Umea Institute of Design, pp. 23-28, 2005.
- [4] C. Dempsey, D. Laurence, and T. E. Juhani, "Gestalt Theory in Visual Screen Design A New Look at an Old Subject," Monash University, pp. 1-8, 2001



Estudio del impacto de las teorías Gestalt en la calidad del software

E.G. Echavarría Álvarez, J.M. Gómez Reynoso

Resumen: La calidad del software representa un importante reto en la actualidad, a menudo se encuentran numerosos ejemplos de software que carecen de ésta. La frustración y la ansiedad son parte de la vida cotidiana de muchos usuarios de sistemas de información computarizados. Con la finalidad de lograr productos de calidad se recurre al empleo de estándares como el ISO/IEC 9126 y el modelo de McCall, los cuales se utilizan en la presente investigación. El usuario tiende a juzgar la calidad del software basándose en la interfaz gráfica de usuario (GUI), por lo que resulta vital realizar una evaluación previa de ésta. Las métricas ayudan a conocer mejor el producto que se ha desarrollado. En la construcción de GUIs intervienen varias disciplinas, entre ellas la psicología. El presente trabajo de investigación busca emplear la teoría psicológica Gestalt en el diseño de GUIs y evaluar el impacto que dicha teoría tiene en la calidad del software, tomando como elementos de medición la facilidad de uso, la facilidad de aprendizaje y el nivel de aceptación por parte del usuario. Al concluir la investigación se espera tener una evaluación objetiva acerca del impacto que tiene dicha teoría en la calidad del software.

Palabras clave: Calidad de software, métricas, GUI, Gestalt.

Abstract: Nowadays software quality represents a major challenge for developers. There are many examples of software with poor quality. Frustration and anxiety are part of everyday's life of many information systems end-users. In order to ensure quality, developers should use standards and models such as ISO/IEC 9126 and McCall's

model, which are used in this research. End-users tend to judge software quality based on the graphical user interface (GUI). Software metrics could help for a better understanding about the software product that is being developed. Developing GUIs involves many disciplines, including the psychology. This research seeks to use the Gestalt psychology theory in the design of GUIs to assess the impact of this theory on software quality. In order to do so, we measure elements such as ease-of-use, ease to learn and end-user acceptance. Upon completion of this research, it is expected to have a better understanding about the impact that Gestalt might have in software quality.

Keywords: Software quality, metrics, GUI, Gestalt.

Introducción

Una de las mayores preocupaciones de la industria del software es el desarrollo de productos con calidad. De acuerdo a Pressman (Pressman, Ingeniería del Software Un Enfoque Práctico, 2007), el software de computadora es la tecnología más importante en el ámbito mundial. A pesar de su importancia, existen muchos ejemplos de productos defectuosos y con mala calidad, lo que resulta en la insatisfacción de los usuarios y en pérdidas económicas [2]. Por ejemplo, Howles [2] explica que para el año 2003 se estimó que el impacto económico de los defectos del software en Estados Unidos representó alrededor del 1% del producto interno bruto, lo cual representa una idea clara de la situación.

La calidad del software es un concepto complejo del cual existen diversas definiciones, de acuerdo al IEEE [3] la calidad de un sistema, componente o proceso de desarrollo de software, se obtiene en función del cumplimiento de los requerimientos iniciales especificados por el cliente o usuario

Erika Guadalupe Echavarría Álvarez, lerik_g@hotmail.com
Juan Manuel Gómez Reynoso, jmgr@correo.uaa.mx, Universidad Autónoma de Aguascalientes, Av. Universidad # 740 C.P. 20100, Aguascalientes México



final. La calidad del software está compuesta por un conjunto de factores [4].

Para alcanzar un producto con calidad, la medición resulta esencial (Pressman, 2007) ya que solo se puede mejorar lo que se puede medir. Con la finalidad de realizar la medición y obtener una guía o punto de referencia de los atributos del software han surgido diferentes modelos de calidad [5], ejemplos de estos son: modelo de McCall, ISO/IEC 9126, Bohem, FURPS y Dromey [6]. A pesar de que no se trata de un modelo exhaustivo, el ISO/IEC 9126 constituye un estándar de calidad de software más completo hasta la fecha [7] y es el que más refleja la visión del usuario [8]. El modelo se basa en el trabajo de McCall publicado en 1977, el cual sigue siendo tan válido ahora como entonces (Pressman, 2007). La calidad en general depende del punto de vista de la persona que la juzga, tal como sucede en el software. Pressman (Pressman, Ingeniería del Software Un Enfoque Práctico, 2007) afirma que la calidad de un software es juzgada de acuerdo a su interfaz de usuario y que un mal diseño de esta reduce la capacidad para aprovechar las ventajas y funcionalidades de una aplicación. En consecuencia, una interfaz deficiente puede llevar al fracaso una aplicación bien diseñada y con una implementación sólida. Asimismo, expone que la interfaz de usuario es el elemento más importante de un sistema o producto de software. Es por esto que se debe poner especial atención en la calidad del diseño.

De forma ideal, la evaluación de una interfaz de usuario se debe llevar a cabo en torno a la usabilidad del sistema [9]. La usabilidad es uno de los factores que componen la calidad del software, y resulta no ser factible de ser medido de manera directa. ISO/IEC 9126 define la usabilidad como la capacidad de un software de ser comprendido, aprendido, usado y ser atractivo para el usuario, en condiciones específicas de uso (Pressman, Ingeniería del Software Un Enfoque Práctico, 2007).

Adicionalmente, los usuarios demandan sistemas fáciles de aprender y que les ayuden a hacer su trabajo, que no los detenga, engañe o confunda, ni les lleve a cometer errores o les dificulte la terminación del trabajo [10].

En el área de diseño de interfaces existe una falta de directrices basadas en investigaciones actuales

de la ciencia cognitiva [11]. El diseño de estas constituye un campo interdisciplinario donde participan áreas como: psicología, ergonomía, tecnología, diseño gráfico, entre otras. Las ciencias cognitivas estudian los procesos de la mente humana: como aprende, como recuerda, como procesa la información y que hace con ella [12]. El equilibrio y organización general de los elementos gráficos es determinante en la creación de interfaces gráficas [13].

En el campo de la psicología se encuentra la teoría Gestalt, la cual es considerada como la teoría formal de la percepción visual [14], dicha teoría explica como la mente percibe los elementos gráficos que llegan a ella por medio de los sentidos. La teoría está constituida por una serie de principios o leyes que se aplican para sugerir como presentar los elementos visuales, a fin de lograr resultados efectivos [15].

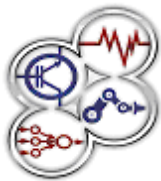
En el presente trabajo se propone aplicar las leyes de la teoría Gestalt al diseño de las GUIs y medir el impacto que dicha teoría podría tener en la calidad del software. Hasta el momento no se ha encontrado en la literatura un estudio similar, por lo que podría ser importante medir el impacto del diseño, con esta aproximación tomando como elementos de medición la facilidad de uso, la facilidad de aprendizaje y el grado de aceptación por parte del usuario.

Teoría Gestalt

La teoría Gestalt surge en 1912 en Alemania, teniendo como principales autores a Wertheimer, Koffka y Köler [16]. Inicialmente, dicha teoría fue estudiada solamente en el campo de la psicología, pero sus conceptos se han aplicado en diversas áreas de la ciencia [17], por ejemplo: diseño de diagramas [18], patrones de lenguaje [19], estética de interfaces de usuario [20], entre otras.

La palabra Gestalt recibe el significado de forma [21]. Propone una serie de principios que tratan de explicar cómo organizamos elementos individuales dentro de grupos, por lo tanto, se puede utilizar para explicar la forma en que percibimos y reconocemos patrones [17].

De entre las leyes que componen dicha teoría sobresalen 11, las cuales representan los principales aspectos de esta acerca de las formas



visuales [15]. A continuación se describen brevemente dichas leyes:

Similaridad

Los elementos visuales que son similares en forma, tamaño, color, proximidad y dirección son percibidos como parte de un grupo, incluso si los elementos están separados [22] (Ver Figura 1).

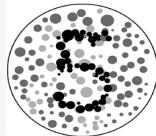


Figura 1. Similaridad (Adaptado de [22])

Continuidad

El ojo humano está diseñado para crear relaciones entre las formas (Ver Figura 2). La continuidad se produce cuando el ojo sigue a lo largo de una línea recta, curva, o de una secuencia de formas, incluso cuando cruza sobre las formas negativas y positivas [22].



Figura 2. Continuidad

Punto focal

El principio del punto focal se refiere al elemento que captura la atención entre un grupo de elementos (Ver Figura 3). La gente percibe los elementos como puntos focales si los atributos de esos elementos son significativamente diferentes a los demás [15].



Figura 3. Punto focal (Adaptado de [15])

Figura fondo

El principio ayuda a identificar los objetos (figura) a diferencia de su superficie (fondo) (Ver Figura 4). Esta ley de percepción depende del contraste. Las imágenes y el texto deben ser visibles para poder ser entendidas [22]. Por ejemplo:



Figura 4. Figura fondo (Adaptado de [22])

Unidad/Armonía

El principio de Unidad permite que un solo elemento pueda pertenecer a una sola fuente o grupo [17] (Ver Figura 5). La unidad implica una congruencia o acuerdo entre los elementos de un grupo [15].



Figura 5. Unidad/Armonía (Adaptado de [15])

Balance

El principio de balance también es llamado principio de simetría. La gente tiende a sentirse más cómoda con un sentido de equilibrio al ver, oír o tocar (Ver Figura 6). La idea de éste principio es el de lograr una ponderación igual de los atributos cuando se muestra un elemento [17].

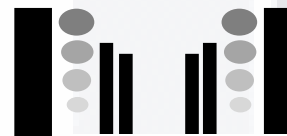


Figura 6. Balance

Proximidad

Los objetos que están ubicados cerca uno del otro parecen parte de un grupo, mientras que los que están ubicados de manera distante, parecen ser de un grupo distinto [22] (Ver Figura 7).

Amarillo	Durazno	Amarillo	Morado	
Morado	Fresa	Durazno	Fresa	Azul
Azul	Pera		Pera	Rojo
Rojo	Uva	Verde	Uva	
Verde	Mandarina		Mandarina	Manzana
Negro	Manzana	Negro	Naranja	
Blanco	Naranja		Piña	Blanco
	Piña			

Figura 7. Proximidad (Adaptado de [22])

Cierre

Los seres humanos tienen una tendencia natural a cerrar visualmente los huecos en una forma,



especialmente en formas familiares (Ver Figura 8). Cuando falta información, nos centramos en lo que está presente, se tiende a ignorar las partes que faltan y rellenar los huecos con una línea familiar. Buscamos cerrar formas para que sean estables [22].



Figura 8. Cierre

Correspondencia isomorfa

El principio propone que no todas las imágenes tienen el mismo significado para todos, sino que las interpretamos según nuestra experiencia [15](Ver Figura 9).



Figura 9. Correspondencia isomorfa (Adaptado de [15])

Concisión/Pregnancia

También es llamado el principio de buena forma. Una buena forma es un diseño simple o un diseño simétrico (Ver Figura 10).



Figura 10. Concisión/Pregnancia

Simplicidad

La simplicidad funciona bien si el mensaje gráfico es despejado, pero si los gráficos son complejos o ambiguos el proceso de percepción del mensaje resulta confuso (Ver Figura 11).



Figura 11. Simplicidad

Dichas leyes o principios pueden ser interpretados como bases o reglas para el diseño de interfaces de usuario, con ellas se busca construir productos cuyos elementos sigan una estructura que la mente reconozca y asocie de manera natural, para con ello buscar que la interfaz resulte con un alto grado de usabilidad y en consecuencia lograr la calidad. Galitz [23] reporta que ya se han utilizado los principios de la teoría Gestalt en el diseño de interfaces de usuario en un estudio realizado por Tullis en 1981, que a su vez se basaba en otro estudio realizado por Zahn en 1971, en los cuales se tuvo como resultado la reducción de tiempos de localización visual de elementos cuando se utilizaba el principio de proximidad principalmente.

Planteamiento de la investigación

Jurán [24] explica que la satisfacción del cliente es el resultado alcanzado cuando las características del producto responden a las necesidades del cliente; generalmente es sinónimo de satisfacción con el producto, de ésta manera, dicho producto podría ser percibido como de calidad. Así mismo expone que si un cliente se siente satisfecho con un producto continuará adquiriéndolo. La insatisfacción con un producto tiene su origen en las no-conformidades con las características definidas para el mismo, y es debido a esto que los clientes pueden presentar quejas al respecto. De aquí que éstos consideren que el producto carece de calidad. Los productores emplean como una estrategia competitiva el factor calidad ya que puede permitir el establecimiento de un punto de diferenciación entre las opciones disponibles. Es importante mencionar que la calidad de cualquier producto o servicio es vital para su éxito.

El International Organization for Standardization (ISO) sobresale como uno de los más importantes proveedores de estándares de calidad para el área de tecnología. Algunos de los orientados hacia la calidad del software son: ISO/IEC 9126, ISO/IEC 14598, ISO 9241-11. Sommerville [9] argumenta que en el ámbito del desarrollo de software, algunas personas piensan que la calidad se puede lograr definiendo estándares y procedimientos organizacionales de calidad que comprueban si



estos estándares son seguidos por el equipo de desarrollo. Asimismo, proporciona un argumento de que los estándares deben encapsular las buenas prácticas, las cuales nos llevan inevitablemente a productos de alta calidad. Adicionalmente, explica que mientras estándares y procedimientos son las bases de la gestión de la calidad, los gestores de calidad experimentados reconocen que hay aspectos intangibles en la calidad del software (i.e. elegancia, legibilidad, etc.) que no pueden ser incorporados en los estándares.

Entre los primeros trabajos realizados referentes a la calidad del software se encuentra la clasificación de los factores propuesto por McCall (ver Figura 12), el cual establece tres áreas principales que intervienen en la calidad del software. Dicho modelo fue creado con la finalidad de mejorar y aumentar la calidad del producto.



Figura 12. Modelo de McCall

Al igual que la definición de calidad, la definición de calidad de interfaces de usuario resulta compleja ya que depende de la percepción de quien la juzga. Según Somerville [9] “de forma ideal, una evaluación de una interfaz de usuario se debe llevar a cabo en torno a la usabilidad del sistema” [9 pág. 350]. Una mala interfaz de usuario puede contribuir al error humano, incluso en daños personales y financieros, y explica que quizá la evaluación de la calidad de las interfaces sea algo en lo que los fabricantes de software no estén dispuestos a invertir, pese a su gran importancia, dado que pueden llegar a representar elevados costos.

En conclusión, la presente investigación evalúa aspectos de sistemas de información considerados como relevantes por parte de los usuarios finales de una forma estructurada y tomando como bases estándares con el fin de minimizar la subjetividad que la calidad del software tiene. Siendo esta es

una de las principales contribuciones identificadas.

Debido a que la literatura no reporta un estudio en el cual se analice el impacto que las teorías Gestalt pueden tener en el desarrollo de GUIs en la calidad desde el punto de vista de los usuarios, en el presente trabajo se propone realizar un estudio que evalúe dicho impacto, tomando como elementos de medición la facilidad de uso, la facilidad de aprendizaje y la aceptación por parte del usuario, basándose en los atributos de calidad que proponen el modelo de McCall y el ISO/IEC 9126.

Modelo de investigación

Para la realización del presente trabajo se propone el siguiente modelo de investigación, (ver Figura 13).

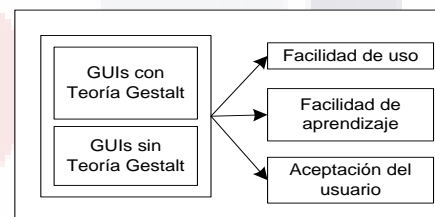


Figura 13. Modelo de investigación

Hipótesis

En la presente investigación se tienen las siguientes hipótesis:

H1: Las GUI construidas bajo los principios de la teoría Gestalt son más fáciles de usar comparadas con las GUI que se construyen sin estos principios.

H2: Las GUI construidas bajo los principios de la teoría Gestalt son más fáciles de aprender comparadas con las GUI que se construyen sin estos principios.

H3: Las GUI construidas bajo los principios de la teoría Gestalt son más aceptadas por los usuarios comparadas con las GUI que se construyen sin estos principios.

Diseño del experimento

Se tendrá 1 solo grupo de sujetos, de los cuales se obtendrán 2 mediciones:

1. Evaluación de un sistema con interfaces gráficas desarrolladas sin los principios de la teoría Gestalt.



2. Evaluación de un sistema con interfaces gráficas desarrolladas usando los principios de la Teoría Gestalt.

El sistema que se pretende evaluar es el portal Web de una institución educativa de nivel medio superior, que ya se encuentra construido, el cual será el sistema con interfaces gráficas sin teoría Gestalt. Se pretende rediseñar las interfaces de dicho sistema, basándose en los principios que propone la teoría Gestalt. Ambos sistemas serán evaluados por los alumnos del plantel mediante la aplicación de un cuestionario cuyas preguntas serán respondidas utilizando una escala de likert de 7 puntos. Las preguntas se basan en los elementos de medición, siguiendo los estándares de producción de software con calidad, establecidos por ISO/IEC 9126 en el área de usabilidad y la teoría de McCall. Además, se obtendrá la información demográfica de los participantes. Para efectos de medir la confiabilidad del cuestionario, se llevó a cabo un estudio piloto aplicando un análisis de confiabilidad a través de una prueba de normalidad y alfa de Cronbach, en el cuál todos los factores evaluados obtuvieron un valor de alfa superior a 0.81. Las métricas que se pretende utilizar están dadas por el modelo de McCall, respecto al área de operación del producto, en la cual se proponen propone 4 métricas para la evaluación del factor Facilidad de uso.

Estado actual de la investigación

Hasta el momento se ha llevado a cabo la revisión de la literatura correspondiente, se tiene construido y validado el instrumento de evaluación y se trabaja en la construcción del sistema con GUIs basadas en la teoría Gestalt. Asimismo, se han identificado una serie métricas a utilizar en el estudio.

Resultados esperados

Al terminar la investigación se espera haber evaluado el impacto de las teorías Gestalt en la calidad del software, tomando como elementos de medición la facilidad de uso, la facilidad de aprendizaje y el grado de aceptación por parte del usuario. Se espera que los sistemas bajo Gestalt sean más aceptados con respecto a los que no están construidos con dicha teoría.

Conclusión

El presente trabajo de investigación busca evaluar el impacto que podría tener el empleo de la teoría Gestalt en la calidad del software. Se considera importante evaluar la calidad del software desde el punto de vista del producto, ya que es éste con el que el usuario está en contacto. A menudo la atención se centra en la evaluación de la calidad del proceso de desarrollo, dejando de lado la calidad del producto. Se debe tomar en cuenta que la calidad del proceso no asegura la calidad del producto. Es de relevante importancia realizar estudios en torno a la calidad de la interfaz de usuario dado que por medio de ésta se juzga la calidad de un sistema en su totalidad. Al terminar la investigación se espera contribuir con conocimiento acerca de la evaluación de la calidad del software y el diseño de GUIs.

Referencias

- [1] Pressman, Roger S., *Ingeniería del Software Un Enfoque Práctico*. México : McGrawHill, 2007. p. 958.
- [2] Howles, Trudy., "Widespread Effects of Deffects." *Wilson Web*, 2003, pp. 58-63.
- [3] Dávila, Leticia and Mejía, Pedro., "Evaluación de la Calidad de Software en Sistemas de Información en internet." *CINVESTAV IPN*, 2003, pp. 1-11.
- [4] Cavano, Joseph and McCall, James A., "A Framework for the measurement of Software Quality." *ACM Software Quality Assurance*, 1978, pp. 133-139.
- [5] Khan, R, Mustafa, K and Ahson, S., *Software Quality Concepts and Practices*. Oxford U.K. : Alpha Science, 2008.
- [6] Ortega, Maryoly., "Construction of a Systemic Quality Model for evaluating a software process." *Software Quality Journal*, 2003, pp. 219-242.
- [7] Abran, Alain, Khelifi, Adel and Suryan, Witold., "Usability Meanings and Interpretations in ISO standards." *Software Quality Journal*, 2003, pp. 323-336.
- [8] Dolado, José and Fernandez, Sanz., *Medición para la gestión en la ingeniería del software*. Madrid : Ra-Ma, 2000.
- [9] Somerville, Ian., *Ingeniería del software*. Madrid : PEARSON Addison Wesley, 2006.
- [10] Constantine, L., "What DO Users Want? Engineering Usability in Software." *Windows Teach Journal*, 1995.
- [11] Aberg, Gabriel and Chang, Jessica., "Applying Cognitive Science Research in Graphical User Interface (GUI)." *Umea Institute of Design*, 2005, pp. 23-28.



- [12] Mercovich, Eduardo., "La intersección entre factores humanos, diseño gráfico, interacción y comunicación." Buenos Aires, Argentina : s.n., 1999. SIGGRAPH.
- [13] Buitrón de la Torre, Marcela., *Consideraciones para el diseño de interfaces gráficas de usuario en ambientes virtuales educativos.* Universidad Autónoma Metropolitana. México : s.n., 2004. Tesis de maestría.
- [14] Gordon, IAN E., *Theories of visual perception.* USA & Canada : Psychology Press, 2004.
- [15] Dempsey, Chang, Laurence, Dooley and Juhani, Tuovinen E., "Gestalt Theory in Visual Screen Design A New Look at an Old Subject." Monash University, 2001, pp. 1-8.
- [16] Ginger, Serge and Ginger, Anne., *La Gestalt Una terapia de contacto.* Mexico : El manual moderno, 1993.
- [17] Dempsey, Chang and Keith V., Nesbitt., "Developing Gestalt-based Design Guidelines for Multi-sensory Displays." NICTA-HCSNet Multimodal User Interaction Workshop(MMUI2005), 2006, Vol. 57.
- [18] Lemon, Krystle, et al., "An Empirical Study of the effects of Gestalt Principles on Diagram Understability." IEEE, 2007, pp. 156-165.
- [19] Flieder, Karl., "Learning from an extended context of patterns in Science of design." Converging on a "Science of Design" through the synthesis of Design Metodologies, 2007.
- [20] Youn, Kyung, et al., "Interaction Gestalt and the Design of Aesthetic Interactions." Designing Pleasurable Products and Interfaces, 2007, pp. 239-254.
- [21] Salama, Hector., *Psicoterapia Gestalt Proceso y Metodología.* Mexico, DF : Alfaomega, 2001.
- [22] Graham, Lisa., "Gestalt Theory in Interactive Media Design." Journal of Humanities & Social Sciences, 2008, Issue 1, Vol. 2, pp. 1-12.
- [23] Galitz, Wilbert., *The Essential Guide to User Interface Design: An introduction to GUI design principles and Techniques.* Indianapolis, Indiana : Wiley Publishing, Inc, 2007.
- [24] Juran, J.M., *Juran on Quality by Design.* Madrid España : Díaz de Santos, 1996.
- [25] IEEE., "IEE Standards Collection: Software Engineering ." *IEE Standard 610.12-1990.* 1993.
- [26] ISO/IEC., "Software engineering — Software product Quality Requirements and Evaluation (SQuaRE) Guide to SQuaRE." 2005.
- de la información en el área de ingeniería de software en la UAA.
Juan Manuel Gómez Reynoso, Ing. en Sistemas Computacionales, estudió la maestría en Informática y Tecnologías Computacionales y la Especialidad en Redes en la UAA, México. Posteriormente, estudió la maestría en Ciencias en Sistemas de Información y el Doctorado en Ciencias en Sistemas de Información en Claremont Graduate University, USA. Es profesor de tiempo completo de la UAA desde 1990. Actualmente es coordinador de la carrera de Ing. en Sistemas Computacionales. Ha publicado artículos en revistas internacionales y nacionales, así como participado en congresos internacionales tales como AMCIS, IIMA, Conf-IRM, SDISC, entre otros. Sus principales intereses de investigación son: Desarrollo de software mediante Teorías de Diseño de Sistemas de Información (ISDT), medición de la calidad de software, cómputo móvil, y participación de los usuarios en la evaluación de sistemas

Currículo corto de los autores

Erika G. Echavarría Álvarez Licenciada en Informática egresada de la Universidad Autónoma de Aguascalientes (UAA) en 2008. Actualmente estudia la maestría en Ciencias exactas sistemas y

Bibliografía

- Aberg, G., & Chang, J. (2005). Applying Cognitive Science Research in Graphical User Interface (GUI). *Umea Institute of Design* , 23-28.
- Abran, A., Khelifi, A., & Suryn, W. (2003). Usability Meanings and Interpretations in ISO standards. *Software Quality Journal* , 323-336.
- Adamaczyk, K. A. (2005). The impact of Memory on HCI. 1-11.
- Adaptado de Pressman, R. S. (2007). *Ingeniería del Software Un Enfoque Práctico*. México: McGrawHill.
- Ahonen, J. J., & Junttila, T. (2003). A Case Study on Quality-Affecting Problems in Software Engineering Projects. *IEEE International Conference on Software—Science, Technology & Engineering (SwSTE'03)* , 1-9.
- Als, B. S., Jensen, J., & Skov, M. (2005). Comparison of think - Aloud and Constructive Interaction in Usability Testing with Children. *ACM* , 9-16.
- Amasaki, S., Yoshitomi, T., Mizuno, O., Takagi, Y., & Kikuno, T. (2005). A New Challenge for Applying Time Series Metrics Data to Software Quality Estimation. *Software Quality Journal, Springer Science* , 177-193.
- Andersson, T., & Aimo, T. (1992). Measurement of software quality. *Abo Academi University Finland* , 1-14.
- Bosh, J., & Juristo, N. (2003). Designing Software Architectures for Usability. *International Conference on Software Engineering* , 757-758.
- Botella, P., Burgués, X., Carvallo, J., Franch, X., Grau, J. M., & Quer, J. (2004). iso iec 9126 in practice: what do we need to know? *First Software Measurement European Forum (SMEF)* .
- Buitrón de la Torre, M. (2004). *Consideraciones para el diseño de interfaces gráficas de usuario en ambientes virtuales educativos*. Tesis de maestría, Universidad Autónoma Metropolitana, México.
- Cavano, J., & McCall, J. A. (1978). A Framework for the measurement of Software Quality. *ACM Software Quality Assurance* , 133-139.
- Chua, B., & Dyson, L. E. (2004). Applying the ISO 9126 model to the evaluation of an e-learning system. *21 st ASCILITE Conference* , 184-190.

- Cole Ian, J. (2008). *Ian Cole's Work Page*. Recuperado el 2010, de <http://www-users.york.ac.uk/~ijc4/work.htm>
- Constantine, L. (1995). What DO Users Want? Engineering Usability in Software. *Windows Teach Journal* .
- Curtis, B. (1984). FIFTEEN YEARS OF PSYCHOLOGY IN SOFTWARE ENGINEERING: INDIVIDUAL DIFFERENCES AND COGNITIVE SCIENCE. *IEEE* , 97-106.
- Dávila, L., & Mejía, P. (2003). Evaluación de la Calidad de Software en Sistemas de Información en internet. *CINVESTAV IPN* , 1-11.
- Dempsey, C., & Keith V., N. (2006). Developing Gestalt-based Design Guidelines for Multi-sensory Displays. *NICTA-HCSNet Multimodal User Interaction Workshop(MMUI2005)* , 57.
- Dempsey, C., Laurence, D., & Juhani, T. E. (2001). Gestalt Theory in Visual Screen Design A New Look at an Old Subject. *Monash University* , 1-8.
- Dolado, J., & Fernandez, S. (2000). *Medición para la gestión en la ingeniería del software*. Madrid: Ra-Ma.
- Evan, E. (1989). A heuristic for software evaluation and selection. *John Wiley & Sons, Inc. New York, NY, USA* , 707-717.
- Fernandez Sanz, L. (15 de Septiembre de 2008). *La importancia de la calidad del software: Baquia knowledge Center*. Recuperado el 5 de Abril de 2010, de Baquia knowledge Center: <http://www.baquia.com/articulos/software/noticia/14128/la-importancia-de-la-calidad-del-software>
- Flieder, K. (2007). Learning from an extended context of patterns in Science of design. *Converging on a "Science of Design" through the synthesis of Design Metodologies* .
- Florac, W. A. (1992). *Software Quality Measurement:A Framework for Counting Problems and Defects*. Pittsburgh, Pennsylvania: Carnegie Mellon University.
- Galitz, W. (2007). *The Essential Guide to User Interface Design: An introduction to GUI design principles and Techniques*. Indianapolis, Indiana: Wiley Publishing, Inc.
- Ginger, S., & Ginger, A. (1993). *La Gestalt Una terapia de contacto*. Mexico: El manual moderno.
- Gordon, I. E. (2004). *Theories of visual perception*. USA & Canada: Psychology Press.
- Graham, L. (2008). Gestalt Theory in Interactive Media Design. *Journal of Humanities & Social Sciences* , 2 (1), 1-12.

- Groocock. (2005). *La Cadena de la Calidad*. Díaz de Santos.
- Guillaume, P. (1975). *La Psychologie de la Forme*. Buenos Aires, Argentina: Psique.
- Hair, J. F., Anderson, R. E., Tatham, R. L., & Black, W. C. (1998). *Multivariate Data Analysis. Fifth edition*. Upper Saddle River, NJ : Prentice Hall.
- Hansen, B. L., & Ghare, P. M. (1990). *Control de calidad: Teoría y Aplicaciones*. Díaz de Santos.
- Henderson, R. D., Smith, M., Podd, J., & Varela-Alvarez, H. (1995). A comparison of the four prominent user-based methods for evaluating the usability of computer software. *Ergonomics, Vol 38* , 2030 - 2044 .
- Howles, T. (2003). Widespread Effects of Defects. *Wilson Web* , 58-63.
- IEEE. (1990). *IEEE Standard Glossary of Software Engineering Terminology*. New York, USA: IEEE Standards Board.
- INEGI. (2005). *Instituto Nacional de Estadística y Geografía*. Obtenido de <http://cuentame.inegi.org.mx/poblacion/asistencia.aspx?tema=P>
- ISO/IEC. (2005). Software engineering — Software product Quality Requirements and Evaluation (SQuaRE) Guide to SQuaRE.
- ISO (1991), Vol. ISO 9126 (Ed, ISO).
- Juran, J. (1996). *Juran on Quality by Design*. Madrid España: Díaz de Santos.
- Juran, J. M., Nicolau, J., & Gozalbes, M. (1990). *Juran y la planificación para la calidad*. Díaz de Santos.
- Kerlinger, F. N., & Lee, H. B. (2000). *Foundations of Behavioral Research. Fourth Edition*. New York: Harcourt College Publishers.
- Khan, R., Mustafa, K., & Ahson, S. (2008). *Software Quality Concepts and Practices*. Oxford U.K.: Alpha Science.
- Ko, A. J., Myers, B. A., & Horng Chau, D. (2006). A Linguistic Analysis of How People Describe Software Problems. *Visual Languages and Human-Centric Computing (VL-HCC'06)* , 1-8.
- Kumar, A., Grover, P., & Kumar, R. (2009). A Quantitative Evaluation of Aspect-Oriented Software Quality Model. *ACM SIGSOFT Software Engineering* , 1-9.
- Lemon, K., Allen, E. B., Carver, J. C., & Bradshaw, G. L. (2007). An Empirical Study of the effects of Gestalt Principles on Diagram Understability. *IEEE* , 156-165.

- Lewis, W. E. (2009). *Software Testing and Continuous Quality improvement*. Boca Raton, FL: Taylor & Francis.
- Machiraju, V. (1996). A survey on Research in Graphical User Interfaces. *Utah University* , 1-16.
- Mandel, T. (1997). The Golden Rules of User Interface Design. En T. Mandel, *The Elements of User Interface Design* (pág. 28). John Wiley & Sons.
- Martínez, S. (2007). *Biblioteca digital: conceptos, recursos y estándares*. Alfagrama .
- McCall A., J., Richards, P., & Walters, G. (Noviembre de 1977). Factors in software Quality, Concept and definitions of software quality. *Final Technical Report RADC-TR-369 Vol I* . General Electric Company.
- Mercovich, E. (1999). La intersección entre factores humanos, diseño gráfico, interacción y comunicación. *SIGGRAPH*. Buenos Aires, Argentina.
- Milicic, D. (2005). Software Quality Models and Philosophies. En P. Berander, L.-O. Damm, J. Eriksson, T. Gorschek, K. Henningsson, P. Jonsson, y otros, *Software quality attributes and trade-offs* (pág. 100). Suecia: Blekinge Institute of Technology.
- Minguet, J. M., & Hernandez, J. (2003). *La calidad del software y su medida*. Madrid: Centro de Estudios Ramón Areces.
- Montaudon, C. (2004). *Historia de la calidad Mundial*. Puebla: Univ. Iberoamericana de Puebla.
- Mordecki, D. (2001). Intuición y creatividad tienen puntos de contacto y fricción en la interfaz. *Faz* , 15-18.
- Nasib S., G. (2005). Factors Affecting Effective Software Quality Management Revisited. *ACM SIGSOFT Software Engineering Notes* , 1-4.
- Neumann, P. G. (Enero de 2009). *Forum on Risks to the public in computers and related systems*. Recuperado el 2010, de The risks digest: <http://catless.ncl.ac.uk/Risks/25.51.html#subj1>
- Newman, W. L. (2000). *Social Research Methods, 5ta. Edición*. Boston MA: Allyn and Bacon.
- Nielsen Research, C. (2009). *Nielsen Research*. Recuperado el 2010, de Nielsen: http://en-us.nielsen.com/content/nielsen/en_us/industries/media.html
- Nielsen, J. (2005). *Jakob Nielsen's Website*. Recuperado el 2010, de http://www.useit.com/papers/heuristic/heuristic_list.html

- Ortega, M. (2003). Construction of a Systemic Quality Model for evaluating a software process. *Software Quality Journal* , 219-242.
- Perlman, G. (1996). Practical Usability Evaluation. *Conference on Human Factors in Computing Systems* , 348-349.
- Piattini, M. G., & García, F. (2008). *Medición y estimación del software: técnicas y métodos para mejorar la calidad y productividad*. México DF: Alfaomega.
- Preece, J., Rogers, Y., Sharp, H., Beyon, D., Holland, S., & Carey, T. (1994). *Human-computer interaction*. Harlow England: Adison-Wesley.
- Pressman, R. S. (2007). *Ingeniería del Software Un Enfoque Práctico*. México: McGrawHill.
- Pyzdek, T. (2003). *Quality Engineering Handbook*. New York: Marcel Dekker, Inc.
- Real Academia, E. (2001). *Diccionario de la Lengua Española*. Recuperado el 2009, de Diccionario de la lengua Española: <http://www.rae.es/rae.html>
- Salama, H. (2001). *Psicoterapia Gestalt Proceso y Metodología*. Mexico, DF: Alfaomega.
- Sharp, H., Rogers, Y., & Preece, J. (2007). *Interaction Design, beyond human - computer interaction*. England: Jhon Wiley & Sons, Ltd.
- Shneiderman, B. (1999). Human Values and the Future of Technology: A declaration of Responsibility. *Computer and Society* , 1-9.
- Shneiderman, B. (1981). PUTTING THE HUMAN FACTOR INTO SYSTEMS DEVELOPMENT. *SIGCPR '81: Proceedings of the eighteenth annual computer personnel research conference* , 1-13.
- Shneiderman, B. (2000). Universal Usability. *Communications of the ACM* , 7.
- Sibisi, M., & Van Waveren, C. C. (2007). A Process Framework for Customising Software Quality Models. *IEEE* , 1-7.
- Sinay, S., & Blasberg, P. (1997). *Gestalt para principiantes*. Era Naciente.
- Somerville, I. (2006). *Ingeniería del software*. Madrid: PEARSON Addison Wesley.
- Tognazzini, B. (2001). *Ask Tog*. Recuperado el 2010, de Ask Tog: <http://www.asktog.com/basics/firstPrinciples.html>
- Uvigo. (2005). *Universidad de Vigo, Diseño Industrial*. Recuperado el 2009, de <http://webs.uvigo.es/disenoindustrial/docs/Introduccion%20a%20la%20calidad.pdf>

- Vega, C. (2008). *Mejores prácticas para el establecimiento y aseguramiento de la calidad del software*. Veracruz: Universidad Cristobal Colón.
- Vera Campos, J. (Octubre de 2006). *Mati Sobre la letra digital: Diseño gráfico fundamental*. Recuperado el 2010, de Mati Sobre la letra digital:.
- Youn, K., Stolterman, E., Jung, H., & Donaldson, J. (2007). Interaction Gestalt and the Design of Aesthetic Interactions. *Designing Pleasurable Products and Interfaces* , 239-254.

