



UNIVERSIDAD AUTONOMA DEL ESTADO DE MÉXICO
CENTRO UNIVERSITARIO UAEM TEXCOCO



MODELO DE MEDICIÓN DE LA USABILIDAD DEL PRODUCTO SOFTWARE
CON UN SISTEMA DE MÉTRICAS ASISTIDA POR COMPUTADORA

TESIS

PARA OBTENER EL GRADO DE

MAESTRA EN CIENCIAS DE LA COMPUTACIÓN

PRESENTA

L.C. Ma. Dolores Arévalo Zenteno

TUTOR ACADÉMICO

M. en C. Sergio Ruiz Castilla

TUTORES ADJUNTOS

M. en I. S. Ma. De Lourdes Hernández Prieto

M. en I. S. C. Cuauhtemoc Hidalgo Cortés

*A mis padres
en agradecimiento a su confianza en mí*

Índice de Contenido

Introducción.....	9
Planteamiento del problema	10
Propósito de la investigación.....	13
Objetivo General.....	13
Objetivos particulares.....	13
Justificación de la investigación	13
Hipótesis	14
1 Estado del Arte.....	15
1.1 Integración de la Ingeniería de Software y la usabilidad.....	15
1.2 Patrones de usabilidad: Mejora de la usabilidad del Software desde la Arquitectura	17
1.3 Ciclo de Vida en Estrella	20
2 Calidad de software.....	22
2.1 Definición	22
2.2 Herramientas de calidad.....	23
2.2.1 Herramientas de Gestión	23
2.2.2 Herramientas de Creatividad	24
2.2.3 Herramientas Estadísticas	24
2.2.4 Herramientas de Diseño	24
2.2.5 Herramientas de Medición	25
2.3 Calidad del Software	25
2.3.1 Gestión de la Calidad del Software.....	27
2.3.2 Garantía de la Calidad del Software	28
2.3.3 Planificación de la Calidad del Software	29
3 Métricas del software	37
3.1 Clasificación de las métricas del software	37
3.1.1 Algunas clasificaciones de métricas	38
3.2 Métricas del proceso.....	40
3.3 Métricas del proyecto	41
3.4 Métricas de productos.....	42
3.5 Modelo de la planificación predictiva	43
3.6 Integración de las Métricas dentro del Proceso de Software	44
4 Usabilidad del software	46
4.1 Justificando la Usabilidad	46

4.2	Definición	47
4.3	Atributos	47
4.4	Usabilidad y accesibilidad	49
4.5	Ingeniería de la usabilidad	51
4.6	Estándares de Proceso de Desarrollo	51
4.8.2	Medición Empírica	53
4.9	Métodos de evaluación	55
4.10	Evaluación heurística	56
4.11	Ordenamiento de tarjetas (Card Sorting)	60
5	Metodología	66
5.1	Propuesta de modelo de medición de usabilidad	66
5.1.1	Elección de Métodos de Evaluación	67
5.2	Diseño del sistema <i>MEDUSA</i>	68
5.3	Selección del Sistema	70
5.4	Evaluación Heurística.....	70
5.4.1	Variables Utilizadas: Principios Heurísticos	70
5.4.2	Rangos de Severidad.....	71
5.4.3	Indicadores	72
5.4.4	Selección de evaluadores	72
5.4.5	Preparando los recursos de la Evaluación	73
5.4.6	Cuestionario a Aplicar	73
5.4.7	Práctica de la Elaboración	76
5.5	Prueba de Usuarios	76
5.5.1	Cuestionario a aplicar	76
5.5.2	Rango de Severidad	78
5.5.3	Indicadores a Utilizar	78
5.5.4	Selección de Usuarios	78
5.5.5	Preparando Recursos de la Evaluación	79
5.5.6	Proceso de Datos y Resultados.....	79
6	Conclusiones de la Investigación.....	93
6.1	Introducción	93
6.2	Metas Alcanzadas.....	93
6.3	Respuesta a las Preguntas de Investigación.....	93
6.4	Acerca de los Resultados obtenidos	94
7	Discusión	96
	Referencias	97
	Diagramas de caso de uso.....	100

Diagramas de secuencia	101
Diagrama de estado	102
Diagrama de colaboración	103

Índice de figuras

FIGURA 1 ATRIBUTOS DE LA CALIDAD DEL SOFTWARE	10
FIGURA 2 EVALUACIÓN DEL PRODUCTO FINAL	12
FIGURA 3 ESQUEMA O MODELO DEL PROCESO	16
FIGURA 4 PARADIGMA DE PATRONES DE USABILIDAD	17
FIGURA 5 RELACIONES ENTRE ATRIBUTOS, PROPIEDADES Y PATRONES.	18
FIGURA 6 RELACIONES ENTRE ATRIBUTOS, PROPIEDADES, PATRONES DE USABILIDAD Y PATRONES ARQUITECTÓNICOS	18
FIGURA 7 CICLO DE VIDA EN ESTRELLA	21
FIGURA 8 MODELOS DE CALIDAD	31
FIGURA 9 PRINCIPIOS DE LA NORMA ISO 9001	32
FIGURA 10 NORMA ISO 9001	33
FIGURA 11 MODELO EFQM 2010	34
FIGURA 12 MODELO CAF 2006	35
FIGURA 13 SIX SIGMA COMO MEJORA DE PROCESO CONTINUA	36
FIGURA 14 CONCEPTO DE MÉTRICAS	37
FIGURA 15 IMPLEMENTACIÓN DE PRÁCTICAS DE MEDICIÓN DE SOFTWARE	37
FIGURA 16 MÉTRICAS DE PREDICCIÓN Y CONTROL	40
FIGURA 17 PROCESO DE RECOPIACIÓN DE MÉTRICA	45
FIGURA 18 CARD SORTING MUESTRA EL MODELO MENTAL DE ORGANIZAR LA INFORMACIÓN	60
FIGURA 19 MODELO DE EVALUACIÓN DE LA USABILIDAD	67
FIGURA 20 CASO DE USO DE MEDUSA	68
FIGURA 21 DISEÑO DE BASE DE DATOS PARA MEDUSA	69

Índice de tablas

TABLA 1 LISTA DE PATRONES DE USABILIDAD	20
TABLA 2 CONCEPTO DE CALIDAD	23
TABLA 3 REQUERIMIENTOS DE CALIDAD	43
TABLA 4 MÉTRICAS DEL PRODUCTO Y MODELO DE DESARROLLO	43
TABLA 5 FUNDAMENTOS DE ELECCIÓN DEL SISTEMA	70
TABLA 6 RANGOS DE SEVERIDAD. ELABORACIÓN PROPIA.....	72
TABLA 7 CUESTIONARIO DE EVALUACIÓN HEURÍSTICA.....	75
TABLA 8 PREGUNTAS HEURÍSTICAS	78
TABLA 9 FRECUENCIAS PARA PREGUNTA 1 DEL CUESTIONARIO	81
TABLA 10 FRECUENCIAS PARA PREGUNTA 2 DEL CUESTIONARIO	82
TABLA 11 FRECUENCIAS PARA PREGUNTA 3 DEL CUESTIONARIO	83
TABLA 12 FRECUENCIAS PARA PREGUNTA4 DEL CUESTIONARIO	84
TABLA 13 FRECUENCIAS PARA PREGUNTA4 DEL CUESTIONARIO	85
TABLA 14 VISIBILIDAD EN EL CONTEXTO	86

Índice de gráficas

GRÁFICA 1 FRECUENCIAS PARA PREGUNTA 1 DEL CUESTIONARIO	81
GRÁFICA 2 FRECUENCIAS PARA PREGUNTA 2 DEL CUESTIONARIO	82
GRÁFICA 3 FRECUENCIAS PARA PREGUNTA 3 DEL CUESTIONARIO	83
GRÁFICA 4 FRECUENCIAS PARA PREGUNTA 4 DEL CUESTIONARIO	84
GRÁFICA 5 FRECUENCIAS PARA PREGUNTA 5 DEL CUESTIONARIO.....	85
GRÁFICA 6 FRECUENCIAS PARA LAS PREGUNTAS DE VISIBILIDAD EN EL CONTEXTO DEL CUESTIONARIO	86
GRÁFICA 7 DE COINCIDENCIA ENTRE EL SISTEMA Y EL MUNDO REAL DEL CUESTIONARIO	87
GRÁFICA 8 DE LIBERTAD Y CONTROL POR PARTE DEL USUARIO DEL CUESTIONARIO	88
GRÁFICA 9 DE CONSISTENCIA Y ESTÁNDARES DEL CUESTIONARIO	89
GRÁFICA 10 DE PREVENCIÓN DE ERRORES	90
GRÁFICA 11 DE RECONOCER ES MEJOR QUE RECORDAR	91

Introducción

Los sistemas de información deben cumplir con el objetivo para el cual fueron creados, como primer punto, pero también deben orientar acerca de cómo ser usados, facilitar su aprendizaje, además de buen desempeño, sin complicaciones de redundante funcionalidad. Para ello el sistema debe incluir características de usabilidad, que permite identificar los elementos anteriores y sus niveles de compromiso con el cumplimiento de los requisitos de efectividad, eficiencia y satisfacción para un contexto de uso determinado; evaluar no solo es útil para calificar de bueno o malo un sistema, sino también para reflexionar y corregir problemas, y por ende perfeccionar procesos y resultados.

Esta investigación presenta una propuesta metodológica que permita llevar a cabo una evaluación de la usabilidad, sobre sistemas creados por organizaciones no especialistas en el desarrollo de software y que tienen la necesidad de automatizar sus procesos.

Para el objetivo se aborda estado del arte de la usabilidad, con el fin de conocer lo que se ha hecho en el tema, hasta ahora. Posteriormente el tema de Calidad del software toma importancia como un proceso de mejora continua dentro de las empresas. Es indudable que las métricas del software no se pueden eludir ya que presentan los fundamentos de cualquier decisión en la gestión de sistemas de información.

Se culmina con un estudio de la usabilidad, Para realizar la evaluación de la usabilidad, existe una gran trayectoria. En esencia se puede partir de tres estilos: evaluaciones altamente subjetivas, evaluaciones objetivas y evaluaciones que combinen las dos. La primera de ellas, determinar si una aplicación es o no Usable se encuentra, típicamente, por la aplicación de cuestionarios de actitud del usuario frente al computador. Si bien, uno de los elementos centrales de la usabilidad podría ser la satisfacción, es una visión incompleta y poco predecible como medida de esta característica. Existe el otro caso, métodos donde la evaluación gira en torno a heurísticas o reglas universales de diseño con lo cual es un experto el que asume la responsabilidad de dar su concepto sobre la usabilidad del sistema.

La propuesta metodológica pretende ser una guía para que éste tipo de organizaciones puedan realizar su evaluación, muchas evaluaciones de la usabilidad, requieren de planeación y documentación, es por ello que se desiste de su aplicación. La propuesta va acompañada de un sistema CAME para su fácil aplicación, el cual contiene la guía completa de la evaluación y despliegue de histogramas que den fundamento al nivel de usabilidad del sistema.

Propósito y Organización

Planteamiento del problema

En diversos entornos industriales y académicos, a nivel internacional, la calidad del software ha sido evaluada mediante distintos estudios analíticos. De dichos entornos, la evaluación de la calidad del software ha evolucionado, ahora se tienen modelos formales estadísticos que se basan en métricas como fundamento para el aseguramiento, control y evaluación de la calidad de un producto o proceso de software. Grandes compañías, fundamentan su marco de producción de software con éste enfoque estadístico, lo cual las ha convertido en pioneras de ésta área.

Debido a la importancia que la calidad de software ha despertado en los últimos años GRIHO hace referencia a los atributos más importantes de la Calidad del software de manera resumida en la figura 1:

- Funcionalidad
- Fiabilidad
- usabilidad
- Eficiencia
- Portabilidad
- Mantenibilidad

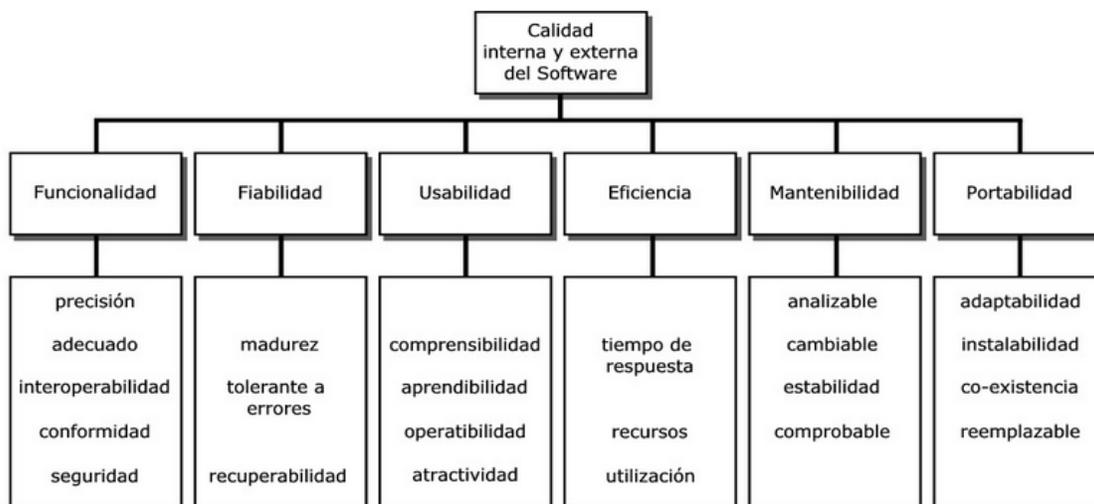


Figura 1 Atributos de la Calidad del Software (GRIHO, 2009)

Para conocer el grado en el que un producto software tiene estos criterios es necesario llevar a cabo un proceso de medición sistemático que arroje métricas

para dar indicio del hecho. Para producir un producto software de calidad el equipo de desarrollo deberá llevar una administración y control de que lo producido es realmente lo correcto, debe evaluar su trabajo mediante métricas del software adecuadas. Se han estudiado y analizado métricas generales y específicas, todas requieren de una recolección de datos para poder emitir un número que deberá ser interpretado y utilizado para la toma de decisiones en favor de la calidad del software, estos datos requieren análisis, depuración, recolección y manipulación para su interpretación (GRIHO, 2009).

Los costos de una evaluación del software incluyendo revisiones, inspecciones y pruebas suelen ser muy elevados si no se realiza de manera eficiente. Las empresas, principalmente fábricas de software optan por utilizar herramientas de automatización que permitan recolectar datos para la generación de métricas; las organizaciones, que hacen software propio, lo que menos desean es contar sus variables o líneas de código. Si pudieran los diseñadores, codificadores, administradores de base de datos y demás involucrados calificar de una manera automatizada su trabajo con respecto a un producto software, entonces se podría corregir ó prevenir algún error, ó simplemente mejorar.

De acuerdo al estudio realizado por (Gutiérrez Gasca, Gutiérrez Tornés, Pérez, & Márquez López, 2008) se concluyó que de las empresas que construyen software, (independiente al ramo que se dedican) un 35% tiene de 3 a 5 empleados, un 56% de las empresas no realiza documentación de calidad, 64% de las empresas sí mide la calidad del producto final. De dicho porcentaje, el 67% realiza principalmente pruebas unitarias y de funcionalidad, un 33% comenta que la miden a través de una encuesta de satisfacción del cliente que se realiza posteriormente a la entrega del producto. Con los resultados obtenidos, se concluye observar que más de la mitad de las empresas que participaron en la encuesta (64%) realiza la evaluación de su producto con base al apego a los requerimientos del cliente y no desde el punto de vista del producto mismo, donde la calidad está determinada por el cumplimiento de las siguientes características: confiabilidad, mantenibilidad, eficiencia, portabilidad y usabilidad.

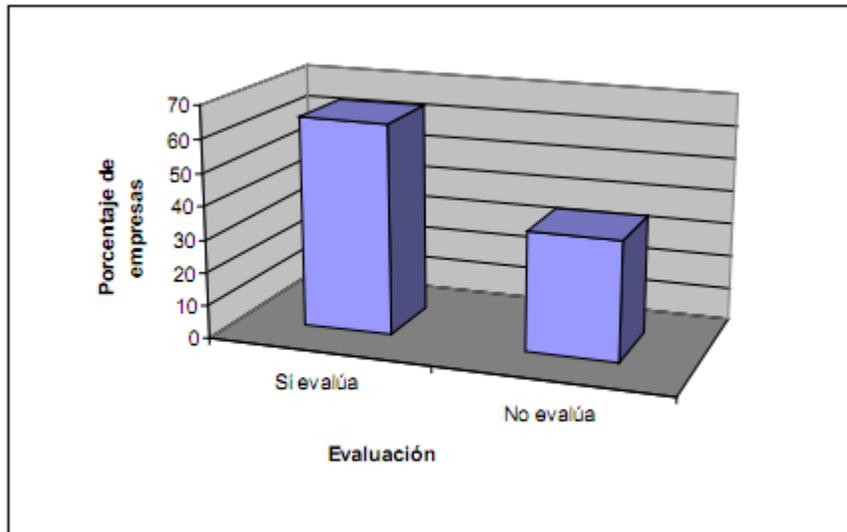


Figura 2 Evaluación del Producto Final (Gutiérrez Gasca, Gutiérrez Tornés, Pérez, & Márquez López, 2008)

En la figura 2 se puede observar que 64% de las empresas sí mide la calidad del producto final. De acuerdo a los resultados obtenidos, más de la mitad de las empresas que participaron en la encuesta (64%) realiza la evaluación de su producto utilizando el enfoque del productor, es decir, la calidad está determinada por el apego a los requerimientos del cliente (en su mayoría funcionales) y no desde el punto de vista del producto, donde la calidad está determinada por el cumplimiento de las siguientes características: confiabilidad, mantenibilidad, eficiencia, portabilidad y usabilidad. La industria se ha centrado casi por completo en la aplicación de los modelos de calidad del proceso (Chrissis, Konrad, & Shrum, 2006) dejando en un menor plano a los modelos de producto ya terminado.

Las interrogantes que surgen después de la explicación y que se deberán responder en ésta investigación, son las siguientes:

- ¿Es posible medir la usabilidad del producto software que desarrollan organizaciones no dedicadas a la fabricación de software, sin tener profesionales de la Ingeniería de la usabilidad?
- ¿Cómo se puede medir la usabilidad del producto software que producen organizaciones no dedicadas a la fabricación de software con métricas asistidas por computadora?
- ¿Cuáles son los criterios apropiados para la evaluación de la usabilidad de productos software?
- ¿En qué medida los resultados de la evaluación heurística por expertos y la evaluación de la encuesta entre usuarios, corresponden?

Propósito de la investigación

Existen evidencias de que las empresas mexicanas tienen un conocimiento reducido de los modelos de calidad de software, desconocen los métodos específicos para evaluar la calidad de su producto y, finalmente, expresan la necesidad de contar con modelos integrales que valoren la calidad del proceso y del producto (Gutiérrez Gasca, Gutiérrez Tornés, Pérez, & Márquez López, 2008).

Se requiere especificaciones detalladas para realizar el estudio de los atributos de calidad, lo cual puede ser muy extenso, por lo que la presente investigación se centrará en la medición de la usabilidad en el contexto de las áreas de desarrollo de las organizaciones. El tipo de investigación a realizar será mixto se utilizarán datos cuantitativos y cualitativos para fundamentar los resultados.

El elemento sustancial será especificar un marco de trabajo para la medición de la usabilidad, característica importante de calidad del producto software, así como determinar cuáles son los parámetros de evaluación adecuados que permitan implementar métricas de usabilidad en empresas construyen su propio software en México. Se hace el diseño de un sistema CAME (Métricas asistida por Computadora) para la administrar el proceso de medición del producto software, dentro del contexto especificado.

Objetivo General

Crear un modelo de medición de la usabilidad en productos software desarrollados en empresas que construyen su propio software México, con un sistema de métricas asistido por computadora.

Objetivos particulares

- Crear un modelo para medir la usabilidad del producto software.
- Desarrollar un software de métricas asistidas por computadora que pueda medir la usabilidad.

Justificación de la investigación

La presente investigación proporciona una puerta de entrada al área de las mediciones de software a nivel nacional, aunque a nivel internacional éste ha sido un tema de debate y de creación de organizaciones, foros, congresos y demás; aquí en México, en el ámbito académico aún se realizan estudios para convencer acerca de la pertinencia de impartir o no una asignatura cuyo principal objetivo sea Métricas del Software. No se puede hablar mucho de la aplicación de Métricas

aplicadas en fábricas de software nacionales, aún continúan con los métodos tradicionales de desarrollo de software, con el fundamento de que las revisiones de la calidad son caras, consumen tiempo y retrasan la entrega del software. Sería posible acelerar el proceso de revisión utilizando herramientas que hicieran valoraciones automáticas de la calidad del software.

La organización de la investigación podrá apoyar a profesionistas del área que deseen introducirse en el tema de métricas, sin perderse en el pantano de información que se ha generado de ellas.

La investigación puede ser utilizada por una gran cantidad de empresas que fabriquen software y requieran medir el grado de usabilidad, sin tener herramientas de apoyo sofisticadas que involucren tiempos de implementación que afecten el tiempo de desarrollo.

Los beneficios de la usabilidad son amplios y tienen impacto tanto desde el punto de vista de la imagen del software como desde el punto de vista económico. Sin embargo, es importante tener en cuenta que los beneficios llegan una vez que el proyecto está en el aire, por lo que la única garantía para generar software usables es tomar en cuenta deliberadamente la usabilidad desde el primer día del proyecto.

Una observación relevante es que desarrollar software usable no es a priori ni más caro ni más barato que desarrollar software no-usable. La cantidad de sitios con abultado presupuesto, alto esfuerzo de diseño gráfico e interfaces de pésima usabilidad así lo prueban

Medir la usabilidad en todo el ciclo de vida del software y generar un sistema CAME requiere de implementar un marco de trabajo incluyendo paradigmas de usabilidad entre las fases del ciclo de vida. El presente trabajo pretende usar métodos de evaluación en el producto software implantado. Los beneficios de utilizar los resultados de ésta investigación redundan en utilizar menor esfuerzo para la aceptación del producto software, dada la definición de usabilidad hecha por (ISO, 1998) a la que hace referencia (Piattini Velthuis, García Rubio, & Caballero Muñoz, 2006) determina cuán fácil un sistema software es de aprender y usar, cuánto de productivos son los usuarios que trabajan con ese sistema y cuánta ayuda necesitarán.

El uso de un método de evaluación del software puede determinar el grado de éxito del software con el usuario, de otra manera acelerar una fase de mantenimiento que considere los resultados de la evaluación.

Hipótesis

A través de una aplicación de software se puede medir la usabilidad de los productos software que producen empresas pequeñas y medianas de México.

1 Estado del Arte

La usabilidad es un término subjetivo y es una característica de la calidad de los sistemas, de acuerdo a organismos internacionales de estándares. Han predominado las propuestas de usabilidad con el método de Diseño Centrado en el Usuario (DCU) que utiliza usuarios representativos. Se presentan algunos trabajos realizados.

1.1 Integración de la Ingeniería de Software y la usabilidad

El Modelo de Proceso de la Ingeniería de la usabilidad y accesibilidad propone un enfoque donde se aplican aspectos del DCU a las fases de desarrollo de software, con el fin de mejorar los productos a través de flexibilidad, entendimiento y fácil manejo por los usuarios pero desde las etapas de desarrollo de software y no desde el producto final como lo utilizan las fábricas de software; de esta manera el modelo agrega 3 componentes esenciales:

- El usuario que interactúa en cada etapa de desarrollo.
- El prototipo.
- Evaluación del producto.

A pesar de ser un modelo que toma en cuenta aspectos de desarrollo de software y de diseño centrado en el usuario, su enfoque radica en la utilización de la usabilidad y accesibilidad dentro de las fases del proceso de desarrollo de software pero sin alterarlas. Este modelo relaciona las fases de desarrollo de software con los principios básicos de la ingeniería de la usabilidad.

La figura refleja muy claramente, con una codificación en colores, estos tres conceptos a modo de tres pilares básicos:

- La Ingeniería del Software, en el formato clásico de ciclo de vida en cascada iterativo o evolutivo.
- El prototipado, como metodología que engloba técnicas que permitirán la posterior fase de evaluación.
- La evaluación que engloba y clasifica a los métodos de evaluación existentes.

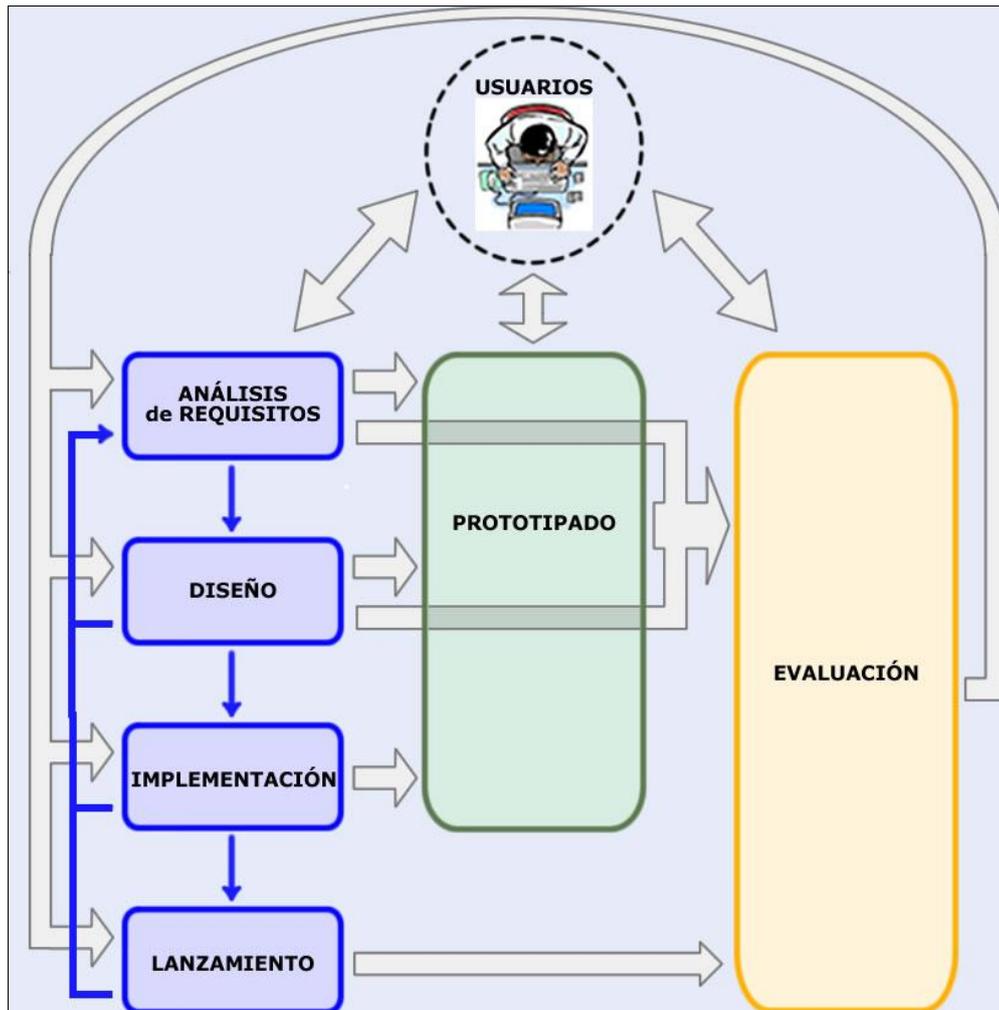


Figura 3 Esquema o Modelo del Proceso (MPIu+a, 2005)

Para este modelo es importante el usuario, de tal forma que realmente el software tenga la característica de amigable, así como también cualquier otro implicado en el sistema software aún cuando no sea usuario final. ver figura 3. Como se observa en la figura el usuario es el centro de desarrollo, específicamente en las fases que interviene. El modelo es iterativo como lo lleva la ingeniería de software en la figura está representado en la flecha que retorna a la fase desde la fase de lanzamiento a la fase análisis de requisitos. El prototipado se lleva a cabo en las tres primeras fases en comunicación continua con el usuario, previo análisis de requisitos con el usuario. La evaluación es a partir del prototipo y el Lanzamiento, de tal manera que puede llevarse una retroalimentación de sus resultados hacia las fases de desarrollo.

Al aplicar el modelo se debe realizar una interfaz sencilla, utilizando una metodología sencilla y comunicación con el usuario, utilizando métodos descriptivos en lenguaje natural junto con herramientas de uso habitual, facilita el proceso comunicativo entre las personas que intervienen en el desarrollo (Granollers & Lorés, MODELO de PROCESO de la Ingeniería de la usabilidad. Integración de la Ingeniería de Software y la usabilidad, 2008).

1.2 Patrones de usabilidad: Mejora de la usabilidad del Software desde la Arquitectura

Es una aproximación para mejorar la usabilidad de un software aplicando un proceso específico de diseño para usabilidad. Se relaciona con el desarrollo de técnicas y procedimientos para mejorar la usabilidad, desde momentos tempranos del desarrollo de software. Esta aproximación difiere de la idea tradicional de medir y mejorar la usabilidad una vez finalizado el sistema, al igual que el anterior modelo. Aunque el esquema está enfocado al desarrollo de software mezclado con usabilidad parece no tomar en cuenta criterios de accesibilidad que también son muy importantes para el desarrollo de un producto.

Surge del proyecto STATUS (SoFTware Architectures That support USability), cuyo objetivo es el desarrollo de técnicas y procedimientos a incorporar durante el diseño de un sistema software con el fin de conseguir mejoras en la usabilidad del sistema a construir, se debe observar la siguiente figura para identificar el modelo propuesto.

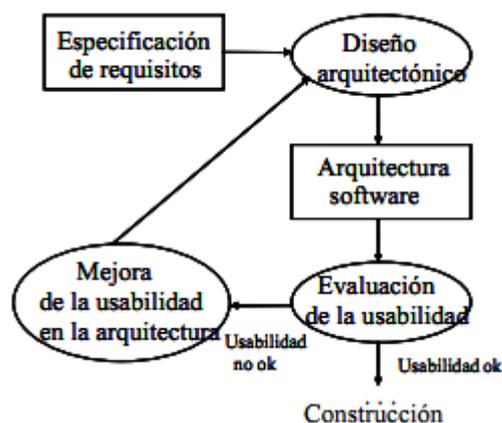


Figura 4 Paradigma de Patrones de usabilidad (Sánchez & Moreno, 2003, pág. 120)

El proceso de diseño comienza con la construcción de un modelo de la arquitectura software a partir de un conjunto de requisitos funcionales, , la mayoría de requisitos no funcionales son por lo general estudiados más adelante. Así, el diseño preliminar obtenido es evaluado con respecto a ciertos atributos de calidad, concretamente la usabilidad en este caso. Si el valor de usabilidad obtenido es adecuado con respecto a lo solicitado en los requisitos el diseño finalizaría y se pasaría a las etapas de desarrollo posteriores. Si la usabilidad necesita mejorarse, se entraría en un ciclo de mejora en la arquitectura software y nuevas evaluaciones, hasta que el nivel de usabilidad sea el adecuado. Este proceso no implica que no se evalúe la usabilidad del sistema una vez terminado, simplemente se pretende adelantar el ciclo de evaluación/mejora de la usabilidad con el fin de ahorrar esfuerzos en el proceso de desarrollo.

El modelo se centra en realizar mejoras en la usabilidad en la fase de diseño, específicamente al definir la arquitectura, para ello se descompone en 3 niveles de abstracción: atributos de usabilidad, propiedades de usabilidad y patrones de usabilidad. Los atributos clásicos son: aprendizaje, eficiencia, fiabilidad y satisfacción; estos conceptos identifican las características de la usabilidad. Las propiedades se utilizan para identificar mecanismos incorporables en el diseño y los mecanismos son llamados patrones. En la figura 5 se da la relación entre estos conceptos.

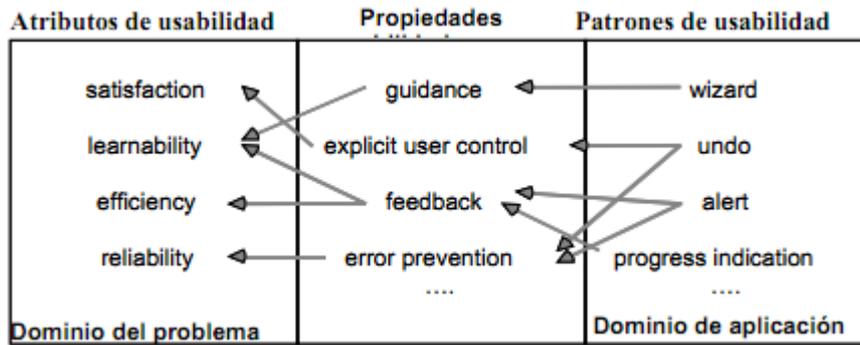


Figura 5 Relaciones entre atributos, propiedades y patrones. (Sánchez & Moreno, 2003)

En el proyecto STATUS, además de la idea del patrón de usabilidad, también se usa el concepto de patrón arquitectónico (descripción de las clases y de los objetos que trabajan conjuntamente para resolver un problema concreto). Dado que el patrón de usabilidad se ha definido como un mecanismo a ser incorporado en el diseño de una arquitectura software a fin de abordar una propiedad de usabilidad concreta, un patrón arquitectónico determina cómo se incorpora este patrón de usabilidad en una arquitectura software; así se agrega una nueva columna a la tabla, con el fin de crear la relación entre atributos, propiedades, patrones de usabilidad y patrones arquitectónicos. Ver figura 6.

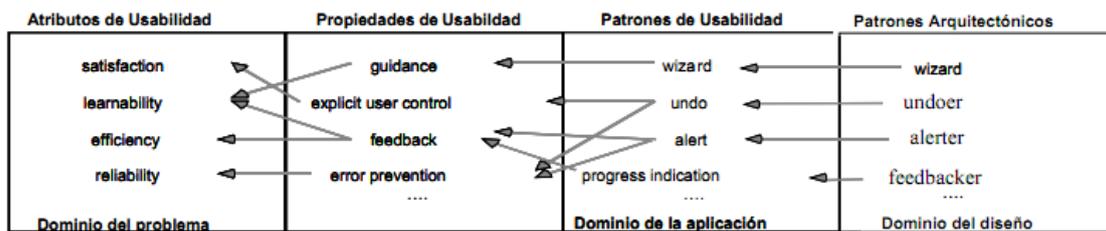


Figura 6 Relaciones entre Atributos, Propiedades, Patrones de usabilidad y Patrones Arquitectónicos (Sánchez & Moreno, 2003)

Cabe mencionar que los patrones arquitectónicos se establecen utilizando dos pasos: Mediante la inducción se establecen patrones. Primero se hacen modelos de diseño sin patrones de usabilidad, después se pide se incluya la funcionalidad correspondiente a los patrones de usabilidad, a partir de ahí utilizando procesos de abstracción se diseñan patrones de usabilidad.

Se detallan los elementos que deben tener los patrones arquitectónicos:

- Nombre del patrón arquitectónico. Los patrones deben tener nombres sugestivos que den una idea del problema que abordan y de su solución.

- Problema. Describe cuándo aplicar el patrón y en qué contexto. En el caso de los patrones arquitectónicos, el problema se refiere a un patrón de usabilidad concreto que ha de ser materializado.

- Solución. Describe los elementos que conforman la arquitectura, sus relaciones, sus responsabilidades, etc. la solución para un patrón específico se hará explícita a partir de una.

- Representación gráfica. Una figura que representa los componentes de la arquitectura y sus interacciones. Las flechas numeradas entre los distintos componentes representarán las interacciones. Las flechas con las líneas ininterrumpidas indican el flujo de datos, mientras que las líneas interrumpidas representan el flujo de control entre los componentes.

- Participantes. Una descripción de los componentes que participan en la solución propuesta y las iteraciones (representadas por flechas) para determinar cómo deben asumir sus responsabilidades.

- Ventajas para la usabilidad. Descripción de los aspectos de usabilidad (propiedades de usabilidad) que pueden mejorarse al incluir el patrón apropiado.

- Beneficios de usabilidad. Una argumentación razonada acerca del impacto que tiene la aplicación del patrón en la usabilidad.

- Consecuencias. Impacto del patrón en otros atributos de calidad, como la flexibilidad, la portabilidad, la mantenibilidad, etc. Al igual que la característica anterior, ésta se rellenará con los resultados de la experiencia empírica.

- Patrones relacionados. Cuáles son los patrones arquitectónicos más íntimamente relacionados con éste y cuáles son las diferencias entre ellos.

Estos patrones se deben implementar desde la fase de Diseño para y los patrones a utilizar dependerán del tipo de sistema (Sánchez & Moreno, 2003).

Propiedad de Usabilidad	Patrón de Usabilidad
NATURAL MAPPING	
CONSISTENCY (functional, interface, evolutionary)	
ACCESSIBILITY (internationalisation)	Different languages
CONSISTENCY, ACCSESIBILITY (multichannel, disabilities)	Different access methods
FEEDBACK	Alert
ERROR MANAGEMENT, FEEDBACK	Status indication
EXPLICIT USER CONTROL, ADAPTABILITY (user expertise)	Shortcuts (key and tasks)
ERROR MANAGEMENT (error prevention)	Form/field validation
ERROR MANAGEMENT (error correction),	Undo
GUIDANCE, ERROR MANAGEMENT	Context-sensitive help
GUIDANCE, ERROR MANAGEMENT	Wizard
GUIDANCE, ERROR MANAGEMENT	Standard help
GUIDANCE,	Tour
ERROR MANAGEMENT	
MINIMISE COGNITIVE LOAD,	
ADAPTABILITY, ERROR MANAGEMENT (error prevention)	Workflow model
ERROR MANAGEMENT (error correction)	History logging
GUIDANCE,	
ERROR MANAGEMENT (error prevention)	Provision of views
ADAPTABILITY (user preferences)	User profile
ERROR MANAGEMENT,	
EXPLICIT USER CONTROL	Cancel
EXPLICIT USER CONTROL	Multi-tasking
MINIMISE COGNITIVE LOAD	
ERROR MANAGEMENT (error prevention)	Commands aggregation
EXPLICIT USER CONTROL	Action f or multiple objects
MINIMISE COGNITIVE LOAD, ERROR MANAGEMENT (error prevention)	Reuse information

Tabla 1 Lista de Patrones de usabilidad (Sánchez & Moreno, 2003)

1.3 Ciclo de Vida en Estrella

Ferré hace referencia a la propuesta de Hix y Hartson, se trata de un ciclo de vida centrado en la evaluación de la usabilidad, los resultados de cada actividad se evalúan antes de continuar con la siguiente. No especifican un orden entre las actividades, éste enfoque tiene iteraciones cortas. Esta aproximación trata las técnicas de especificación de usabilidad y su evaluación formativa a través de test de usabilidad con usuarios reales, aunque no especifica los procesos de análisis de sistemas que son comunes al desarrollo de la interacción con el usuario y al desarrollo de la parte interna del sistema.

En la figura 7 se puede observar las actividades del ciclo de estrella, en donde las actividades en el sentido del reloj, desde análisis de sistemas hasta prototipado, son las que se relacionadas con HCI mientras que las otras son las relacionadas con la construcción del sistema.

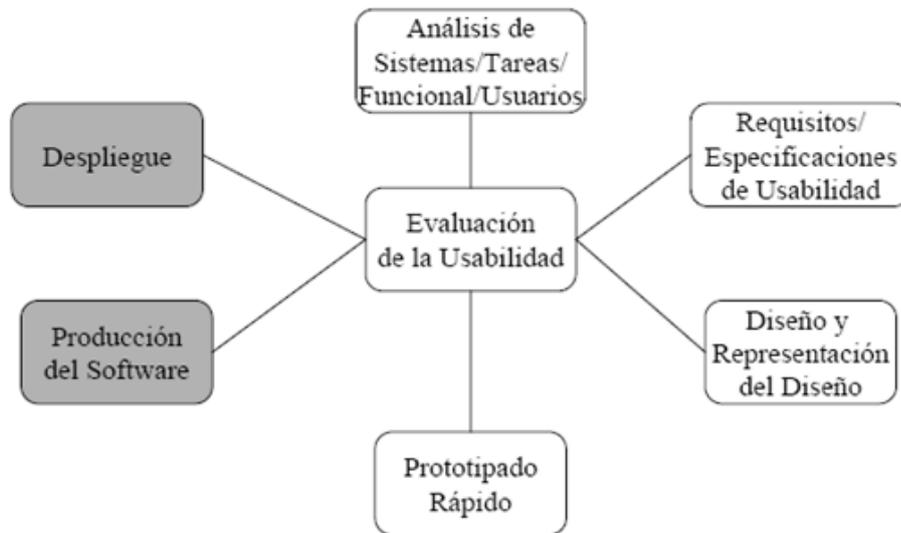


Figura 7 Ciclo de Vida en Estrella (Ferré, 2005)

Las Tareas Relativas al análisis de sistemas incluyen: Análisis de la necesidad de usuario y de tareas y asignación de funcionalidad. En cuanto al diseño, se detallan guías de diseño y los estilos de interacción, en cuanto al diseño como producto. Se trata de la aproximación al desarrollo de sistemas interactivos que mejor transmite el enfoque iterativo que debe ser inherente a todo el desarrollo. Esta visión del desarrollo, en el que los tipos de actividades y la secuencia en la que se realizan están completamente separados. Esta aproximación trata muy bien las técnicas de especificación de usabilidad y la evaluación formativa de usabilidad a través de test de usabilidad con usuarios reales (Ferré, 2005).

2 Calidad de software

2.1 Definición

El éxito de cualquier producto se basa en la pertenencia de un nivel adecuado de calidad. Para que el proyecto, en el que surge el producto, sea rentable para la organización, debe realizarse con la máxima eficiencia y productividad. Antes de generar procesos eficientes y productivos, se analiza el concepto de calidad.

De acuerdo a la Real Academia Española (Real Academia Española, 2001) calidad es:

- Propiedad o conjunto de propiedades inherentes a algo, que permiten juzgar su valor
- Buena calidad, superioridad o excelencia..
- Carácter, genio, índole.
- Condición o requisito que se pone en un contrato.
- Estado de una persona, naturaleza, edad y demás circunstancias y condiciones que se requieren para un cargo o dignidad.
- Nobleza del linaje.
- Importancia o gravedad de algo.

Las organizaciones prefieren tomar la primera definición, en donde se pretende identificar características que superiores a otras, aunque la percepción de superior podría tener tintes subjetivos de acuerdo a las diferentes percepciones con que los humanos observan la realidad. Los llamados Gurús expertos en la definición de Calidad como Deming, Juran, Crosby, Taguchi, Ishikawa y Geingenbaum (Piattini, García, Caballero, 2007), cuyas definiciones y propuestas de gestión de calidad, ver Tabla 2.1, sólo permiten concluir que no hay una definición única y absoluta, sino que se trata de interpretaciones de acuerdo a diferentes enfoques. Piattini hace referencia a (Garvín, 1984) especificando las 5 vistas de la calidad Tabla 1.1.

- Vista Trascendental, no se define pero se percibe cuando hace falta.
- Vista de Usuario, es adecuación al propósito.
- Vista de Fabricante, es conformidad con las especificaciones.
- Vista del Producto, es unión a las características inherentes del producto.

PHILIP CROSBY	KAORU ISHIKAWA	GENICHI TAGUCHI	JOSEPH M. MURÁN	WILLIAM EDWARD DEMING
Calidad es el cumplimiento de los requerimientos, donde el sistema es la prevención, es estándar cero defectos y la medida es el precio del incumplimiento.	Calidad es cuando se logra un producto sea útil, económico y satisfactorio para el consumidor.	La mínima pérdida ocasionada a la sociedad desde el envío del producto al cliente hasta su uso total.	La calidad es estar en forma para el uso, desde los puntos de vista estructurales, sensoriales, orientados en el tiempo, comerciales y éticos en base a parámetros de calidad de diseño, de cumplimiento, de seguridad y servicio.	Calidad se transforma en un proceso de mejora continua, se analiza cada parte del proceso para conocerlo y evitar futuros errores, logrados los objetivos del primer esfuerzo continuar estableciendo, y no dejar de seguir el Proceso

Tabla 2 Concepto de Calidad (autoría propia)

El término calidad tiene una historia que llega a la presente época con las normas ISO 9000, en donde se hace referencia al término requisito “necesidad o expectativa establecida generalmente implícita u obligatoria”, satisfacción del cliente “grado en que se han cumplido sus requisitos” y capacidad “aptitud para realizar un producto que cumple los requisitos para el mismo”, como se puede apreciar se requiere considerar al cliente o usuario del producto para analizar el nivel de cumplimiento y satisfacción por parte del cliente o usuario. Dada la diversidad de usuarios que se pueden tener, se requiere de una gran gama de herramientas analíticas para determinar el porcentaje, a continuación se hace una referencia general a éstas.

2.2 Herramientas de calidad

Son todas las gráficas requeridas para explicar y detallar procesos (Diagramas de Flujo), para identificar, explorar y mostrar todas las posibles causas de un problema específico (Diagrama de causa-efecto), para establecer jerarquías de los problemas o causas que lo generan (Diagrama de Pareto), lista de verificación, chequeo o cotejo para identificar y analizar tanto los problemas como sus causas, representaciones gráficas utilizadas para determinar desde un punto de vista estadístico si un proceso o no está bajo control (Diagrama de control, en cualquiera de sus modalidades), los histogramas que sirven para ilustrar la frecuencia de eventos relacionados entre sí y por último diagramas que estudian la posible relación entre dos variables objeto de estudio de un control de calidad.

2.2.1 Herramientas de Gestión

En orden de jerárquico se hace referencia a los diagramas que más se utilizan para la buena gestión. Los diagramas de afinidad sirven para organizar un gran número de ideas en categorías relacionadas o afines, herramientas

utilizadas para identificar las causas más significativas de un problema (Diagrama de relaciones), diagramas para representar las relaciones entre varios factores (Diagrama de Matriz) y también se pueden utilizar representaciones jerárquicas de niveles de complejidad de un proceso o producto (Diagrama de Árbol).

2.2.2 Herramientas de Creatividad

Se refiere a la lluvia de ideas o analogías o mapas conceptuales, se detalla con más profundidad la lluvia de ideas. Se utiliza para obtener el mayor número de ideas o soluciones en el menos tiempo posible, para seleccionar posteriormente las más indicadas, se deben conocer los objetivos y pueden participar todos los miembros de un equipo de trabajo o bien participan con ideas según vayan surgiendo. Se tienen tres fases en la lluvia de ideas.

- Definición y Comunicación del asunto a tratar
- Exposición de ideas
- Selección de ideas

2.2.3 Herramientas Estadísticas

Se utilizan para identificar la capacidad de un proceso, que es el grado de aptitud que tiene para cumplir con las especificaciones técnicas deseadas. Si la capacidad es alta se dice que es capaz, cuando se mantiene estable a través del tiempo entonces está bajo control. Se tienen las siguientes herramientas:

- Histogramas
- Gráficos de Control
- Gráficos de Probabilidad
- Estudios de índices de Capacidad

2.2.4 Herramientas de Diseño

Se utilizan principalmente cuando se diseñan o planifican nuevos productos o procesos. El Diagrama de Despliegue de la Función de Calidad (QFD) se utiliza para planificar nuevos productos y servicios o realizar mejoras en los ya existentes, se puede ver como un sistema estructurado que facilita el medio para identificar necesidades y expectativas de los clientes y traducirlas al lenguaje de la organización, esto es, a requerimientos de calidad internos, desplegándolas en la etapa de planificación con la participación de todas las funciones que intervienen en el diseño y desarrollo del producto o servicio.

Tiene dos propósitos:

- Desplegar la calidad del producto o servicio. Es decir, el diseño del servicio o producto sobre la base de las necesidades y requerimientos de los clientes.
- Desplegar la función de calidad en todas las actividades y funciones de la organización.

Consta de las siguientes fases:

- Identificar y jerarquizar a los clientes.
- Identificación de las expectativas del cliente.
- Conversión de la información en descripciones verbales específicas.
- Elaboración y administración de la encuesta a clientes.
- Despliegue de la calidad demandada.
- Despliegue de las características de Calidad

2.2.5 Herramientas de Medición

Se utilizan para identificar problemas y sus posibles costos (Coste de calidad COQ), también se utilizan para comparar las mejores prácticas de las organizaciones (Benchmarking) y también para determinar la naturaleza de los procesos (Encuesta)

La lista anterior es utilizada en todos los proyectos para la mejora o creación de un producto nuevo o servicio y es un soporte vital en las diferentes ingenierías, hasta antes de los años setentas, el software era construido sin muchas bases de gestión y mucho menos de gestión de la Calidad, lo que tuvo como resultado tener software que no cumplía con los requisitos del cliente, que era entregado en fechas atrasadas, con costos muy altos y sin un plan de mantenimiento viable por lo que sobrevino la crisis del software; término que se acuñó por primera vez en la conferencia de la OTAN. Después ésta se especifico el término Ingeniería del Software, que como cualquier otra ingeniería especializada requiere de llevar a cabo procesos de Gestión para la creación de Productos de Calidad.

2.3 Calidad del Software

El interés de la investigación es exclusivamente el producto software, por lo que se hace mención a la calidad del software y se discutirá el tema. En (Weitzenfeld, 2004, p. 56) se hace referencia al concepto dado por el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE) según el cual Calidad del Software es “el grado en que un sistema, componente o proceso cumple con los requerimientos especificados y las necesidades del cliente o usuario”. También se menciona la definición dada por la norma ISO-9000 de la organización Internacional para la estandarización (ISO) “es el grado (pobre,

bueno o excelente) en que un conjunto de características inherentes del software cumplen con los requisitos del sistema”.

A la hora de definir la calidad del software se debe diferenciar entre la calidad del producto software y la calidad del proceso de desarrollo de éste. No obstante, las metas que se establezcan para la calidad del producto van a determinar los objetivos a establecer de calidad del proceso de desarrollo, ya que la calidad del primero va a depender, entre otros aspectos, de ésta. Sin un buen proceso de desarrollo es casi imposible obtener un buen producto. Este proceso constituye el objeto del presente trabajo. La calidad del software está directamente relacionada con su proceso de desarrollo. Se considera que un proceso bien conocido y ampliamente utilizado, sustentado en medición y predicción de eventos, permite controlar en buena medida la producción de software y, en consecuencia, producir software de calidad.

El éxito de un proyecto de desarrollo de software se basa en la obtención de un nivel adecuado de calidad en el software desarrollado para satisfacer al usuario a la vez que también supone el cumplimiento de plazos y presupuestos para el cliente. Por supuesto, para que el proyecto sea rentable e interesante para la organización de desarrollo, debe realizarse con la máxima eficiencia y productividad. Para poder combinar ambas facetas de manera eficaz, no se debe establecer un proceso de aseguramiento de calidad del software adaptado a las necesidades técnicas del proyecto sino también a la organización de desarrollo logrando minimizar los recursos para la consecución y el control de calidad.

La calidad del producto software se diferencia de la calidad de otros productos de fabricación industrial, el software tiene sus propias características específicas.

- El software es un producto mental, no restringido por las leyes de la Física o por los límites de los procesos de fabricación. Es algo abstracto, un intangible.
- Se desarrolla, no se fabrica. El costo está fundamentalmente en el proceso de diseño, no en la posterior producción en serie, y los errores se introducen también en el diseño, no en la producción.

Los costos del desarrollo de software se concentran en las tareas de Ingeniería, mientras que en la fabricación clásica los costos se acentúan más en las tareas de producción.

El software no se deteriora con el tiempo. No es susceptible de los efectos del entorno y su curva de fallos es muy diferente de la del hardware. Todos los problemas que surjan durante el mantenimiento estaban allí desde el principio y afectan a todas las copias del mismo; no se generan nuevos errores.

- Es artesanal en gran medida. El software, en su mayoría, se construye a medida, en vez de ser construido ensamblando componentes existentes y ya probados, lo que dificulta aún más el control de su calidad.

- El mantenimiento del software es mucho más complejo que el mantenimiento del hardware. Cuando un componente del hardware se deteriora se sustituye por una pieza de repuesto, pero cada fallo en el software implica un error en el diseño o en el proceso mediante el cual se tradujo el diseño en código máquina ejecutable.

- Es engañosamente fácil realizar cambios sobre un producto software, pero los efectos de estos cambios se pueden propagar de forma explosiva e incontrolada.

- Como disciplina, el desarrollo de software es aún muy joven, por lo que las técnicas de las que dispone aún no están perfeccionadas.

- El software con errores no se rechaza. Se asume que es inevitable que el software presente algunos errores de poca importancia.

Es un hecho que el software debe cumplir con los requerimientos del cliente, ésta afirmación no es cuestionable; lo que se analiza con más detenimiento es que cumpla con las necesidades del cliente o usuario, para ello se debe realizar la pregunta ¿Cómo identificar las necesidades del usuario?. En el presente y futuros años un sistema software tiene una gran variedad de usuarios, por lo que dar cumplimiento a ésta diversidad de personas se requiere crear procesos detallados que identifiquen necesidades no funcionales modelo (comunes al grupo de personas al que está dirigido el software) para implementar la solución dentro del sistema. ISO menciona el término de características inherentes, lo cual involucra que el software debe llevar implícitas éstas características, o bien que son inseparables a éste, por lo que es cuestionable que algunas características sean consideradas de Calidad y no características implícitas.

2.3.1 Gestión de la Calidad del Software

Según ISO es un conjunto de actividades de la función general de la dirección que determina la calidad, los objetivos y las responsabilidades y se implanta por medios tales como la planificación de la calidad, el control de la calidad, el aseguramiento (garantía) de la calidad y la mejora de la calidad, en el marco del sistema de calidad (ISO, 9000).

La gestión de la calidad del software provee una comprobación independiente de los procesos de desarrollo del software. Los procesos de gestión de calidad comprueban las entregas del proyecto para asegurarse que concuerdan con los estándares y metas organizacionales, el trabajo lo realiza un equipo independiente al de desarrollo, para evitar toma de decisiones por consideraciones de presupuesto o agenda. La calidad puede lograrse definiendo estándares y procedimientos organizacionales de calidad que comprueban si estos estándares son seguidos por el equipo de desarrollo, pero también existen aspectos intangibles en los que aún se investiga como implementarlos bajo estándares.

La gestión de calidad es importante para equipos que desarrollan sistemas grandes y complejos, para sistemas pequeños se debe adoptar

aproximaciones más informales, la comunicación es más directa e informal, lo importante es introducir la cultura de calidad y asegurarse que todo el equipo de desarrollo hagan una aproximación positiva a la calidad del software (Sommerville, 2005). La gestión de calidad implica las siguientes actividades:

- Garantía de la Calidad.
- Planificación de la Calidad.
- Control de Calidad.

2.3.2 Garantía de la Calidad del Software

La garantía de calidad del software, también llamada aseguramiento de calidad, es el conjunto de actividades planificadas y sistemáticas necesarias para aportar la confianza en que el producto software satisfará los requisitos dados de calidad, se diseña para cada aplicación antes de comenzar a desarrollarla y no después. Se ocupa de definir o seleccionar los estándares que deben ser aplicados al proceso de desarrollo o al producto software. Existen dos tipos de estándares:

- Estándares de Producto. Incluyen estándares de documentación y estándares de codificación.
- Estándares de Proceso. Definen los procesos que deben seguirse durante el desarrollo del software

Los estándares de producto se aplican a las salidas del proceso software y los estándares de procesos incluyen actividades de proceso específicas que garantizan que se sigan los estándares de producto. Se han desarrollado estándares nacionales e internacionales para abarcar la terminología, los símbolos de los diagramas, los procedimientos para derivar y redactar los requerimientos de software, los procedimientos de garantía de calidad, y los procesos de verificación y validación del software.

El aseguramiento pretende dar confianza en que el producto tiene calidad y suele llevarse a cabo en:

- Métodos y herramientas de análisis, diseño, programación y prueba.
- Inspecciones técnicas formales en todos los pasos del proceso de desarrollo del software.
- Estrategias de prueba multiescala.
- Control de la documentación del software y de los cambios realizados.
- Procedimientos para ajustarse a los estándares.
- Mecanismos de medida, es decir, extracción de métricas.

- Registro de auditorías y realización de informes.

Las actividades para el aseguramiento de calidad del software son:

- Métricas de software para el control del proyecto.
- Verificación y validación del software a lo largo del ciclo de vida, incluye las pruebas y los procesos de revisión e inspección.
- La gestión de la configuración del software.

2.3.3 Planificación de la Calidad del Software

Es el proceso en el que se desarrolla un plan de calidad para un proyecto, en éste se define la calidad del software deseado y describe como valorar ésta. El plan de calidad selecciona los estándares organizacionales y un proceso de desarrollo particulares. Sommerville hace referencia a (Humphrey, 1989) que sugiere una estructura para un plan de calidad.

- Introducción al producto. Contiene la descripción del producto.
- Planes de producto. Contiene las fechas de terminación y las responsabilidades importantes.
- Descripciones del proceso. Contiene los procesos de desarrollo y de servicio a utilizar para el desarrollo del producto.
- Metas de calidad. Contiene metas y planes de calidad, incluyendo los atributos de calidad importantes del producto.
- Riesgos y gestión de riesgos. Contiene los conflictos clave que podrían afectar la calidad del producto y las acciones para contrarrestar los riesgos.

Los planes difieren del tamaño y tipo de sistema. Existe una amplia variedad de atributos de calidad, no es posible optimizar todos los atributos en el sistema software, el plan de calidad, define los más importantes para el sistema.

2.3.4 Control de la Calidad del Software

Son las técnicas y actividades de carácter operativo, utilizadas para satisfacer los requisitos relativos a la calidad, centradas en dos objetivos fundamentales:

- Mantener bajo control un proceso.
- Eliminar las causas de los defectos en las diferentes fases del ciclo de vida.

En general son las actividades para evaluar la calidad de los productos desarrollados.

El control de calidad implica vigilar el proceso de desarrollo para asegurar que siguen los procedimientos y estándares de garantía de calidad. Existen dos enfoques complementarios que se utilizan para comprobar la calidad de las entregas de un proyecto:

- Revisiones de la calidad donde el software, su documentación y los procesos utilizados en su desarrollo son revisados por un grupo de personas.
- Valoración automática utilizando software que compara los estándares que se aplican a éste proyecto en particular.

2.3.5 Sistema de calidad

Corresponde a la estructura organizativa, procedimientos, procesos y recursos necesarios para implantar la gestión de calidad, debe adecuar a los objetivos de calidad de la empresa. La dirección de la empresa es la responsable de fijar la política de calidad y las decisiones relativas a iniciar, desarrollar, implantar y actualizar el sistema de calidad (Cuevas, 1999). Un sistema de calidad consta de varias partes:

- Documentación. Manual de calidad. Es el documento principal para establecer e implantar un sistema de calidad. Puede haber manuales a nivel de empresa, departamento, producto,
- Parte física. Locales, herramientas ordenadores, etc.
- Aspectos humanos. Formación de personal, Creación y coordinación de equipos de trabajo
- Normativas
- ISO 9000: Gestión y aseguramiento de calidad (conceptos y directrices generales)
- Recomendaciones externas para aseguramiento de la calidad (ISO 9001, ISO 9002, ISO 9003)
- Recomendaciones internas para aseguramiento de la calidad (ISO 9004)
- Software Engineering Institute (SEI) Capability Maturity Model (CMM)

2.3.6 Certificación de la calidad

La calidad de software no se certifica, lo que se certifica son los procedimientos para construir un software de calidad, los procedimientos. Un sistema de certificación de calidad permite una valoración independiente que debe demostrar que la organización es capaz de desarrollar productos y

servicios de calidad. Los pilares básicos de la certificación de calidad son tres (Cuevas, 1999, p. 8):

- Una metodología adecuada
- Un medio de valoración de la metodología

La metodología utilizada y el medio de valoración de la metodología deben estar reconocidos ampliamente por la industria

2.3.7 Modelos y normas de calidad

La normalización consiste en un proceso donde se elaboran guías, normas y convenciones sobre una determinada materia, con el objeto de definir, simplificar y especificar las actividades relacionadas con la materia de que se trate. Un modelo de calidad del software es un conjunto de buenas prácticas para el ciclo de vida del software, enfocado en los procesos de gestión y desarrollo de proyectos. Es importante mencionar que la calidad de software no se certifica, lo que se certifica son los procedimientos para construir un software de calidad, los procedimientos deben ser correctos y estar en función de la normalización. Existen muchos modelos de calidad, se exponen sólo algunos a los cuales hace referencia (Piattini, 2007, pp 49-71). Figura 8.

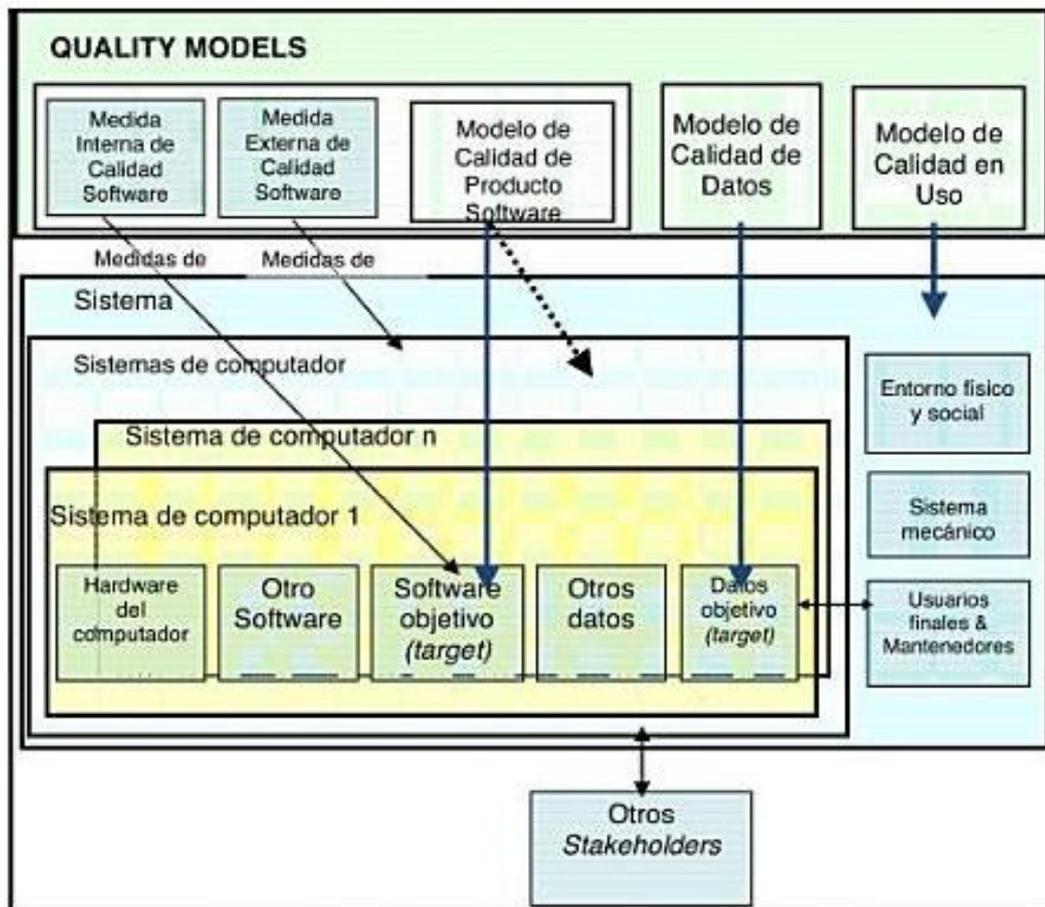


Figura 8 Modelos de Calidad (Carego, Moraga, & Piattini, 2010)

2.3.8 Gestión de la Calidad Total

Requiere un cambio de actitud para satisfacer las necesidades del cliente, esto debe generar una cultura de calidad en todas las áreas y fases de la organización. Se auxilia en el ciclo de Walter: Planear, Hacer, Verificar y Actuar (PDCA por sus siglas en inglés). De la misma forma se apoya en otros principios:

- Compromiso de la alta gestión con todos los empleados
- Reducción de ciclos de producción
- Producción Justo a tiempo
- Reducción de costos
- Implicación y enriquecimiento del puesto de trabajo del personal (empoderamiento).
- Reconocimiento y celebración
- Benchmarking
- Toma de decisiones basada en hechos

2.4 Norma ISO 9001

Son normas de “calidad” y “gestión continua de calidad”, establecidas la organización internacional para la estandarización (ISO) que se pueden aplicar en cualquier tipo de organización o actividad sistemática, que esté orientada a la producción de bienes o servicios. Se componen de estándares y guías relacionados con sistemas de gestión y de herramientas específicas como los métodos de auditoría (el proceso de verificar que los sistemas de gestión cumplen con el estándar). Figura 9.



Figura 9 Principios de la Norma ISO 9001 (BAC International Ban, 2010)

2.4.1 Norma ISO 9001: Fuente

Las organizaciones, de cualquier género, se apoyan en esta norma para fundamentar que tienen capacidad de satisfacer al cliente a través de la aplicación del sistema de la gestión de calidad. Se centra en el un ciclo de mejora continua que considera de manera importante los siguientes aspectos denotados en la figura 10:

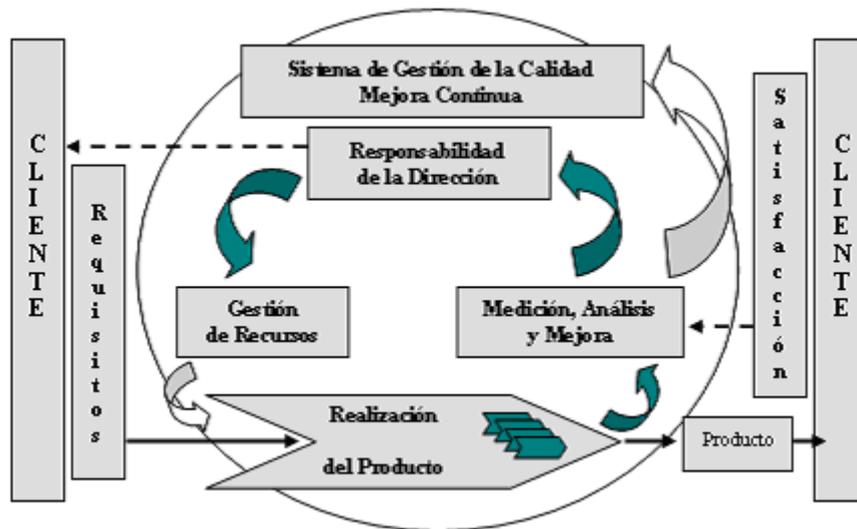


Figura 10 Norma ISO 9001 (GES, 2008)

Sistema de Gestión de Calidad, la norma determina identificar procesos, determinar su secuencia, criterios y métodos necesarios para la eficacia, asegurarse de la disponibilidad de recursos, realizar la verificación de los procesos e implementar acciones de mejora continua. Además de señalar la documentación necesaria que fundamente el sistema de gestión de calidad.

Responsabilidad de la dirección se consideran aspectos importantes para contar con el apoyo del Director para la satisfacción del cliente (compromiso, enfoque, política, responsabilidad, autoridad, comunicación y revisión)

Gestión de recursos, se debe asegurar la obtención total para cumplir con el cliente. la Norma distingue 3 tipos de recursos sobre los cuales se debe actuar: recurso humano, infraestructura y ambiente de trabajo

Fase de realización del producto, aquí se sitúa los requisitos para los procesos que recopilan información, la analizan y que actúan en consecuencia. El objetivo es mejorar continuamente la capacidad de la organización para suministrar productos que cumplan los requisitos. El objetivo declarado en la Norma, es que la organización busque sin descanso la satisfacción del cliente a través del cumplimiento de los requisitos:

- Requisitos generales.
- Seguimiento y medición.

- Control de producto no conforme.
- Análisis de los datos para mejorar el desempeño.
- Mejora

2.5 Modelo EFQM

Norma Europea utilizada para la autoevaluación, el benchmark, para identificación de áreas de mejora, fundamentos de la cultura organizacional y estructuras de sistemas de gestión de la organización.

Estimular y asistir a las organizaciones en toda Europa para participar en las actividades de mejoramiento enfocadas en lograr la excelencia en la satisfacción al cliente, la satisfacción de los empleados, el impacto en la sociedad y en los resultados de negocio.

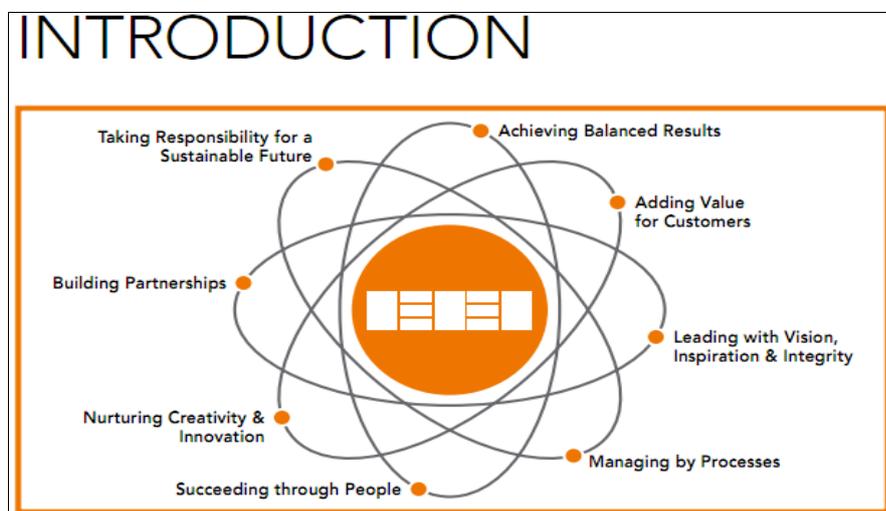


Figura 11 Modelo EFQM 2010 (EFQM, 2008)

EFQM se fundamenta en principios que están orientados en la excelente satisfacción del cliente algunos son: orientación a resultados, orientación al cliente, liderazgo, gestión por procesos y hechos, etc. Ver figura 11.

Los niveles de madurez que tiene este modelo son cinco que se diferencian principalmente por la cantidad de datos utilizados por la evaluación, así como por los resultados de mejora obtenidos.

2.6 CAF Marco común de evaluación

Se fundamenta en EFQM y su propósito es identificar características en organizaciones públicas, ser herramienta para los servidores públicos, ser puente hacia otros modelos y facilitar el benchmark. Ver Figura 12



Figura 12 Modelo CAF 2006 (Peralta & Hidalgo, 2007)

El CAF tiene cuatro propósitos principales (Peralta & Hidalgo, 2007, pág. 9):

- 1. Introducir a la administración pública de forma progresiva, en los principios de la gestión de la calidad total y guiarla por medio del conocimiento y la utilización de la autoevaluación, a partir de la Planificación y el Desarrollo de actividades hasta completar el ciclo “Planificar, Desarrollar, Controlar y Actuar” (PDCA);
- 2. Facilitar la autoevaluación de una organización pública con el fin de obtener un diagnóstico y definir acciones de mejora;
- 3. Hacer de puente entre los diferentes modelos que se usan en la gestión de la calidad.
- 4. Facilitar el benchlearning entre las organizaciones del sector público.

2.7 Seis SIGMA

Seis Sigma es una metodología de mejora de procesos, centrada en la eliminación de defectos. Ver figura 13. Se fundamenta en EFQM la estadística, es decir la desviación estándar (3,4 defectos por cada millón). Sus principios son: Prevenir defectos, reducir variación, centrarse en el cliente, tomar decisiones basadas en hechos y alentar el trabajo en grupo. Las fases son:

- Definir el problema y especificar prioridades
- Medir el proceso actual

- Analizar lo que está mal y soluciones
- Mejorar el proceso implantado
- Controlar el proceso mejorado

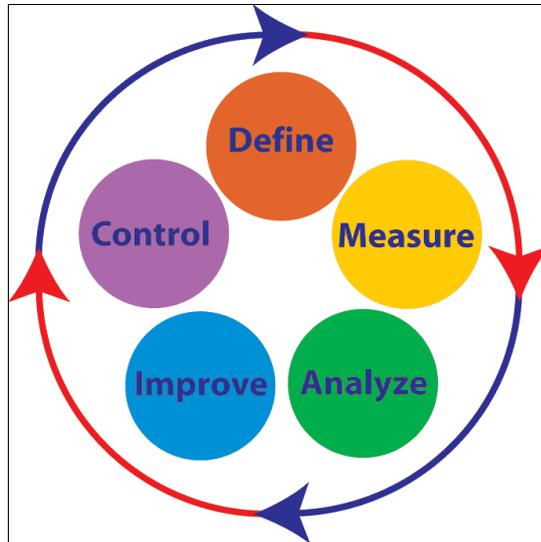


Figura 13 Six Sigma como Mejora de Proceso Continua (6Sigma, 2009)

En cuanto a la calidad del software se puede concluir que los requisitos del software son la base de las medidas de calidad. La falta de concordancia con los requisitos es una falta de calidad. Los estándares o metodologías definen un conjunto de criterios de desarrollo que guían la forma en que se aplica la ingeniería del software. Si no se sigue ninguna metodología siempre habrá falta de calidad

3 Métricas del software

3.1 Clasificación de las métricas del software

La Ingeniería de software, al igual de otras ingenierías necesita de métricas e indicadores para poder especificar, predecir, evaluar y analizar distintos atributos y características de las entidades (productos, productos, etcétera) que participan en el desarrollo y mantenimiento del software. El objetivo de todo proceso de medición es recopilar indicadores cuantitativos sobre entidades software. Para realizar la medición es necesario identificar las entidades y sus atributos a medir.

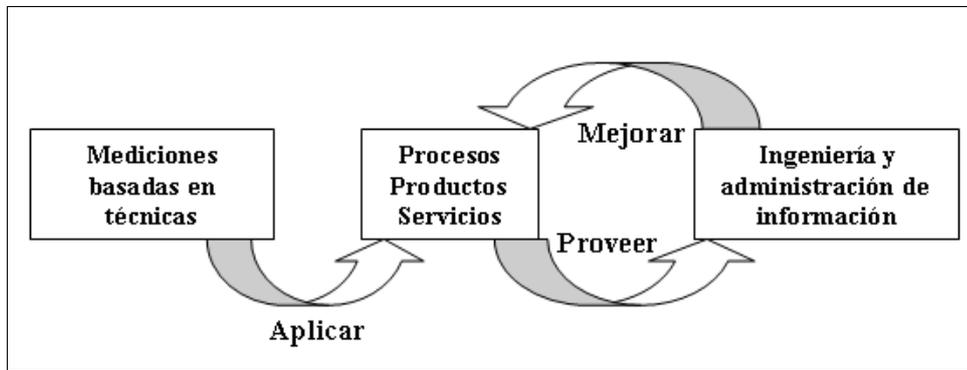


Figura 14 Concepto de Métricas (La Torre, 2008)

La aplicación de un enfoque cuantificable al desarrollo, operación y mantenimiento del software es una tarea compleja que requiere disciplina, estudio y conocimiento de las métricas e indicadores para los distintos objetivos de medición y evaluación, todo con el fin de garantizar la construcción de un software de calidad.

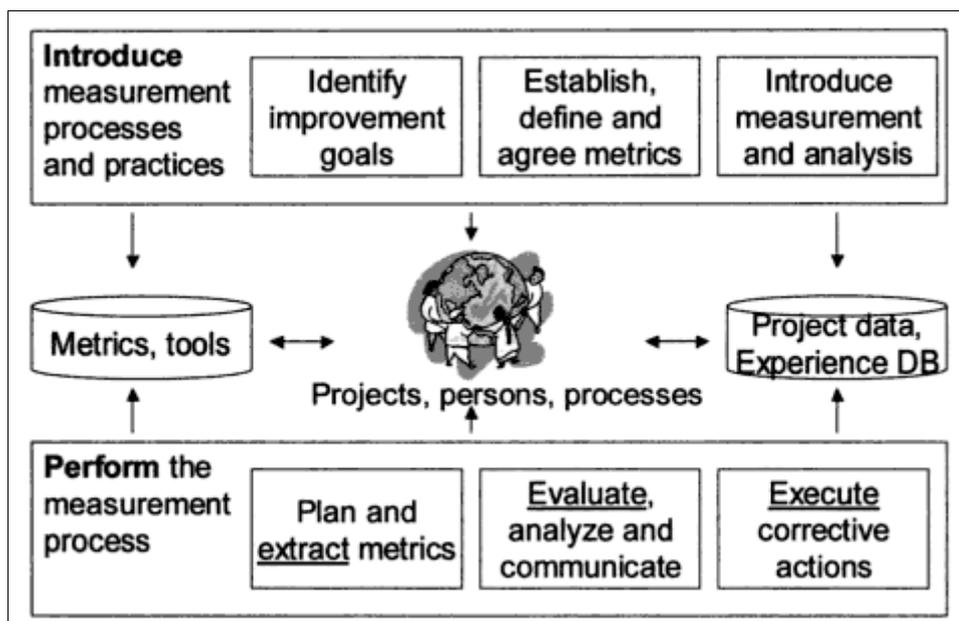


Figura 15 Implementación de Prácticas de Medición de Software (Ebert, Dumke, & Shmietendorf, 2005)

3.1.1 Algunas clasificaciones de métricas

Existen diversas clasificaciones de las métricas, los autores utilizan diferentes abstracciones o enfoques para su clasificación. Una clasificación interesante se muestra a continuación por Parte de Ejiogu:

- Métricas de complejidad: Son todas las métricas de software que definen de una u otra forma la medición de la complejidad; Tales como volumen, tamaño, anidaciones, costo (estimación), agregación, configuración, y flujo. Estas son los puntos críticos de la concepción, viabilidad, análisis, y diseño de software.

- Métricas de calidad: Son todas las métricas de software que definen de una u otra forma la calidad del software; Tales como exactitud, estructuración o modularidad, pruebas, mantenimiento, reusabilidad, cohesión del módulo, acoplamiento del módulo, etc. Estas son los puntos críticos en el diseño, codificación, pruebas y mantenimiento.

- Métricas de competencia: Son todas las métricas que intentan valorar o medir las actividades de productividad de los programadores o practicantes con respecto a certeza, rapidez, eficiencia y competencia.

- Métricas de desempeño: Corresponden a las métricas que miden la conducta de módulos y sistemas de un software, bajo la supervisión del sistema operativo o hardware. Generalmente tienen que ver con la eficiencia de ejecución, tiempo, almacenamiento, complejidad de algoritmos computacionales, etc.

- Métricas estilizadas: Son las métricas de experimentación y de preferencia; Por ejemplo: estilo de código, indentación, las convenciones denominando de datos, las limitaciones, etc. Pero estas no se deben confundir con las métricas de calidad o complejidad.

- Variedad de métricas: tales como portabilidad, facilidad de localización, consistencia. Existen pocas investigaciones dentro del área.

La Clasificación anterior fue dada por (Ejiogu, 1991, pág. 15).

La siguiente clasificación de Fenton parte de número de entidades que abarca o bien de la dependencia de otra métrica

- Métricas Directas: Son aquellas que se obtienen a través de un proceso de medición directo, es decir, que no involucra a ninguna otro atributo. Estas métricas involucran una sola entidad software. Ejemplos: Cantidad de enlaces rotos en una página web, Número de líneas de código.

- Métricas Indirectas: Son aquellas que se obtiene a partir de métricas directas. Se caracterizan de obtener medidas de dos o más entidades software o bien de provenir de otra métrica. Ejemplos: Porcentaje de Enlaces rotos en una aplicación web, Número de líneas de código por semana. (Fenton, 1991).

También las métricas se identifican por la entidad que evalúan de acuerdo al estudio realizado por Sommerville:

- Métricas de Proceso: Miden los atributos relacionados con las actividades relacionadas con el software
- Métricas del Proyecto: Basadas en la gestión del proyecto.
- Métricas del Producto: Miden los hitos de resultantes de una actividad de proceso.

Las métricas de proceso de software se emplean para fines estratégicos, y las métricas del proyecto de software son tácticas, éstas últimas van a permitir proporcionar al desarrollador de proyectos del software una evaluación al proyecto que sigue en continuo desarrollo, equivalentemente podrá ver los defectos que logren provocar riesgos a largo plazo (áreas problema); y observar si el área de trabajo (equipo) y las distintas tareas se ajustarán. En la mayoría de los desafíos técnicos, las métricas ayudan a entender tanto el proceso técnico que se utiliza para desarrollar un producto, como el propio producto. En el proceso para intentar mejorarlo, el producto se mide para intentar aumentar su calidad.

Estas clasificaciones de métricas fortalecen la idea, de que más de una métrica puede ser deseable para valorar la complejidad y la calidad del software.

El proceso software constituye la base a partir de la cual se realiza el trabajo dentro de una organización. Estos procesos en la práctica conforman un proyecto y al elaborar estos se utilizan recursos y se obtienen proyectos.

Las métricas colaboran con el control o la predicción, éstas influyen en la toma de decisiones de la gestión. Las métricas de control suelen estar asociadas con los procesos y las métricas de predicción están asociadas al producto (Sommerville, 2005) Ver figura 16.

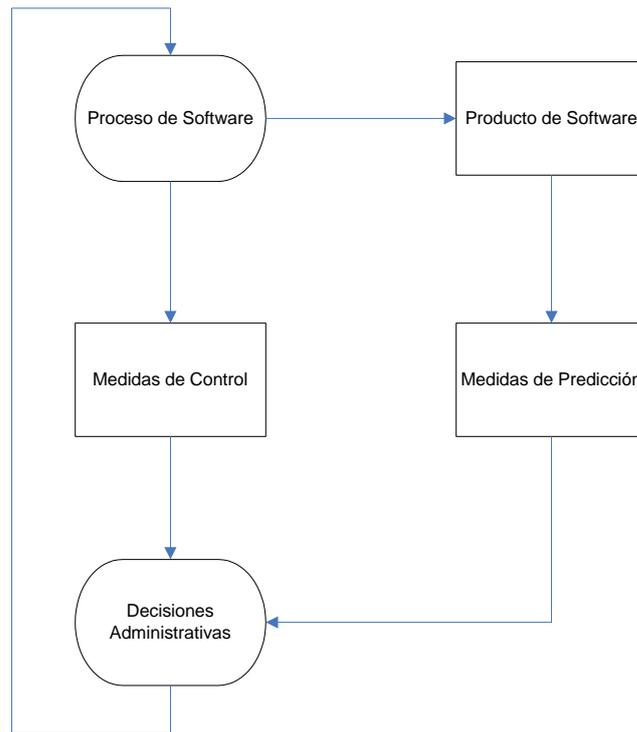


Figura 16 Métricas de Predicción y Control (Sommerville, 2005)

3.2 Métricas del proceso

Implican las mediciones de las actividades relacionadas con el software, por ejemplo: esfuerzo, el coste y los defectos. Los datos cuantitativos de los procesos software, la medición de procesos y producto es importante para la mejora de procesos. Las métricas de procesos se utilizan para evaluar si la eficiencia de un proceso ha mejorado. Por ejemplo se puede medir el esfuerzo y tiempo dedicados a las pruebas según Humphrey.

Se tienen tres clases de métricas de proceso:

- El tiempo requerido para completar un proceso en particular.
- Los recursos requeridos para un proceso en particular.
- El número de ocurrencias de un evento en particular.

Estas métricas evalúan el proceso de fabricación del producto correspondiente. Algunos ejemplos clásicos de este tipo de métricas son el tiempo de desarrollo del producto, el esfuerzo que conlleva dicho desarrollo, el número y tipo de recursos empleados (personas, máquinas), el coste del proceso, etc.

El tiempo requerido para completar un proceso en particular (tiempo total del proyecto, por ingeniero, por actividad, etcétera) es un indicador de la mantenibilidad del sistema a tener en cuenta. Aunque no se puede generalizar,

cuanto mayor es el tiempo total y por ingeniero para desarrollar un sistema mayor será su complejidad y por lo tanto más difícil será de mantener.

De la misma manera, cuantos mayores sean los costes requeridos para un proceso en particular (esfuerzo en personas-día, costes de viajes, recursos de hardware), menor será la mantenibilidad del sistema. Además, de estas, el número de defectos descubiertos durante la fase de pruebas y las métricas relacionadas (Humphrey, 1995).

Los procesos software esta interrelacionado con el producto, la tecnología y el personal para influenciar en la calidad del software y la eficacia organizacional. Ya que con ello se forma el entorno de desarrollo, las condiciones del negocio y características del cliente (Fenton & Pfleeger, 1997).

Estas métricas son medidas indirectamente a través de medidas de errores descubiertos antes de liberar, los defectos que reporten los usuarios finales, los productos de trabajo entregados, el esfuerzo humano gastado, el tiempo de la planificación consumido, concordancia con la planificación, etc.

Grady (Grady, 1992) es referenciado por Phillips menciona que existen datos públicos y privados. Los primeros incluyen información de productividad por individuo y se acopla al PSP (Personal Software Process), ya que se argumenta que la calidad comienza con el individuo. Esta información privada se convierte en pública al ser agrupada toda la información individual de un equipo.

El mismo autor proporciona sugerencias para implementar un programa de métricas del software: concretamente pide sentido común, retroalimentación, establecimiento de metas, no utilizar métricas para medir o amenazar individuos, tomar datos para decisiones de mejora y considerar métricas importantes (Phillips, 1992).

3.3 Métricas del proyecto

Existen dos objetivos que fundamentan las métricas del proyecto: primero son un gran apoyo para minimizar los tiempos de desarrollo ya que proporcionan indicadores que ayudan a realizar ajustes para evitar demoras, identificar riesgos y en medida de lo posible evitar que se conviertan en problemas. Segundo, se utilizan para valorar la calidad del producto, tomando como base la información de proyectos anteriores.

El objetivo de estas métricas es táctico, ya que proporcionan datos cuantitativos para adaptar flujos de trabajo y actividades técnicas. Estas métricas y los indicadores derivados son utilizados por el gestor de proyectos para adaptar el flujo de trabajo y las actividades técnicas. Las primeras métricas se obtienen en la fase de estimación.

A medida que mejora la calidad, los errores se minimizan y el número de defectos disminuyen, la cantidad de trabajo que ha de rehacerse también se reduce. Esto lleva a una reducción del costo global del proyecto.

Otro modelo de métricas del proyecto de software sugiere que todos los proyectos deberían medir: (John A. Mayhauser 1991)

- Entradas: la dimensión de los recursos (p. ej personas, medio ambiente) que se requieren para realizar el trabajo.
- Salidas. medidas de las entregas o productos creados durante el proceso de ingeniería del software
- Resultado. medidas que indican la efectividad de las entregas

En realidad, este modelo se puede aplicar tanto al proceso como al proyecto. En el contexto del proyecto, el modelo se puede aplicar recursivamente a medida que aparece cada actividad estructural. Por consiguiente, las salidas de una actividad se convierten en las entradas de la siguiente. Las métricas de resultados se pueden utilizar para proporcionar una indicación de la utilidad de los productos cuando fluyen de una actividad (o tarea) a la siguiente.

3.4 Métricas de productos

Los productos del software son las salidas del proceso de producción del software. Éstas incluyen todos los artefactos entregados o documentos que son productos durante el ciclo de vida del software. Las métricas de los productos usualmente cuantifican algunos aspectos de calidad relacionados a la tabla 3.

No siempre será posible medir éstos directamente, especialmente cuando las métricas son predecibles, por ejemplo, cuando el software no está en desarrollo y la métrica es usada para predecir su calidad operacional; entonces los subfactores y atributos deliberados son medidas que correlacionan dando como resultado una métrica indirecta.

Hay varias razones para medir un producto:

- 1. Para indicar la calidad del producto.
- 2. Para evaluar la productividad de la gente que desarrolla el producto.
- 3. Par evaluar los beneficios en términos de productividad y de calidad, derivados del uso de nuevos métodos y herramientas de la ingeniería de software.
- 4. Para establecer una línea de base para la estimación.
- 5. Para ayudar a justificar el uso de nuevas herramientas o de formación adicional.

Lista de Requerimientos de Calidad	
Las capacidades de los productos	El grado de la funcionalidad
Usabilidad	fiabilidad
desempeño (eficiencia)	seguridad
factores humanos	Portabilidad
La extensión de la solución	Futuro/expectativas del tiempo de vida
Compatibilidad	Mantenibilidad:
Adaptabilidad	Correctivo
Recursos	Perfectivo
Limitaciones personales	Instalación
Limitaciones presupuestales	Horario
Beneficios:	Limitaciones físicas
Tangibles	Documentación
Intangibles	Limitaciones organizacionales

Tabla 3 Requerimientos de Calidad (Mayhauser,1991)

3.5 Modelo de la planificación predictiva

Es intuitivamente obvio que un programa largo tiende a ser más complejo y más costoso a desarrollar y mantener. Un programa largo también tiende a tener un vocabulario grande y usa más operaciones y operandos. También tiene más líneas de código (LOC). La figura 4 nos muestra una relación entre métricas y el modelo de desarrollo para las variables que planean. El signo “?” en la fórmula nos indica un factor de tamaño en donde puede estar influyendo en los parámetros de la salida del modelo. Este modelo de fórmulas sólo trabaja si el tamaño tiene una crucial influencia en el tiempo, costo y requisitos de los recursos para un proyecto y si la relación entre tamaño y complejidad es fuerte.

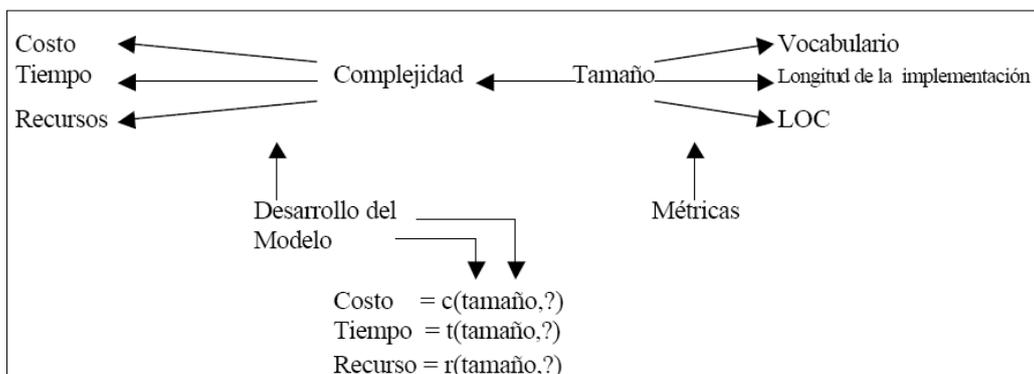


Tabla 4 Métricas del Producto y modelo de Desarrollo (Piattini,2010)

Cuando el tamaño no es el único contribuidor significativo a la variable planeada, el modelo a predecir entonces será multidimensional y más complejo.

Un modelo es una relación de la forma $Y = f(x_1, x_2, \dots, x_m)$. Y es la variable que deseamos predecir. Esta es llamada variable dependiente. La X_i son las variables independientes en que la predicción depende de Y. Mayrhauser menciona que hay dos caminos para establecer esta relación funcional:

- El primero es, por análisis estadístico de los datos del caso. Este es un modelo empírico. Muchos modelos de costo de estimación son de este tipo, considerándose son tan representativos como los datos sobre los que estos son basados y la calidad de los métodos estadísticos de la inferencia que se usan.
- El segundo camino se lleva a cabo a través de razonamiento analítico y pruebas que establecen la relación entre variables independientes y dependientes. Algunos modelos analíticos postulan una familia de funciones con situaciones constantes.

Estos tienen que ser medidos separadamente o estimados por métodos estadísticos. El modelo se calibra a su medio (o ambiente) en la aplicación.

3.6 Integración de las Métricas dentro del Proceso de Software

La mayoría de los desarrolladores de software todavía no miden, y por desgracia, la mayoría no desean ni comenzar. Como se ha señalado, el problema es cultural. En un intento por recopilar medidas en donde no se haya recopilado nada anteriormente, a menudo se opone resistencia: “¿Por qué necesitamos hacer esto?”, se pregunta un administrador de proyectos fatigado. “No entiendo por qué”, se queja un profesional saturado de trabajo. ¿Por qué es tan importante medir el proceso de ingeniería del software y el producto (software) que produce? La respuesta es relativamente obvia. Si no se mide, no hay una forma real de determinar si se está mejorando, y si no se está mejorando, se está perdido. La medición es una de las “medicaciones” que pueden ayudar a curar el “mal del software”. Esta proporciona beneficios al nivel de proyecto, estratégico y técnico.

La administración de alto nivel puede establecer objetivos significativos de mejora del proceso de ingeniería del software solicitando y evaluando las medidas de productividad y de calidad. Se señaló que el software es un aspecto de administración estratégico para muchas compañías. Si el proceso por el que se desarrolla puede mejorarse, entonces puede producirse un impacto directo en lo sustancial. Pero para establecer objetivos de mejora, se debe comprender el estado actual de desarrollo del software.

Los rigores del trabajo diario de un proyecto del software no dejan mucho tiempo para pensar en estrategias. Los administradores del proyecto de software están más preocupados por aspectos mundanos desarrollo de estimaciones significativas del proyecto, producción de sistemas de alta calidad

y terminar el producto a tiempo. Mediante el uso de la medición para establecer una línea base del proyecto, cada uno de estos asuntos se hace más fácil de manejar. Ya se ha apuntado que la línea base sirve como un lineamiento para la estimación. Además, la recopilación de métricas de calidad permite a una organización entender su proceso de ingeniería del software para eliminar las causas originales de los defectos, que tienen el mayor impacto en el desarrollo del software.

Técnicamente las métricas del software, cuando se aplican al producto, suministran beneficios inmediatos. Cuando se ha terminado el diseño del software, la mayoría de los que desarrollan pueden estar ansiosos por obtener respuestas a preguntas como:

¿Qué requisitos del usuario son más susceptibles al cambio?

¿Qué módulos del sistema son más propensos a error?

¿Cómo se debe planificar la prueba para cada módulo?

¿Cuántos errores (de tipos concretos) puede esperar cuando comience la prueba?

Se pueden encontrar respuestas a esas preguntas si se han recopilado métricas y se han usado como guía técnica.

El proceso que se establece en una línea base (datos recopilados de Proyectos de desarrollo de software anteriores), ver figura 17, idealmente, los datos necesarios para establecer una línea base han sido recopilados a medida que se ha ido progresando. Por desgracia, este no es el caso. Por consiguiente, la recopilación de datos requiere una investigación histórica de los proyectos anteriores para reconstruir los datos requeridos, una vez que se han recopilado medidas (el paso más difícil), el cálculo de métricas es posible. Dependiendo de la amplitud de las medidas recopiladas, las métricas pueden abarcar una gran gama de métricas, tales como: LDC y PF, así como métricas de calidad y orientadas a objetos. Finalmente, las métricas se deben evaluar y aplicar durante la estimación, el trabajo técnico, el control del proyecto y la mejora de proyectos, la evaluación de métricas se centra en las razones de los resultados obtenidos, y produce un grupo de indicadores que guían el proyecto o el proceso (Pressman, 2005).

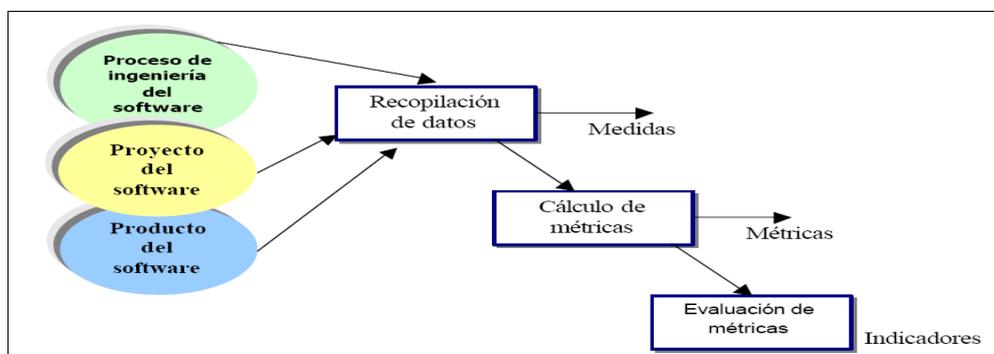


Figura 17 Proceso de Recopilación de Métrica (Pressman, 2005)s

4 Usabilidad del software

4.1 Justificando la Usabilidad

Seguramente el lector ha tenido experiencias frustrantes al utilizar un teléfono celular, un fax, un auto, un sistema software y que sin duda cumplieron su función, pero no fue fácil, se invirtió mucho tiempo, se tuvo que revisar varias veces el manual de usuario o simplemente fue terrible. Es difícil imaginar que un dispositivo para contrarrestar incendios requiera de habilidades específicas, podría ser la diferencia entre la vida y la muerte. En la vida cotidiana se utilizan máquinas que tan sólo por las instrucciones ya se creó confusión y optamos por un método manual. Esto hace que la usabilidad impacte a toda la gente, todos los días, independiente al país economía o clase.

En el siglo XXI la interacción con las tecnologías de información es innegable, hasta las regiones más recónditas los gobiernos han generado proyectos de implementación de tecnología para la comunicación o investigación, en la mayoría de las partes del mundo los niños desde pequeños usan la computadora, los teléfonos celulares, etcétera. El acceso a Internet permite que una diversidad de personas se vea involucrada en la consulta de información digital por lo que los sistemas software requieren de características más que intuitivas para ser usados. Pero la usabilidad como característica también debe estar presente en objetos (máquinas, sistemas software, aparatos de medicina) especializados, no sólo en objetos comunes.

Todavía el siglo pasado el problema en algunos países era la cantidad de personas que no sabían leer (sin duda todavía existe, pero no en la misma proporción), ahora la población demanda sistemas software para realizar procesos más rápidamente, para la transparencia de información para con los gobiernos, para leer libros virtuales.

La tecnología crece y se vuelve más compleja, si a ello se agrega un diseño también complejo y éste no se controla, entonces se dará continuación a la crisis del software.

La usabilidad es una característica del software bastante compleja y que abarca una gama muy amplia de aspectos. Para que un producto software sea usable es necesario ocuparse de este atributo de calidad durante prácticamente todo el proceso de construcción de dicho sistema, desde la especificación de requisitos hasta las pruebas.

La interacción con los sistemas software se ha convertido en parte de la vida diaria del ser humano, de tal manera que es un factor a considerar en la construcción de estos es, precisamente, la Interacción Computadora-Humano (HCI). Las características genéricas de cómo percibe la realidad debe ser considerado en el diseño, análisis y evaluación de interfaces computadora-humano. Los ingenieros en software deben considerar HCI como un factor para la calidad de los sistemas software, específicamente en la característica de la usabilidad.

4.2 Definición

Se considera a Jakob Nielsen el padre de la usabilidad, que se especializa en diseño para Internet. Nielsen hace referencia a la usabilidad como un atributo de calidad que evalúa como las interfaces de usuario son fáciles de usar (ALERTBOX, 2009). La palabra usabilidad hace referencia a métodos para mejorar el uso-fácil durante el proceso de diseño. Éste término es utilizado para muchos dispositivos u objetos, no sólo para el sistema software.

Etimológicamente usabilidad proviene de dos palabras usar + habilidad y se puede traducir como “el estado o condición de ser usado ó el grado en el cual un objeto dispositivo o software es fácil de usar sin un previo entrenamiento” (WIKT, 2009). (Tullis, William, 2008) hace referencia a tres definiciones de usabilidad:

- (ISO, 9241-11) “la extensión con la cual un producto puede ser usado y los usuarios logran metas específicas con efectividad, eficiencia y satisfacción en un contexto de uso específico”
- La asociación de Profesionales en Usabilidad (UPA por sus siglas en inglés) se centra en el proceso de desarrollo del producto, “la usabilidad es la aproximación al desarrollo de un producto que incorpora retroalimentación del usuario en el ciclo de desarrollo para reducir costos y crear productos y herramientas para el usuario común”.
- (Steve Krug 2000) afirma que “usabilidad es asegurarse que algo trabaja bien: que una persona con habilidad y experiencia promedio (o incluso menos) puede usar el objeto (software, avión, dispositivo, etcétera) para sus propósitos sin que obtenga frustraciones o desesperanza”

En palabras claras y simples se puede decir que la usabilidad se refiere a la capacidad de un software de ser comprendido, aprendido, usado y ser atractivo para el usuario, en condiciones específicas de uso.

La usabilidad está directamente relacionada a la interfaz de un programa, un sitio Web, un periférico o un sistema. Por ejemplo, en una aplicación, su usabilidad aumenta con el uso de gráficos simples y comprensibles, métodos estandarizados de formas y accesos, y otro tipo de facilidades. En tanto, la usabilidad de una aplicación disminuye si no utiliza formas y métodos de acceso comunes a otras aplicaciones, no utiliza gráficos, etc.

Es claro que se trata de un requerimiento no funcional, por lo que no puede ser directamente medida, pero como toda entidad medible, se tienen que identificar sus atributos que si se pueden medir.

4.3 Atributos

Para Schneiderman hace referencia a los atributos de la usabilidad según Nielsen:

- Facilidad de aprendizaje. Se refiere a que tan rápido el usuario va a aprender a usar un sistema con el cual no había tenido contacto previamente. Este punto se refiere a la consecución de tareas básicas por parte de un usuario principiante.

- Velocidad de desempeño (eficiencia al ser usado. Una vez que el usuario ha aprendido a utilizar el sistema, se va a ponderar el lograr la velocidad con que puede completar una tarea específica.

- Retención sobre el tiempo ó Memorabilidad. Cuando un usuario ha utilizado un sistema tiempo atrás, y tiene la necesidad de utilizarlo de nuevo la curva de aprendizaje debe de ser significativamente menor que el caso del usuario que nunca haya utilizado dicho sistema. Esto es de primordial importancia para aplicaciones usadas intermitentemente.

- Satisfacción subjetiva. Se refiere a la impresión subjetiva del usuario respecto al sistema.

- Tasas de error por parte de los usuarios. Este apartado apunta hacia los errores cometidos por el usuario. Este atributo se refiere a aquellos errores que comete el usuario al utilizar el sistema. Una aplicación ideal evitaría que el usuario cometiera errores y funcionaría de manera óptima a cualquier petición por parte del usuario. En la práctica esto difícilmente se logra. Es vital que una vez que se produzca un error el sistema se lo haga saber rápida y claramente al usuario, le advierta sobre la severidad del mismo y le provea de algún mecanismo para recuperarse de ese error

Con estos atributos se puede apreciar que el usuario tiene un papel importante, es el que hace algo con el objeto (lo usa), por lo que debe ser considerado para detallar la usabilidad, la cuestión que se presenta en este momento es bajo que apreciaciones de éste se va a determinar la usabilidad (Schneiderman, 1998).

De acuerdo a las características de Memorabilidad, fácil de Aprender y satisfacción subjetivas, se deben hacer consideraciones acerca de la psicología, la fisiología y sociología del individuo.

La usabilidad también hacer referencia al grado de facilidad con que una aplicación, sitio Web, periférico o sistema, se adapta a sus usuarios (incluso a usuarios con capacidades disminuidas). La usabilidad puede estar relacionada incluso a la Ergonomía y a la potabilidad de un dispositivo (Nielsen, Usability Engineering, 1993).

Algunos autores no consensan acerca de la relación entre ergonomía (factores humanos) y usabilidad. Se llega a la confusión de que usabilidad es una especialización del software en el tema de la ergonomía, también se tiene idea que es un tema lateral, es decir, ergonomía con enfoque sobre cuestiones fisiológicas (por ejemplo manejar un auto estándar) y usabilidad con enfoque sobre cuestiones psicológicas (por ejemplo reconocer que el trascabo se puede operar de manera estándar). Por ejemplo al usar el sistema del IFAI para la transparencia de la información, sistema software que está dirigido a todos los

mexicanos, se puede detectar que existen términos no conocidos en el lenguaje común de los mexicanos, por lo que algunos ciudadanos desisten de usarlo y por ende investigar (Proceso, 2009), lo que hace que el contexto sociológico también deba ser considerado en la usabilidad.

4.4 Usabilidad y accesibilidad

Accesibilidad significa proporcionar flexibilidad para acomodarse a las necesidades de cada usuario y a sus preferencias y/o limitaciones.

Los seres humanos son diferentes entre sí y en un mundo ideal todas las interfaces de usuario deberían acomodarse a esas diferencias, de tal modo que cualquier persona fuera capaz de utilizarlas sin problemas, sin que nadie se vea limitado en el uso de algo por causa de esas diferencias personales. Es necesario evitar diseñar solamente atendiendo a características de grupos de población específicos, imponiendo barreras innecesarias que podrían ser evitadas prestando más atención a las limitaciones de éstos.

Las capacidades y aptitudes de todas las personas difieren de unas a otras. Existen grupos de población que tienen alguna limitación funcional que les impide acceder a facilidades que desearían, deberían o tienen el derecho de acceder. Por poner algún ejemplo ilustrativo, pertenecen a los grupos mencionados personas con baja o nula visión y/o audición, con discapacidades motrices que les impiden el libre movimiento de sus manos o reducidos niveles de comprensión.

El factor de proporcionar accesibilidad a los sistemas interactivos por las personas es tan importante que en el 2003 el estándar ISO ha publicado la especificación técnica ISO/TS 16071 enmarcada en las especificaciones ergonómicas para que sirva de guía para el diseño de interfaces de ordenador para las personas.

Esta especificación técnica insiste en la necesidad de considerar los aspectos sociales y legislativos para eliminar aquellas barreras que impiden que personas con necesidades especiales puedan participar en las actividades de la vida diaria incluyendo cualquier tipo de servicio, producto o información.

Es importante destacar que la accesibilidad se proporciona mediante una combinación de hardware y software: El primero proporciona los mecanismos físicos que permiten salvar ciertas discapacidades y el segundo proporciona la manera eficaz de acceder a las funcionalidades e informaciones para estos dispositivos y a otros programas (por ejemplo, un navegador web). La especificación técnica mencionada hace referencia sólo a los aspectos del componente software.

Existen mecanismos y herramientas que se adaptan al ordenador a la vez existen estudios y guías para desarrollar aplicaciones que permiten facilitar el uso a las personas con discapacidades.

Usualmente las interfaces se diseñan pensando en una persona estándar con todas las capacidades físicas y cognitivas, lo que frecuentemente deja

fuera a los colectivos de personas con necesidades especiales. El Diseño Universal, conocido también como Diseño para Todos, tiene como objetivo diseñar interfaces que no presenten barreras de accesibilidad. Para ello, es necesario que la interfaz admita el uso de dispositivos de interacción alternativos, adecuados a las capacidades físicas de cada usuario.

Respecto de la necesidad de tener en cuenta la diversidad de capacidades cognitivas, es necesario, por ejemplo, limitar la necesidad de memorizar datos, utilizar metáforas adecuadas a las diversas culturas, a las experiencias previas de los usuarios y al pensamiento humano en general para producir, por ejemplo, sistemas de navegación coherentes e intuitivos.

No sólo se trata pues de accesibilidad para personas con discapacidad, se trata del Diseño para Todos. Realizando los cambios requeridos por las personas con discapacidad se beneficia a todos.

Los ejemplos incluyen a personas con módems lentos que desactivan las imágenes, personas que navegan por la web mientras conducen un automóvil e incluso médicos que acceden a la web mientras sus manos están ocupadas con una intervención quirúrgica.

Así pues, el Diseño Universal supone una estrategia cuyo objetivo es la realización y la composición de los diferentes entornos y productos accesibles y comprensibles, a la vez que usables, en todo el mundo, en la mayor o menor medida y de la forma más independiente y natural posible, sin la necesidad de adaptaciones ni soluciones especializadas de diseño.

En el diseño para todos se interviene sobre los entornos, productos y servicios con la finalidad de que todo el mundo, incluyendo generaciones futuras, independientemente de la edad, el género, las capacidades o la cultura, puedan participar en la construcción de nuestra sociedad, con igualdad de oportunidades para participar en actividades económicas, sociales, culturales, de ocio y recreativas y puedan acceder, utilizar y entender cualquier parte del entorno lo más independientemente posible.

Existen unos Principios de Diseño Universal redactados por un grupo de expertos que sirve de guía para evaluar la incorporación de la accesibilidad en el diseño de los sistemas interactivos. Estos criterios que definen el diseño utilizable para todos no tienen en cuenta aspectos como la estética, el coste, la seguridad o el respeto a la diversidad, aunque deben estar presentes durante el proceso de diseño.

En el escenario europeo existe el grupo de trabajo que forma parte de ERCIM39 denominado "User Interfaces for All" que sistemáticamente promueve la realización proactiva de principios de Diseño para Todos en el área de la HCI. Este grupo trata el desarrollo de interfaces de usuario de aplicaciones interactivas y servicios telemáticos propiciando el acceso y la usabilidad universales para todos los usuarios potenciales, incluyendo personas con diferencias culturales, educativas, de entrenamiento y de empleo, usuarios de ordenadores noveles y expertos, jóvenes y personas con diferentes tipos de discapacidades, todos ellos interactuando con diferentes plataformas tecnológicas en contextos de uso distintos (GRIHO, 2009).

4.5 Ingeniería de la usabilidad

La Ingeniería de usabilidad (IU) es un área de HCI (Human-Computer Interaction, Interacción Humano-Computador) que da pautas para obtener productos con un alto grado de usabilidad, esto mediante la aplicación de distintos métodos en diferentes etapas del proceso de diseño y desarrollo de una manera estructurada y sistemática. El objetivo principal de la IU es mejorar la interfaz de usuario.

En las primeras etapas de desarrollo del producto, la evaluación de usabilidad será utilizada para decidir sobre distintos diseños de interfaces de usuario y finalmente decidir cual será en el que se trabajará. En las siguientes etapas, las evaluaciones de usabilidad ayudarán a verificar que el sistema cumpla con los requerimientos iniciales.

En IU se trata de decidir que atributos del concepto de usabilidad deben de ser priorizados, con el fin de lograr metas verificables y medibles de niveles de usabilidad (Lecerof & Paternó, 1998). Por ejemplo:

- Medir el desempeño de un usuario ejecutando una serie de tareas específicas con respecto al tiempo de terminación de las tareas o en base al número de errores cometidos.
- Determinar los niveles de preferencia subjetiva o el grado de satisfacción.
- La facilidad de aprendizaje podría medirse en base al número de tareas completadas en cierto periodo, número de errores cometidos, o respecto al número de veces que utilizó la opción de ayuda.

4.6 Estándares de Proceso de Desarrollo

Ferré identifica los siguientes análisis de los estándares, que de manera simple o integral contextualizan la usabilidad:

- ISO/IEC 12207:1995. Define un marco común para procesos de ciclo de vida, con una terminología precisa, que puede usarse como referencia para la industria del software. No especifica el papel de las prácticas de usabilidad, sino de manera lateral.
- IEEE 1074-1997. Está destinado al arquitecto de proceso, que debe definir el proceso software que se va a aplicar en una organización de desarrollo. Nombra la usabilidad como objetivo genérico sin llegar en ninguna actividad a identificar de qué manera se puede lograr que el sistema software alcance un nivel suficiente de usabilidad.
- Enmienda 1 al estándar ISO/IEC 12207. Incluye el nuevo proceso de usabilidad entre los procesos de soporte del ciclo de vida y está basada en la información del estándar 13407:1999, el cual trata del estándar de referencia en proceso de desarrollo de campo de la Interacción Hombre-Computadora (HCI). Incorpora las actividades de

Implementación del Proceso de usabilidad, diseño centrado en el humano y aspectos humanos de estrategia, introducción y soporte. Además de incluir la usabilidad a los proceso de desarrollo, hace referencia al especialista en usabilidad que se integre al equipo de desarrollo.

Este compendio realizado por (Ferré, 2005) es de gran utilidad para entender fácilmente los estándares en donde está implicada la usabilidad.

4.7 CONSIDERACIONES DE USABILIDAD

Fuera del ámbito informático, la usabilidad está más relacionada con la ergonomía y los factores humanos.

La ergonomía parte de los principios del diseño universal o diseño para todos. La buena ergonomía puede lograrse mediante el diseño centrado en el usuario, aunque se emplean diversas técnicas. El diseñador de ergonomía proporciona un punto de vista independiente de las metas de la programación porque el papel del diseñador es actuar como defensor del usuario. Por ejemplo, tras interactuar con los usuarios, el diseñador de ergonomía puede identificar necesidades funcionales o errores de diseño que no hayan sido anticipados (WIKT, 2009).

La ergonomía incluye consideraciones como:

- ¿Quiénes son los usuarios, cuáles sus conocimientos, y qué pueden aprender?
- ¿Qué quieren o necesitan hacer los usuarios?
- ¿Cuál es la formación general de los usuarios?
- ¿Cuál es el contexto en el que el usuario está trabajando?
- ¿Qué debe dejarse a la máquina? ¿Qué al usuario?

Las respuestas a estas preguntas pueden conseguirse realizando análisis de usuarios y tareas al principio del proyecto. Otras consideraciones incluyen:

- ¿Pueden los usuarios realizar fácilmente sus tareas previstas? Por ejemplo, ¿pueden los usuarios realizar las tareas previstas a la velocidad esperada?
- ¿Cuánta preparación necesitan los usuarios?
- ¿Qué documentación u otro material de apoyo están disponible para ayudar al usuario? ¿Puede éste hallar las respuestas que buscan en estos medios?
- ¿Cuáles y cuántos errores cometen los usuarios cuando interactúan con el producto?

¿Puede el usuario recuperarse de los errores? ¿Qué han de hacer los usuarios para recuperarse de los errores? ¿Ayuda el producto a los usuarios a recuperarse de los errores? Por ejemplo, ¿muestra el software mensajes de errores informativos y no amenazantes?

¿Se han tomado medidas para cubrir las necesidades especiales de los usuarios con discapacidades? (Es decir, ¿se ha tenido en cuenta la accesibilidad?)

4.8 Diseñando con usabilidad

Cualquier sistema que diseñado para gente debe ser fácil de ser usado, fácil de aprender y aceptado por los usuarios, para diseñar con usabilidad se tienen tres principios identificados por (Gould, Lewis, 2009):

- Enfoque Anticipado en usuario y tareas
- Medición Empírica
- Diseño Iterativo

4.8.1 Enfoque anticipado en usuarios y tareas

El equipo de diseño debe ser manejado y estar en contacto con usuarios potenciales. Varios Métodos de evaluación incluyen personas, modelado cognitivo, inspecciones, observación, prototipado y métodos de prueba con los usuarios potenciales.

Las consideraciones de la usabilidad así como los usuarios y sus experiencias con sistemas similares deben ser examinados. Como parte del entendimiento de los usuarios, el conocimiento debe tomar un papel en las tareas que el usuario espera realizar.

Esto incluye el análisis de que tareas realizarán los usuarios, cuáles con más importantes y qué decisiones tomarán los usuario mientras usan el sistema. Los diseñadores deben entender las características cognitivas y emocionales del usuario y que están relacionados con el propósito del sistema.

Una forma de hacer hincapié de éstas cuestiones en la mente de los diseñadores es usar personas, las cuales deben ser usuarios representativos. Otra forma más cara es tener a usuarios potenciales trabajando con el equipo de diseño desde el inicio del desarrollo de software.

4.8.2 Medición Empírica

Se hace hincapié a dos factores de medición empírica:

- Pruebas tempranas sobre el software.

□ Pruebas del sistema con usuario potenciales usando mediciones de comportamiento.

Esto incluye pruebas del sistema para la aprendibilidad y usabilidad. Es importante en esta etapa usar datos cuantitativos de la usabilidad como tiempo y errores para completar tareas específicas y número de usuarios para probar, así como también examinar la eficiencia y actitudes de los usuarios probando el sistema.

Revisar o demostrar un sistema antes de las pruebas de usuario. Puede tener resultados engañosos. El énfasis de la medición empírica está en la medición, la medición formal e informal pueden llevarse a cabo e través de una variedad de métodos de evaluación.

4.8.3 Diseño Iterativo

El diseño iterativo es una metodología de diseño basada en un proceso cíclico de prototipado, pruebas, análisis y refinamiento de un producto o proceso. Basado en los resultados de la prueba de la iteración inmediata anterior son realizados cambios y refinamientos. Este proceso tiene como objetivo mejorar la calidad y funcionalidad de un diseño. En un diseño iterativo la interacción con el sistema diseñado es una forma para investigar, para informar y evolucionar un proyecto. Los puntos clave para el diseño iterativo son: identificación de cambios requeridos, habilidad para hacer cambios y buena voluntad para hacer cambios. Cuando un problema es encontrado no existe la receta para dar una solución correcta, pero hay métodos empíricos que pueden ser usados durante el desarrollo del sistema o después de que el sistema es liberado, usualmente en un momento inoportuno. En los últimos tiempos el diseño iterativo trabaja para satisfacer metas específicas, así como para hacer el sistema más amigable, fácil de usar, fácil de operar, simple y más.

4.9 Métodos de evaluación

4.9.1 Evaluación de usabilidad

De acuerdo a Ferré la principal actividad en el proceso de usabilidad es la evaluación. La evaluación de la usabilidad puede ayudar a determinar cuál es el nivel actual de la aplicación y si de hecho el diseño elegido realmente funciona.

Existen diferentes técnicas de inspección para evaluar un sistema. Su uso depende de variables tales como costo, disponibilidad de tiempo, personal calificado para interpretar los datos, entre otros factores. A continuación se describen brevemente algunos de estos métodos de inspección o evaluación:

- *Inspección formal de usabilidad:* Un grupo de expertos realizan una especie de juicio de la interfaz, con uno de los participantes actuando como moderador, destacando las fortalezas y las debilidades de la aplicación.
- *Testeo de usabilidad:* Se realizan pruebas de desempeño de un grupo de usuarios utilizando el sistema a probar y se graban los resultados para un análisis posterior. Esta actividad se puede desarrollar en un laboratorio con condiciones controladas o directamente en el lugar donde se va a utilizar el sistema.
- *Pensar en voz alta:* Se le pide al usuario que realice una serie de tareas específicas. El usuario debe de expresar sus acciones oralmente. Dentro de las instrucciones dadas al usuario de prueba no se le pide que explique sus acciones, simplemente que cada paso que realice lo diga en

voz alta (generalmente el mismo usuario da una serie de explicaciones sin pedírselo de manera explícita).

- *Evaluación heurística y de estándares*: En el área de Interfaces de Usuario existen estándares y de heurísticas aceptados y probados. En este tipo de evaluación un equipo de Ingenieros en usabilidad realizan una revisión conforme a estas normativas.

- *Caminata cognitiva*: Un grupo de expertos simula la manera en como un usuario caminaría por la interfaz al enfrentarse a tareas particulares

De manera clara y concisa se pueden resumir con ésta lista los métodos de evaluación de la usabilidad (Ferré, 2005).

Los métodos de diseño centrado en el usuario permiten lograr metas de usabilidad mucho más altas que las que se podrían lograr aplicando sólo la intuición, las recetas o calculando lo que se supone que los usuarios harán con el producto software.

Muchas de las técnicas se basan en involucrar a los usuarios en todas las etapas del diseño. No se trata de diseñar para los usuarios sino de diseñar con los usuarios (TOOLS, 2009).

El diseño como disciplina puede aportar conocimientos y experiencias muy enriquecedoras para la usabilidad y la accesibilidad. Por ejemplo en temas como tipografía, legibilidad, énfasis, estilo, teoría del color, espacio, composición y equilibrio visual.

La usabilidad y la Arquitectura de la Información aportan sus métodos de diseño centrado en el usuario, técnicas de evaluación, los principios heurísticos, 40 años de estudios sobre Interacción Persona Ordenador (HCI), todas las experiencias previas a la web provenientes del diseño de Interfaces Gráficas de Usuario (GUI) y el aporte de ciencias como archivología y bibliotecología para organizar y rotular espacios compartidos de información.

El diseño y la usabilidad pueden y deben complementarse como disciplinas. Las ventajas pueden verse claramente desde ambos lados. Un buen diseño visual no excluye a la usabilidad sino que la favorece, del mismo modo que un alto grado de usabilidad y el respeto por los usuarios contribuyen a mejorar la imagen de cualquier empresa o institución.

4.10 Evaluación heurística

La evaluación heurística es una forma de inspección de la usabilidad, en donde un especialista de la usabilidad, sigue una lista de heurística de usabilidad establecida.

Por lo regular dos o tres evaluadores de análisis de sistemas realizan su trabajo con una guía establecida, anotan sus observaciones con un grado de severidad. Los evaluadores por lo regular son expertos en HCI, pero algunas

veces la evaluación la realizan otros menos experimentados y realizan reportes con problemas de validez.

Una evaluación heurística puede ser conducida a varias fases el desarrollo del ciclo de vida, aunque es preferible tener realizado una forma de análisis del contexto para ayudar a los expertos a conducir entender las circunstancias del producto (Nielsen, Usability Engineering, 1993).

4.10.1 Las 10 reglas heurísticas

La gama de Heurísticas de usabilidad es muy amplia. La más clásica, la introducida en la conferencia “Evaluación Heurística de interfaces de usuario” en abril de 1990, dada la temporalidad y evolución de los sistemas:

Descripción Heurística

1. Visibilidad del contexto. El Sistema debe mostrar a los usuarios dónde se encuentran y de dónde vienen.
2. Coincidencia entre el sistema y el mundo real. El Sistema deberá expresarse en el lenguaje del usuario, con palabras, frases y conceptos que le sean familiares, cuidándose de no hacerlo con términos propios del sistema informático.
3. Libertad y control por parte del usuario. Mínima cantidad posible de restricciones a los usuarios, permitiéndoles elegir los caminos y las formas de cumplir sus objetivos. Evitar desactivar los controles del navegador.
4. Consistencia y estándares. Los usuarios no deben tener necesidad de discernir si palabras, situaciones o acciones distintas significan lo mismo.
5. Prevención de errores. Sensiblemente mejor que buenos mensajes de error es un diseño cuidadoso que anticipa y previene la ocurrencia de los problemas.
6. Reconocer es mejor que recordar. Minimizar la carga en la memoria del usuario haciendo los objetos, acciones y opciones visibles.
7. Flexibilidad y eficiencia de uso. El Sistema debe estar optimizado para minimizar el esfuerzo que requiere al usuario alcanzar sus objetivos. No solicitar jamás información innecesaria, acortar al mínimo los formularios y procesos.
8. Diseño minimalista y estética. Las páginas no deben contener información que sea irrelevante o remotamente necesaria
9. Escribir para el usuario. Los textos y otros contenidos deben estar optimizados para la Web desde el punto de vista de los usuarios.

10. Ayudar a los usuarios a reconocer, diagnosticar y recuperarse de los errores. Los mensajes de error deben expresarse en lenguaje habitual, indicar con precisión el problema y sugerir constructivamente una solución

Así es como (Nielsen & Molich, Useit, 2005) diseñaron los principios heurísticos para los sistemas de ésta época.

4.10.2 Beneficios

De forma barata y rápida provee de retroalimentación a los diseñadores, los resultados generan buenas ideas para mejorar la interfaz de usuario. La evaluación heurística o por criterios es la realizada por evaluadores especializados a partir de principios establecidos por la disciplina de la IPO/HCI. Esta evaluación detecta aproximadamente el 42% de los problemas graves de diseño y el 32% de los problemas menores, dependiendo del número de evaluadores que revisen el software. Posteriormente se recomienda realizar un test de usuarios para completar la evaluación.

Los problemas de usabilidad encontrados son normalmente enfocados a aspectos de la interfaz que son fácil de demostrar: son de colores, disposición y estructura de la información, consistencia de la terminología, consistencia de los mecanismos de interacción.

En comparación con otras técnicas de evaluación donde el observador debe interpretar las acciones del usuario, en la evaluación heurística no es necesaria la interpretación externa, porque las ideas, comentarios e información elaborada por los evaluadores están contenidos en sus informes. Otra ventaja es que en la evaluación por criterios es posible interrogar a los evaluadores, profundizar en determinadas cuestiones de interés y ayudarles cuando tienen problemas. En los Pruebas de usuario por el contrario, los usuarios no deben disponer de más información que la necesaria para permitir su comportamiento espontáneo. Detectar previamente los problemas más graves de usabilidad en la evaluación por criterios permite realizar posteriormente pruebas de usuarios con mejores resultados.

4.10.3 Método

Se ha observado que aún en base a los mismos principios heurísticos diferentes personas encuentran diferentes tipos de problemas en un software, por ello se han de utilizar varios evaluadores, se recomienda 2 ó 3 según Nielsen.

Se usa para identificar problemas de usabilidad basados sobre principios de factores humanos. El método proporciona recomendaciones para las mejoras del diseño. Sin embargo, como el método depende de expertos, la salida hará hincapié en funcionalidad de la interfaz y el diseño más que las propiedades de la interacción entre el usuario y el producto (Ferré, 2005).

4.10.3.1 Planificación.

El panel de expertos debe ser establecido con tiempo para la evaluación, así como el material y el equipo necesario. Todos necesitan tiempo suficiente para familiarizarse con el producto, en este caso producto software, operando con un conjunto de criterios de evaluación.

4.10.3.2 Ejecutando

Los evaluadores inspeccionan el software individualmente y sólo después de la evaluación pueden comunicarse sus hallazgos. Los evaluadores han de realizar informes por escrito. Una recomendación general es que los evaluadores naveguen a través de todo del sistema al menos dos veces para familiarizarse con su estructura y antes de comenzar con la evaluación propiamente dicha. Las sesiones de evaluación duran aproximadamente una o dos horas por página. Los evaluadores utilizan una lista de cotejo de criterios y cuando sea necesario, incorporarán nuevos principios a las categorías existentes a su lista de cotejo de criterios.

Los evaluadores no sólo elaboran una lista de problemas de usabilidad en el Sistema, sino que han de explicar los problemas de acuerdo con los principios de la usabilidad. El análisis de cada problema se ha de realizar por separado y no en conjunto. Se trata de evitar repetir los errores en el rediseño del software y permitir la solución de problemas concretos sin la necesidad de rediseñar por completo el Sistema.

4.10.3.3 Análisis

Una lista de los problemas identificados, que pueden ser priorizados considerando la gravedad o grado de criticidad de la seguridad que se produce. La jerarquización de la gravedad de los problemas del Sistema facilitará la aplicación posterior de políticas efectivas de rediseño del software. La gravedad de los problemas de usabilidad es medida por tres factores:

- La frecuencia con la que el problema ocurre, ¿es común o poco frecuente?
- El impacto del problema cuando sucede, ¿es fácil o difícil para los usuarios superarlo?
- La persistencia del problema, ¿el problema es resuelto la primera vez que se use el software o aparece repetidamente?

Este método ha sido implementado y divulgado por (Nielsen, usable information technology, 2009)

4.11 Ordenamiento de tarjetas (Card Sorting)

La estructura organizativa y el rotulado de un sistema software también deben diseñarse desde el modelo mental de los usuarios. Por ejemplo, la técnica del ordenamiento de tarjetas (card sorting) nos permite determinar de qué manera los usuarios clasificarían, agruparían y rotularían la información a incluir (Rosenfeld & Morville, 1998, pág. 30).

Este es un método para descubrir la estructura que los usuarios desean del sistema. El investigador escribe cada sentencia (que corresponde a información o proceso del sistema) en una tarjeta pequeña, realiza tantas copias tantas como usuarios representativos tenga. Se barajan los grupos de tarjetas, se entregan a los usuarios y se pide que de manera individual clasifiquen estas tarjetas en grupos de manera ordenada. Los resultados de las ordenaciones individuales se combinan y si es necesario se analizan estadísticamente (Hassan, Martín, & Hassan, 2004, págs. 50-60). Ver Figura 18.



Figura 18 Card Sorting muestra el Modelo Mental de Organizar la Información (Information & Design, 2005)

4.11.1 Beneficios

Si las personas que ordenan son representativas de la población de usuarios para los que la aplicación se está diseñando, entonces el resultado reflejará la estructura en la que los usuarios esperan que las ideas o conceptos deben ser presentados en el sistema

4.11.2 Método

Se recomienda que sean usuarios representativos y que no tengan comunicación durante la sesión de ordenación

4.11.3 Planificación

Se recolectan las declaraciones (funciones, procesos) que se desean analizar. Escribir cada declaración en una tarjeta separada o imprimir si ya están digitalizadas como producto de la fase de análisis, se numeran las tarjetas por la parte de atrás.

Preparar manteles individuales con categorías de declaraciones predefinidas que sirvan para colocar las pilas de tarjetas. Si no se tienen categorías predefinidas asegurar que existe un vacío grande en la mesa para que el usuario coloque y ordene sus pilas de tarjetas.

Reclutar usuarios que serán representativos de la población de usuarios y que. Es útil tener al menos seis informantes.

4.11.4 Ejecutando

Se deben barajar las cartas de cada juego para que todos los informantes no tengan la misma secuencia de las cartas o peor aún, que cada usuario pueda tener la misma decencia de la corrida previa, si la hubo. Los informantes reciben su juego de cartas, y luego ordenarlos en pilas sobre la mesa en manteles individuales, si se tiene una pre-categoría definida. Se puede dar una indicación aproximada del número de pilas de cartas que se esperan tener, para dar a los informantes a un entendimiento común del grado de análisis esperado.

Explicar que una pila debe tener información relacionada e indicar que se le debe poner un nombre a cada pila. Dar la indicación de que las pilas deben estar ordenadas espacialmente sobre la mesa.

4.11.5 Análisis

Si las agrupaciones del usuario son relativamente claras y sencillas, resumir las cartas que se suelen incluir en cada pila y dar un nombre general para el grupo. Sin embargo, el resultado de la tarjeta de clasificación no siempre es claro y sencillo.

El método más común de análisis de datos complejos de Card Sorting es un método estadístico denominado análisis de conglomerados. Existen dos enfoques principales para el análisis de conglomerados: la vinculación y jerárquica. A fin de prepararse para ambos tipos de análisis, construir una matriz de similitud. Si hay n tarjetas, la matriz es $n \times n$ simétrica. La mayoría de los métodos de exigir la parte superior e inferior cuadrantes por cubrir de forma redundante, y para las diagonales para ser rellenos con 1,00. En cada uno de los células, calcular la probabilidad de p de las tarjetas denotado por los coordenadas comunes de estar juntos, es decir, si dos cartas x y y fueron clasificadas en la misma pila q veces y hay m encuestados en total, $p = q / m$.

Algunos métodos de análisis de agrupaciones puede requerir un índice de disimilitud, para calcular $p' = 1 - p$ y poner 0,00 en las diagonales.

4.12 Prototipado Rápido

Prototipado Rápido se describe como un método basado en ordenador que pretende reducir las iteraciones en el ciclo de diseño. Habitualmente se desarrollan prototipos que son rápidamente reemplazados o modificados como consecuencia de los datos proporcionados por continuos experimentos. Efectivamente es pues un método característico del software (esta filosofía se plantea posible para el hardware, pero requiere más medios) y la participación del usuario se relega al test del prototipo. Dos casos particulares son:

- RAD (Rapid Application Development o Desarrollo Rápido de Aplicaciones)
- JAD (Joint Application Development o Desarrollo Conjunto de Aplicaciones)

4.12.1 Beneficios

Problemas de usabilidad potenciales pueden ser detectados en fases tempranas, antes de que haya código. Se promueve la comunicación entre diseñadores y usuarios. Sólo se requiere un mínimo de recursos y materiales necesarios. Los prototipos son rápidos de construir y permite iteraciones de diseño rápido.

4.12.2 Método

4.12.3 Planificación

Organizar un taller en el que asistirán el usuario y el desarrollador. Se necesitará un facilitador y una persona para registrar las cuestiones planteadas durante la reunión. Cuatro etapas de creación de prototipos de papel pueden ser necesarias:

- Diseño de concepto. Para explorar diferentes metáforas y el diseño de estrategias
- Diseño de interacción. Para organizar la estructura de las pantallas de o las páginas
- Diseño de la pantalla. Para el diseño inicial de la pantalla de cada individuo
- Pruebas de pantalla. Para perfeccionar el diseño de la pantalla

□

4.12.3.1 Ejecutando

- Diseño de concepto. Sentarse alrededor de una mesa y esbozar posibles enfoques en un entorno de intercambio de ideas, después evaluar la medida en que cada método se atine a la facilidad de uso de requisitos y objetivos acordados con los stakeholders.
- Diseño de interacción. Usar diagramas de afinidad para estructurar la interfaz de usuario. Escriba el nombre sugerido en cada pantalla o página en un papel para notas pegables, deben agruparse en clusters significativos para el usuario. Después consolidar los duplicados, de un nombre a cada grupo, documentar la secuencia en las que se ejecutarán las tareas de cada grupo.
- Diseño de la pantalla. Alrededor de una mesa haga un esbozo con una lluvia de ideas, utilizar el esbozo para crear las pantallas, ventanas o páginas. Generar vínculos entre pantallas, pedir al usuario llevar a cabo una tarea realista, analizar la secuencia, explique lo que sucede.
- Pruebas de pantalla. Con una herramienta para prototipos realizar un diseño en bruto de cada pantalla, generar vínculos entre las pantallas, preparar la interacción más detalladamente, preparar trozos de papel con menús, cuadros de desplazamiento, cuadros de dialogo, etcétera. El usuario debe simular la ejecución real de la interfaz.

4.13 Prueba de Usuarios (Test de usabilidad)

Hacer un test de usabilidad requiere de conocimientos, tiempo, dedicación y experiencia. Además, hay que conseguir a los usuarios correctos y destinar una parte del presupuesto para pagarles. Los test consisten básicamente en entregar una lista de tareas a los usuarios y observarlos o filmarlos mientras intentan completar cada una de las tareas sin nuestra ayuda. Es increíblemente enriquecedor observar a los usuarios frustrarse una y otra vez con nuestro diseño. Ningún diseño de interacción concebido sólo desde nuestra intuición o experiencia será infalible. Es por eso que es aconsejable comenzar las pruebas con prototipos de baja fidelidad (incluso sobre papel), en las primeras etapas del proceso de diseño.

4.14 Métricas de usabilidad

Aún cuando se tienen las pruebas de usabilidad relativas, se debe medir de manera más objetiva. Mientras que la interpretación de una opinión es un análisis cualitativo o subjetivo por parte del experto, la interpretación de datos objetivos responde a un análisis cuantitativo.

Este tipo de variables se estructuran, como norma general, en tres grandes grupos:

□ *Efectividad*: variables que nos permiten medir la exactitud y la plenitud con la que se alcanzan los objetivos de una tarea concreta. Algunas de las variables más típicas son:

- Porcentaje total de tareas completadas.
- Porcentaje de tareas completadas en el primer intento.
- Porcentaje de usuarios que completan las tareas.
- Ratio de éxitos y fracasos.
- Número de veces que los usuarios solicitan ayuda por no saber qué hacer.

□ *Eficiencia*: se refiere al esfuerzo que un usuario tiene que hacer para conseguir un objetivo. Algunas variables típicas son:

- Tiempo empleado en completar cada tarea.
- Porcentaje o número de errores cometidos.
- Porcentaje de errores o problemas según su severidad.
- Tiempo empleado en recuperarse de los errores.
- Número de clicks realizados para completar una tarea.
- Número de páginas o pantallas o ventanas visitadas para completar una tarea.
- Tiempo empleado en determinadas páginas o grupos de páginas.
- Porcentaje o número de veces que se acude a ayudas, FAQ o similar.

□ *Satisfacción*: se refiere a aquellas que tienen que ver más con lo emocional o subjetivo. Para medir el grado de satisfacción puedes utilizar criterios como:

- Porcentaje de usuarios que después de utilizar el producto lo recomendaría a un amigo.
- Proporción de adjetivos positivos o negativos que cada usuario de al producto.
- Porcentaje de usuarios que califican el producto más fácil de usar que cualquiera de la competencia directa.
- Número de veces que el usuario expresa satisfacción o insatisfacción.

Además de las reflejadas aquí, es posible crear cualquier otra métrica que resulte de utilidad. Por ejemplo, se puede medir el porcentaje de incremento de ventas que tienes antes y después de hacer un rediseño.

4.15 Aplicando las métricas

A diferencia de un test de usuarios, para medir las métricas es necesario tener una representación de usuarios algo mayor. Si para un test de usabilidad con 5 ó 6 usuarios es suficiente, para la medición de métricas es aconsejable contar con al menos unos 20 usuarios para obtener unos resultados más o menos fiables.

Antes de empezar tener en cuenta al menos, lo siguiente:

- Definir las tareas que los usuarios van a realizar. Éstas deben ser cuidadosamente seleccionadas y deben corresponderse con aquellos procesos que se deseen mejorar o sean críticos para el éxito del negocio. Lo normal es centrarse en aquellas tareas más críticas o complejas y que requieren de cierto tiempo y dificultad para completarlas.
- Establecer los objetivos para las tareas que se hayan seleccionado. Es decir, qué objetivo concreto debe alcanzar el usuario para considerar que ha completado la tarea con éxito.
- Definir las variables que van a medir y diseñar los cuestionarios de satisfacción.
- Planificar cómo se van a recoger los datos: existen herramientas como Morae que permiten recabar estos datos de manera casi automática y explotarlos de forma personalizada. Se puede hacer de forma manual, cronometrando los tiempos y recogiendo los datos en alguna plantilla diseñada para tal fin.

5 Metodología

Dado que uno de los objetivos es hacer posible la usabilidad en el software, primero se creó un modelo de medición de la usabilidad del producto software, que culminó en gráficas fáciles de interpretar, éstas son la salida del sistema CAME. La validación en un caso práctico de estudio, se hizo sobre un sistema ya implementado y accedido por toda la población Estudiantil de la Universidad Autónoma del Estado de México. Para ello se detallan las técnicas implementadas en la creación del modelo de medición de la usabilidad y del desarrollo del sistema CAME, nombrado *MEDUSA*, propuesto y ad hoc al modelo.

MEDUSA es un sistema con arquitectura simple de tal manera que se ejecute en arquitectura standalone ó en cliente-servidor; en ambos casos con requerimientos de infraestructura mínima, se utilizó un desarrollo ágil, por lo que la documentación del diseño es sólo la necesaria. Posteriormente para la aplicación de la medición en un caso práctico se realizaron las actividades de la selección del sistema a evaluar, selección de los usuarios representativos, selección de los evaluadores. Con éstos grupos se llevó a cabo dos sesiones de inspección: evaluación heurística y la prueba de Usuarios representativos, con ello se obtienen los resultados a analizar tanto de Evaluación Heurística y Métricas de usabilidad que son mostrados con *MEDUSA*. Se detallan la secuencia y orden de actividades a continuación.

5.1 Propuesta de modelo de medición de usabilidad

El modelo de medición fue diseñado para realizarse en sistemas implantados, después de un periodo corto de uso, con ello los usuarios podrán opinar con mayor seguridad acerca del sistema, lo que dará veracidad a los datos

En el capítulo anterior se estudiaron los métodos de evaluación de la evaluación de la usabilidad, para nuestro caso de estudio se aplicaron la *Evaluación Heurística* y *Consulta al Usuario Directo* en el Modelo de Medición que está representado en la figura 19.

Al observar el modelo, se puede identificar que las evaluaciones se llevan a cabo después de la fase de Lanzamiento, ya que se tiene como objetivo medir la usabilidad en Sistemas creados por organizaciones que fabrican su propio software. Así pues el modelo sirve para sistemas ya implantados. Los resultados obtenidos deben servir para retroalimentación para la fase de mantenimiento y el grado de éxito del sistema.

Las fases anteriores a la aplicación del modelo permanecen en la forma clásica, aunque esto no quiere decir que sólo aplica en cascada, como se ha detallado la evaluación en cuando el sistema ya está en uso y se desea saber su grado de uso y éxito, por lo que es independiente al modelo de desarrollo.

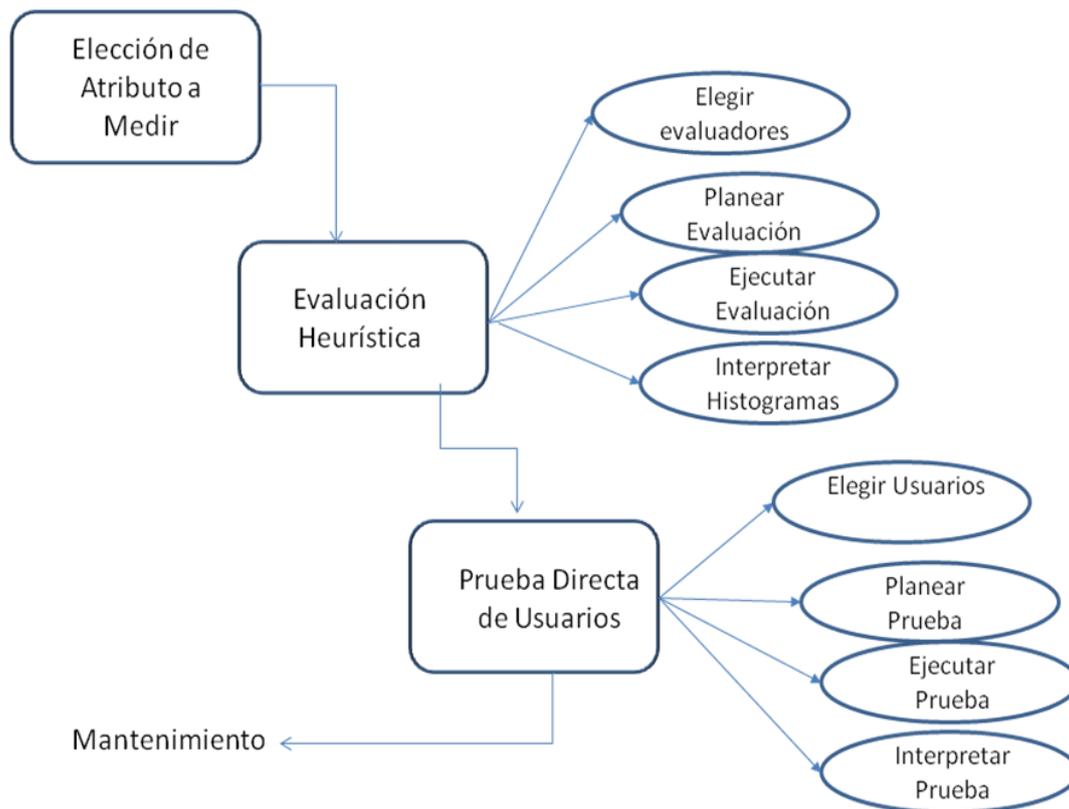


Figura 19 Modelo de Evaluación de la usabilidad (autoría propia)

5.1.1 Elección de Métodos de Evaluación

Los métodos de evaluación utilizados, se podrán aplicar en diferentes tipos de sistemas de las PYMES ya sea sitios web o sistemas internos. Los métodos de evaluación a emplear van a depender del tiempo, economía y experiencia de los evaluadores, pero en esta propuesta se realizan las recomendaciones necesarias para evitar estos inconvenientes cuando se reproduzca el modelo:

- El tiempo: Para planear la evaluación, elegir usuarios y elaboración de resultados. Para algunos métodos el tiempo es muy grande a comparación por ejemplo de *Consulta al Usuario Directo*.
- Dinero: Para Contratar a expertos en evaluaciones de usabilidad ó comprar software.
- Experiencia de los Evaluadores: si son novatos incluso tardaran en obtener resultados, o bien en investigar qué, como y cuándo van a evaluar.

Con base a los fundamentos dados en el capítulo anterior se opta por utilizar *Evaluación Heurística* y *Consulta Directa al usuario*. Otro método de inspección muy utilizado es la *Observación en Vivo* se elimina ya que es demasiado relativa y es más difícil de interpretar por parte de los usuario, ya que requiere habilidades más específicas. En esta investigación se requirió datos cuantitativos, por lo que es descartada. Algunas otras técnicas de

inspección o aplicación se realizan en la construcción del sistema, razón por la cual no fueron elegidos ya que el caso práctico usa un sistema ya en uso.

El benchmarking de los métodos de evaluación en la sección 4.9 permitió hacer fácil la elección de los métodos elegidos

5.2 Diseño del sistema *MEDUSA*

El sistema fue diseñado para la entrada de datos de la Evaluación Heurística y la Prueba de Usuarios, por lo que el siguiente caso de uso representa a grandes rasgos lo que debe hacer, ver Figura 20.

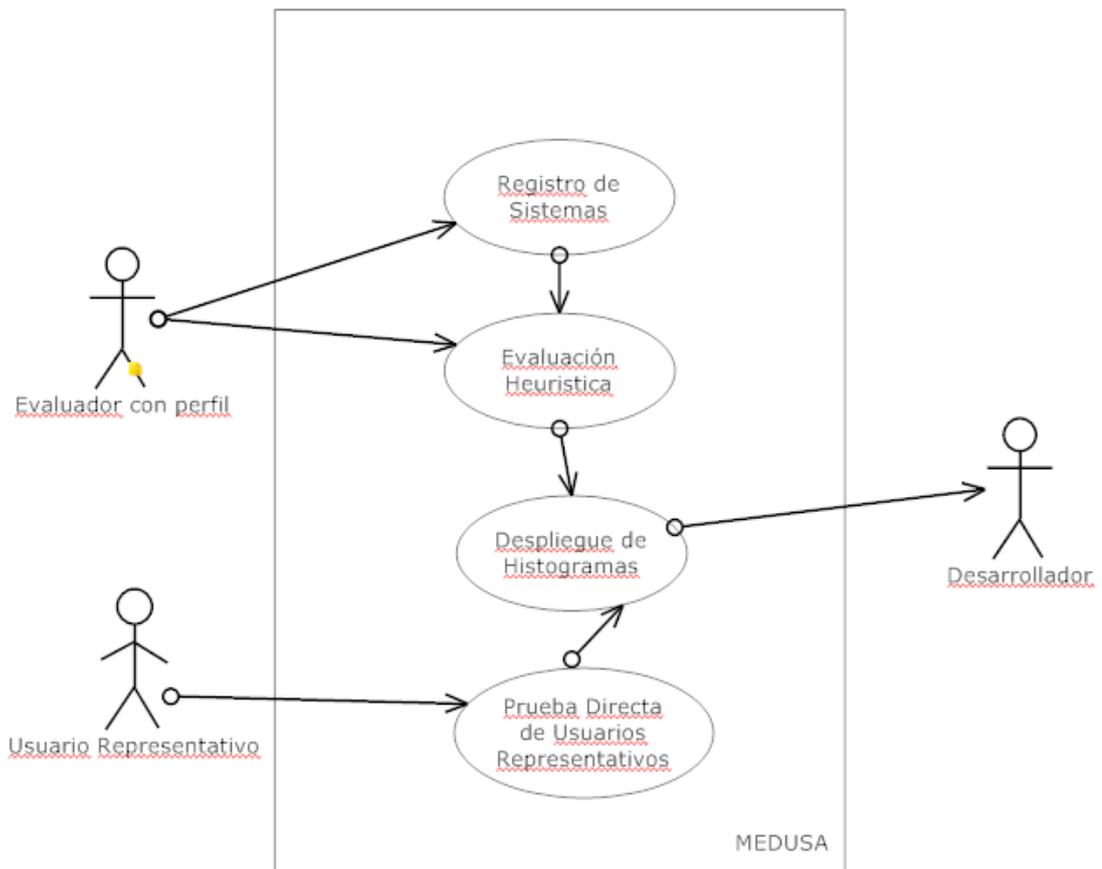


Figura 20 Caso de Uso de MEDUSA (autoría propia)

Para la inspección heurística se tomaron un grupo de preguntas de toda la tabla que proporcionada por Nielsen, se debe permitir la entrada de las respuestas dadas por varios evaluadores o diseñadores.

Para la obtención de Métricas se realizó un cuestionario de preguntas cerradas aplicables a los usuarios, de tal manera que se obtengan indicadores y gráficas para analizar. Toda la información correspondiente al desarrollo del software de Medición de usabilidad se encuentra en anexos para su consulta. El diseño de Base de Datos es el siguiente Figura 21

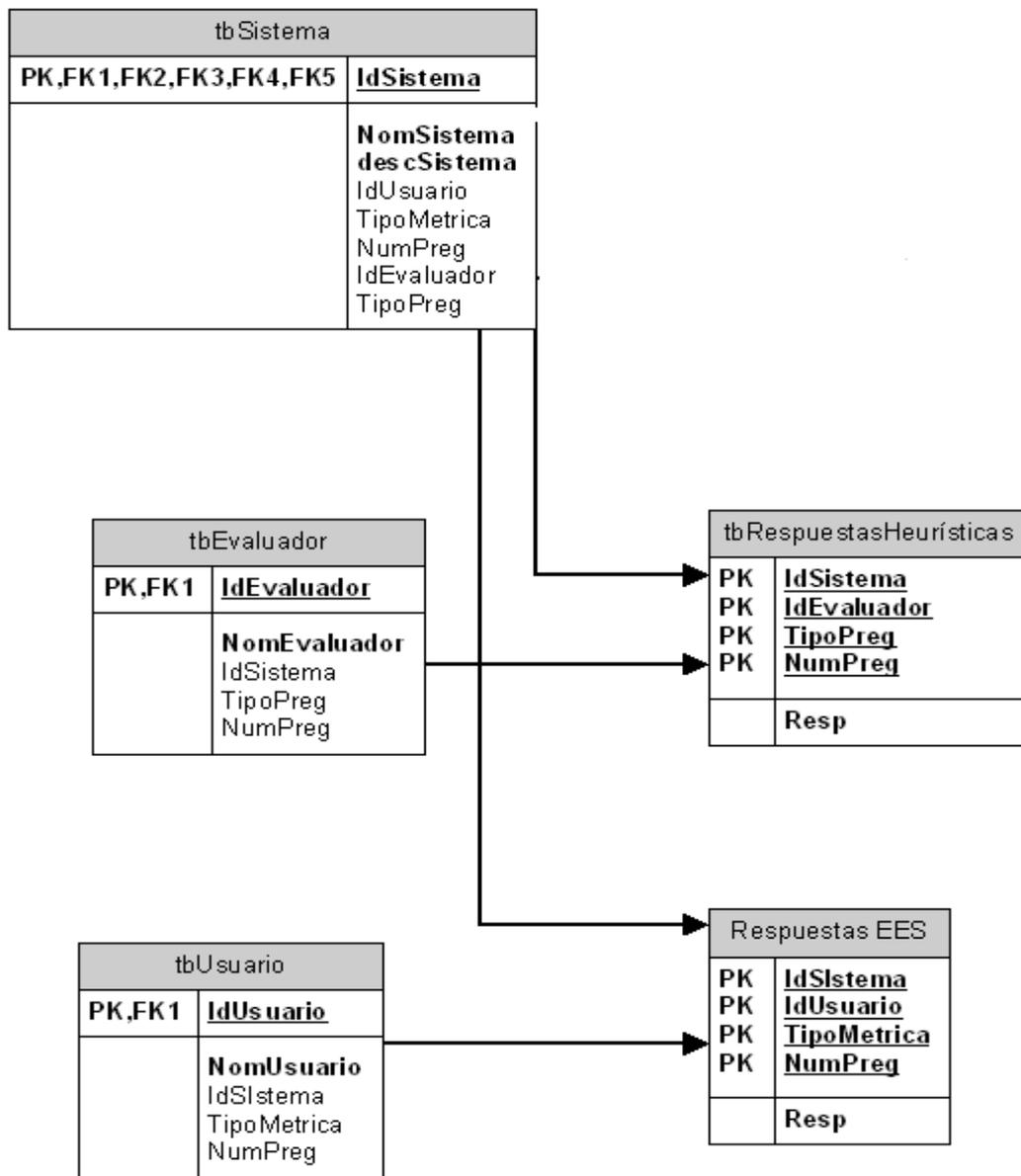


Figura 21 Diseño de Base de Datos para MEDUSA

5.3 Selección del Sistema

Para la elección del sistema a evaluar, fue claro que debía ser un sistema ya implantado e identificando su utilidad en los objetivos de la Institución. Dentro de la lista de sistemas a los que se tuvo acceso y que son para uso de la UAEM (Ya que desde el inicio se estableció que el modelo de medición era aplicable a organizaciones que producen su propio software de manera interna) sólo se encontró uno que es de gran importancia para la institución y al que se puede acceder fácilmente sin afectar datos. Además de estar al alcance de la autora del presente trabajo, para ello se optó por utilizar el sistema de tutorías de la Universidad Autónoma del Estado de México. Con el fin de proporcionar sugerencias a la institución para nuevas versiones del mismo. Es un sistema de gran apoyo al programa de tutorías que requiere de retroalimentación para la mejora continua, elemento importante para la calidad del software, Es mi aportación a la universidad de estudio, los fundamentos utilizados se muestran en la tabla 5.

Razones
Debe ser utilizado por la mayoría de la población Universitaria
La percepción de los alumnos es que se confunden
La comunicación vía su correo implementado es muy básico, y las direcciones destino no se pueden elegir a detalle
Los reportes tardan en salir
La presentación de datos tiene que observarse un lapso de tiempo largo para que sea útil

Tabla 5 Fundamentos de Elección del Sistema

Con el fin de insertar mecanismo de control de la Calidad en la institución, los resultados deben servir para:

- Verificar la existencia de posibles problemas de usabilidad en el Sistema de Tutorías.
- Encontrar posibles soluciones para los problemas encontrados.
- Establecer las bases para futuros desarrollos de este mismo sistema o modificaciones al actual.

5.4 Evaluación Heurística

5.4.1 Variables Utilizadas: Principios Heurísticos

Como se explicó en la sección 4.13 existen principios heurísticos acoplados a los sistemas actuales, se hace un análisis de las propiedades para poder identificar como se va a medir.

1. Visibilidad del Contexto. El usuario debe saber exactamente qué es lo que está haciendo el sistema.

2. Coincidencia entre el sistema y el mundo real. El sistema deberá expresarse en el lenguaje del usuario, con palabras, frases y conceptos que le sean familiares, cuidándose de no hacerlo con términos propios del sistema informático. *Lenguaje Común.*

3. Libertad y Control por parte del usuario. Mínima cantidad posible de restricciones a los usuarios, permitiéndoles elegir los caminos y las formas de cumplir sus objetivos. Evitar desactivar los controles del navegador. *Estado del Sistema, Controles de Despliegue, Deshacer Acciones*

4. Consistencia y Estándares. Los usuarios no deben tener necesidad de discernir si palabras, situaciones o acciones distintas significan lo mismo. *Diseño de Interfaz, uso de estándares*

5. Prevención de Errores. Mejor que buenos mensajes de error es un diseño cuidadoso que anticipa y previene la ocurrencia de los problemas. *Mensajes de error prevención, Recuperación de errores, deshacer acciones.*

6. Reconocer es mejor que recordar. Minimizar la carga en la memoria del usuario haciendo los objetos, acciones y opciones visibles. *Intuitivo, Predecible, Técnica paso a paso.*

7. Flexibilidad y eficiencia de uso. Minimizar el esfuerzo requerido para alcanzar los objetivos, acortar páginas o ventanas y procesos innecesarios. *Intuitivo, Número de ventanas ó páginas visitadas para realizar una acción.*

8. Diseño minimalista y estética. Las páginas no deben contener información que sea irrelevante o remotamente necesaria. *Interfaz had hoc a la plataforma.*

9. Escribir para el Usuario. Los textos y descripciones deben estar optimizados para el sistema desde el punto de vista de los usuarios. *Lenguaje Común, guías eficientes.*

10. Ayudar a los usuarios a reconocer, diagnosticar y recuperarse de los errores. Los mensajes de error deben expresarse en lenguaje habitual, indicar con precisión el problema y sugerir constructivamente una solución. *Prevención de errores, recuperación de errores, mensajes de error, deshacer acciones*

5.4.2 Rangos de Severidad

En las pruebas heurísticas se pidió la opinión de los expertos acerca de del sistema, se utilizó una prueba de actitud, en este caso la escala de Linkert método muy común para este tipo de evaluaciones. Se muestra la tabla 6 con la escala ordinal de rango a utilizar, como se pretendió categorizar la respuesta para dar indicios importancia de la severidad, entonces se implementó como sigue:

Escala	Escala	Fundamento
Severidad 0	No es un problema	No se identifica Como Problema
Severidad 1	Sólo es Cosmético	Se puede componer cuando haya tiempo
Severidad 2	No grave	Se compone catalogando con prioridad baja
Severidad 3	Grave	Se compone catalogando con prioridad alta
Severidad 4	Catastrófico	Es prioritario sea compuesto

Tabla 6 Rangos de Severidad. Elaboración propia

La descripción de las escalas de severidad expresan mayor conocimiento del tema, por lo que los evaluadores debieron analizar cuidadosamente la respuesta adecuada, de acuerdo a su experiencia en el sistema. Con el rango anterior se pudo detallar los puntos en los que el sistema se aparta de los atributos de usabilidad estudiados en la sección 4.13.

5.4.3 Indicadores

Los indicadores a utilizar de acuerdo a la herramienta de medición es la siguiente:

$$f_i^{Pregunta\ n} = \sum_{1}^{Evaluador} R_{valor=i}$$

5.4.4 Selección de evaluadores

De acuerdo a la sección 4.13.1 el perfil del evaluador es que debe ser un Ingeniero en usabilidad. El número de evaluadores depende del costo beneficio, el cual es bajo cuando los evaluadores son empleados. Sin embargo los resultados de 2 evaluadores realmente expertos, dan resultados parecidos a un grupo de 3 a 5 evaluadores simples. Las empresas a las que está enfocada la propuesta no manejan ésta perfil de Ingenieros en usabilidad, por lo se recomienda que sean al menos 3 evaluadores simples, y también se establecieron las siguientes recomendaciones para asegurarse que el evaluador tenga conocimientos necesarios para realizar la su trabajo:

- Experiencia en análisis y diseño usable, para compensar la ausencia de ésta característica se elaboró una guía del evaluador.
- Haber participado en la implementación de al menos 3 proyectos, por lo menos pequeños.

El grupo de evaluadores estará formado por profesionistas expertos en diseño que por fortuna se encuentran en la institución dentro de la plantilla docente. Todos deben de haber realizado o asesorado sistemas ya que con ello se pretende un apoyo bilateral.

Se consideraron 3 evaluadores que cumplen las características anteriores, además de haber impartido las asignaturas de análisis y diseño de sistemas en las licenciaturas de Informática Administrativa e Ingeniería en Computación:

- Evaluador 1: M.C.C Sergio Ruiz Castilla
- Evaluador 2: M.C.C Gumecindo Flores Varilla
- Evaluador 3: Leticia Arévalo Cedillo

5.4.5 Preparando los recursos de la Evaluación

A los evaluadores se les hizo entrega de las guías necesarias para realizar su trabajo, ésta se fundamenta en los trabajos realizados por explicada en la sección 4.13.1. La documentación corresponde a las siguientes fases de la evaluación:

- Guía del Evaluador: Como su nombre lo indica es una referencia rápida para el proceso en cuestión, Para alentar a llevar a cabo este proceso, se indica de manera clara y breve las fases de evaluación heurística. Ver anexos.
- Lista de Cotejo de Evaluación Heurística: Es el documento que el evaluador utilizó para llevar a cabo su trabajo. Se presenta la lista completa proporcionada por Nielsen. Ver Anexos.

Antes de la sesión de evaluación los evaluadores hicieron dos navegaciones sobre todo el sistema de tutorías para conocerlo un poco más. Durante la sesión de evaluación no hubo comunicación entre los evaluadores, el tiempo de duración es de 3 horas.

5.4.6 Cuestionario a Aplicar

Para la creación del cuestionario se consideraron los principios heurísticos como base para redactar las preguntas se consideró que su aplicación es mediante MEDUSA, por lo que la redacción debe ser mucho más fácil sin introducir complejidades de lenguaje o distracciones visuales en el sistema. Se hizo un primer borrador que fue sometido a crítica, por parte del Dr. Alfonso Zarco Hidalgo, cuyos estudios están enfocados al área de Estadística. Con ello se pretendió eliminar errores de redacción y consistencia en las preguntas. La tabla 7 es la versión final del cuestionario.

PREGUNTAS
Visibilidad en el Contexto
1. Algunas veces no se qué debo hacer después con éste software
2. Requiero ingresar los mismos datos continuamente
3. A veces me pregunto si estoy utilizando la función correcta
4. Sé, exactamente, cuando se está realizando un proceso del Sistema
5. Algunos Títulos de Páginas o Ventanas no corresponden exactamente a lo que se hace en ella
Coincidencia entre el Sistema y el Mundo Real
6. Todo el lenguaje utilizado me es conocido, corresponde al lenguaje utilizado en los procesos manuales
7. Algunas veces el formato utilizado tanto en entradas como salidas no es el mismo a lo que llevo en papel
8. Puedo comprender y actuar con base en la información que me proporciona el sistema
9. El software se ha detenido alguna vez inesperadamente
10. Las tareas pueden hacerse de una manera directa con el sistema
Libertad y Control por Parte del Usuario
11. Puedo ingresar a las funciones en cualquier momento
12. La barra de desplazamiento existe y me es útil
13. Tener muchas ventanas o páginas desplegadas en el sistema, no es un problema
14. Me siento al mando de éste software cuando lo uso, es fácil hacer lo que deseo
15. Puedo salir de una Función a la que entré por error fácilmente
Consistencia y Cumplimiento de estándares
16. Rompe con la manera en que trabajo en otras aplicaciones en la Computadora
17. Es raro cuando quiero hacer algo estándar
18. La organización del sistema es lógica, me lleva directamente a lo que quiero hacer
19. Tanto la cantidad como la calidad de la información de apoyo varían a través del sistema
20. Es armonioso el diseño
Prevención de Errores
21. La organización de los menús es lógica
22. Tengo que recurrir frecuentemente a las guías
23. Las tareas pueden hacerse fácilmente
24. Usar este software es frustrante
25. Creo que este software es inconsistente
Reconocer es mejor que recordar
26. Aprender nuevas funciones es difícil
27. Tengo que recurrir a ayuda algunas veces
28. Prefiero quedarme con las funciones que conozco
29. No me gustaría recomendar éste software
Flexibilidad y Eficiencia de Uso
30. Se tienen teclas rápidas para acceder a las funciones
31. Puedo personalizar la aplicación de acuerdo a mis preferencias de uso
32. Nunca aprenderé lo que se ofrece en este software
33. Es fácil hacer que el software haga exactamente lo que yo quiero
34. La velocidad de desempeño es buena en todas las funciones
35. Hay muchos pasos requeridos para hacer algo
Diseño minimalista y Estética
36. El software se presenta de una manera muy atractiva
37. Todas las imágenes presentadas me resultan intuitivas
38. En algunas ventanas o sitios se encuentra saturado de información
39. Las imágenes indican claramente lo que se va a realizar
40. Toda la Información presentada me es útil

Escribir para los usuarios
41. Utilizar el sistema es estimulante
42. Las instrucciones han sido útiles
43. Me es común utilizarlo, estoy acostumbrado al diseño
44. La documentación del software es clara y concisa
45. En cada momento sé lo que estoy haciendo
Ayudar a los usuarios a reconocer, diagnosticar y recuperarse de los errores
46. Los mensajes de error no son adecuados
47. El sistema me ha ayudado a superar todos los problemas que me he tenido al utilizarlo
48. Cuando voy a eliminar datos el sistema me advierte
49. Si ocurre un error me comunica lo que hice mal u omití
50. Aunque tenga errores puedo seguir trabajando sin afectar la información

Tabla 7 Cuestionario de evaluación Heurística

5.4.7 Práctica de la Elaboración

Son tablas que indican por heurística los problemas encontrados en el sistema, a continuación se presentan:

5.5 Prueba de Usuarios

Aunque existen otros métodos de realizar pruebas de usabilidad por parte de los usuarios, ya en el capítulo anterior se pudo apreciar algunas ventajas y desventajas los métodos, por lo que se optó por realizar el método de *Pruebas con los Usuarios*. Esta prueba se realiza para conocer el punto de vista de los usuarios representativos del sistema, sugiere realizar con cuestionarios, por lo que es la opción elegida para hacer una implementación en *MEDUSA*, y se puedan obtener resultados cuantitativos.

5.5.1 Cuestionario a aplicar

Con el fin de realizar comparación de resultados de las evaluaciones implementadas, las preguntas se crean con base a las heurísticas mencionadas anteriormente, pero el enfoque es hacia los usuarios, por lo que las preguntas serían enfocadas a los siguientes criterios, desde el punto de vista usuario y no de experto en diseño, como en la sección anterior:

- Facilidad de aprendizaje
- Velocidad de desempeño
- Tasas de error por parte de los usuarios
- Retención en el tiempo
- Satisfacción subjetiva.

Con base estos atributos se plantean las siguientes preguntas en la tabla 8:

ATRIBUTO	PREGUNTAS
Visibilidad del Contexto	1. Algunas veces no se qué debo hacer después con éste software
	2. Requiero ingresar los mismos datos continuamente
	3. A veces me pregunto si estoy utilizando la función correcta
	4. Sé, exactamente, cuando se está realizando un proceso del Sistema
	5. Algunos Títulos de Páginas o Ventanas no corresponden exactamente a lo que se hace en ella
Coincidencia entre el Sistema y el mundo Real	6. Todo el lenguaje utilizado me es conocido, corresponde al lenguaje utilizado en los procesos manuales
	7. Algunas veces el formato utilizado tanto en entradas como salidas no es el mismo a lo que llevo en papel
	8. Puedo comprender y actuar con base en la información

	que me proporciona el sistema
	9. El software se ha detenido alguna vez inesperadamente
	10. Las tareas pueden hacerse de una manera directa con el sistema
Libertad y Control por Parte del Usuario	11. Puedo ingresar a las funciones en cualquier momento
	12. La barra de desplazamiento existe y me es útil
	13. Tener muchas ventanas o páginas desplegadas en el sistema, no es un problema
	14. Me siento al mando de éste software cuando lo uso, es fácil hacer lo que deseo
	15. Puedo salir de una Función a la que entré por error fácilmente
Consistencia y Cumplimiento de estándares	16. Rompe con la manera en que trabajo en otras aplicaciones en la Computadora
	17. Es raro cuando quiero hacer algo estándar
	18. La organización del sistema es lógica, me lleva directamente a lo que quiero hacer
	19. Tanto la cantidad como la calidad de la información de apoyo varían a través del sistema
	20. Es armonioso el diseño
Prevención de Errores	21. La organización de los menús es lógica
	22. Tengo que recurrir frecuentemente a las guías
	23. Las tareas pueden hacerse fácilmente
	24. Usar este software es frustrante
	25. Creo que este software es inconsistente
Reconocer es mejor que recordar	26. Aprender nuevas funciones es difícil
	27. Tengo que recurrir a ayuda algunas veces
	28. Prefiero quedarme con las funciones que conozco
	29. No me gustaría
Flexibilidad y eficiencia de Uso	30. Se tienen teclas rápidas para acceder a las funciones
	31. Puedo personalizar la aplicación de acuerdo a mis preferencias de uso
	32. Nunca aprenderé lo que se ofrece en este software
	33. Es fácil hacer que el software haga exactamente lo que yo quiero
	34. La velocidad de desempeño es buena en todas las funciones
	35. Hay muchos pasos requeridos para hacer algo
Diseño minimalista y Estética	36. El software se presenta de una manera muy atractiva
	37. Todas las imágenes presentadas me resultan intuitivas
	38. En algunas ventanas o sitios se encuentra saturado de información
	39. Las imágenes indican claramente lo que se va a realizar
	40. Toda la Información presentada me es útil
Escribir para los usuarios	41. Utilizar el sistema es estimulante
	42. Las instrucciones han sido útiles
	43. Me es común utilizarlo, estoy acostumbrado al diseño
	44. La documentación del software es clara y concisa
	45. En cada momento sé lo que estoy haciendo
Ayudar a los usuarios a reconocer, diagnosticar y recuperarse de los errores	46. Los mensajes de error no son adecuados
	47. El sistema me ha ayudado a superar todos los problemas que me he tenido al utilizarlo
	48. Cuando voy a eliminar datos el sistema me advierte
	49. Si ocurre un error me comunica lo que hice mal u omití

	50. Aunque tenga errores puedo seguir trabajando sin afectar la información

Tabla 8 Preguntas Heurísticas

5.5.2 Rango de Severidad

De igual manera que la evaluación heurística, se va a utilizar una prueba de actitud, por lo que se utilizan respuestas cerradas con grado de severidad y acordes a los usuarios:

Escala	Escala	Fundamento
Severidad 0	Totalmente de Acuerdo	No se identifica Como Problema
Severidad 1	De acuerdo	Se puede componer cuando haya tiempo
Severidad 2	Puede ser	Se compone catalogando con prioridad baja
Severidad 3	En desacuerdo	Se compone catalogando con prioridad alta
Severidad 4	Totalmente en desacuerdo	Es prioritario sea compuesto

En la prueba Heurística los resultados de severidad determinan si se deben realizar más pruebas o no. Con el rango anterior se pudo detallar los puntos en los que el sistema se aparta de los atributos de usabilidad estudiados en la sección 4.13

5.5.3 Indicadores a Utilizar

$$f_i^{Pregunta\ n} = \sum_1^{no.usuarios} R_{rango=i}$$

5.5.4 Selección de Usuarios

El grupo utilizado para la prueba de usuarios será un conjunto de 20 usuario representativos dentro de los cuales serán 15 alumnos y 5 profesores tutores. Específicamente se estará el grupo estará compuesto de alumnos de de diferentes semestres para diversificar más la opinión. El universo de población son alumnos y profesores, nos centramos en aquellos alumnos que tienen asignado un tutor y a aquellos profesores que son tutores. Para cerrar más el círculo se seleccionó a aquellos profesores tutores activos y con más accesos al sistema en el semestre pasado (información que proporcione el

sistema), de la misma forma a los alumnos que realmente entran al sistema en el penúltimo semestre.

5.5.5 Preparando Recursos de la Evaluación

A los alumnos se les hizo entrega de las guías necesarias para realizar su trabajo, se les pidió no comunicarse entre sí y realizar una inspección de 20 minutos al sistema SITA y después que respondieran el cuestionario en cuestión.

5.5.6 Proceso de Datos y Resultados

Dado que se hizo la aplicación del cuestionario a usuarios, se tienen 50 respuestas concernientes a la usabilidad por cada Usuario, para analizar esta información se implementaron en el sistema sentencias SQL que resumieran la información de cada pregunta de acuerdo a los rangos preestablecidos, por lo que matemáticamente podemos describir, que para cada pregunta se aplicaron las siguientes fórmulas:

$$f_1^{Preg 1} = \sum_1^{no.usuarios} R_{valor=1}$$

$$U_1$$

$$f_2^{Preg 1} = \sum_1^{no.usuarios} R_{valor=2}$$

$$U_2$$

$$f_3^{Preg 1} = \sum_1^{no.usuarios} R_{valor=3}$$

$$U_3$$

$$f_4^{Preg 1} = \sum_1^{no.usuarios} R_{valor=4}$$

$$U_4$$

$$f_5^{Preg 1} = \sum_1^{no.usuarios} R_{rango=5}$$

De esta manera se obtuvo la frecuencia de cada valor del rango (totalmente en desacuerdo, en desacuerdo, tal vez, de acuerdo, totalmente de acuerdo) establecido para ésta prueba y analizar estos resultados.

Para la traducción de la fórmula matemática en la base de datos se aplicó la siguiente sentencia SQL, para la pregunta 1 que se encuentra en la tabla de VisibilidadContexto:

```
SELECT 1, "Totalmente en Desacuerdo", COUNT(Pregunta1)
```

```
FROM VisibilidadContexto
```

```
WHERE Pregunta1=1
```

```
UNION
```

```
SELECT 2, "Totalmente en Desacuerdo", COUNT(Pregunta1)
```

```
FROM VisibilidadContexto
```

```
WHERE Pregunta1=2
```

```
UNION
```

```
SELECT 3, "Totalmente en Desacuerdo", COUNT(Pregunta1)
```

```
FROM VisibilidadContexto
```

```
WHERE Pregunta1=3
```

```
UNION
```

```
SELECT 4, "Totalmente en Desacuerdo", COUNT(Pregunta1)
```

```
FROM VisibilidadContexto
```

```
WHERE Pregunta1=4
```

```
UNION
```

```
SELECT 5, "Totalmente en Desacuerdo", COUNT(Pregunta1)
```

```
FROM VisibilidadContexto
```

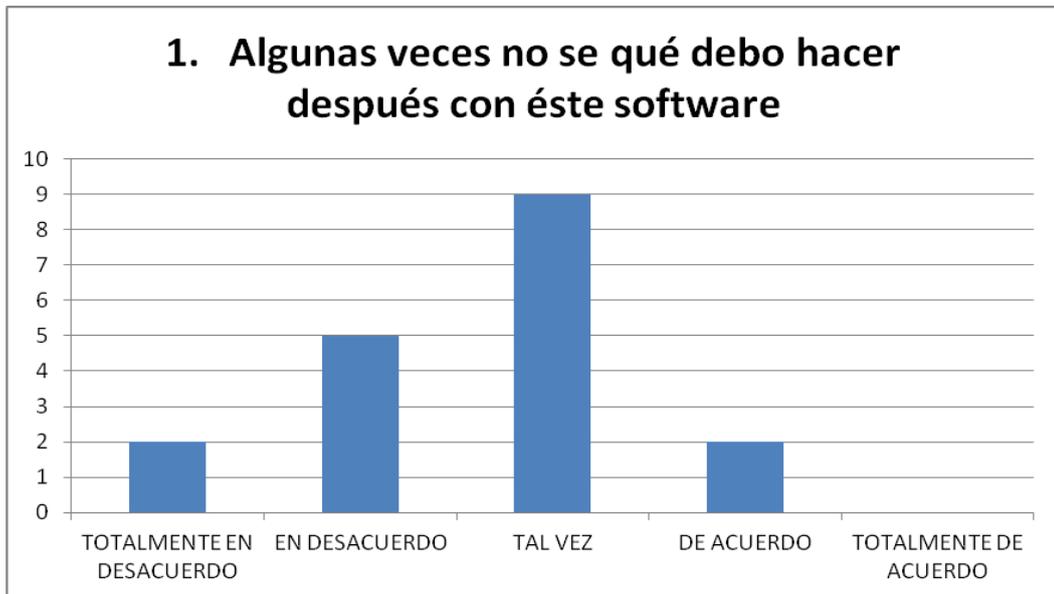
```
WHERE Pregunta1=5
```

Es importante resaltar que éste método se aplicó a las 50 preguntas que conforman el Cuestionario de usabilidad, más sin embargo se detalla sólo para una pregunta, para las demás se presentan las sentencias en el Anexo B, porque aplica idéntico proceso, obviamente los resultados son diferentes para cada pregunta.

Con la sentencia SQL se obtuvo la siguiente información de la pregunta 1 y posteriormente es graficada

1. Algunas veces no se qué debo hacer después con éste software	
RANGO	FRECUENCIA
TOTALMENTE EN DESACUERDO	2
EN DESACUERDO	5
TAL VEZ	9
DE ACUERDO	2
TOTALMENTE DE ACUERDO	0

Tabla 9 Frecuencias para Pregunta 1 del cuestionario



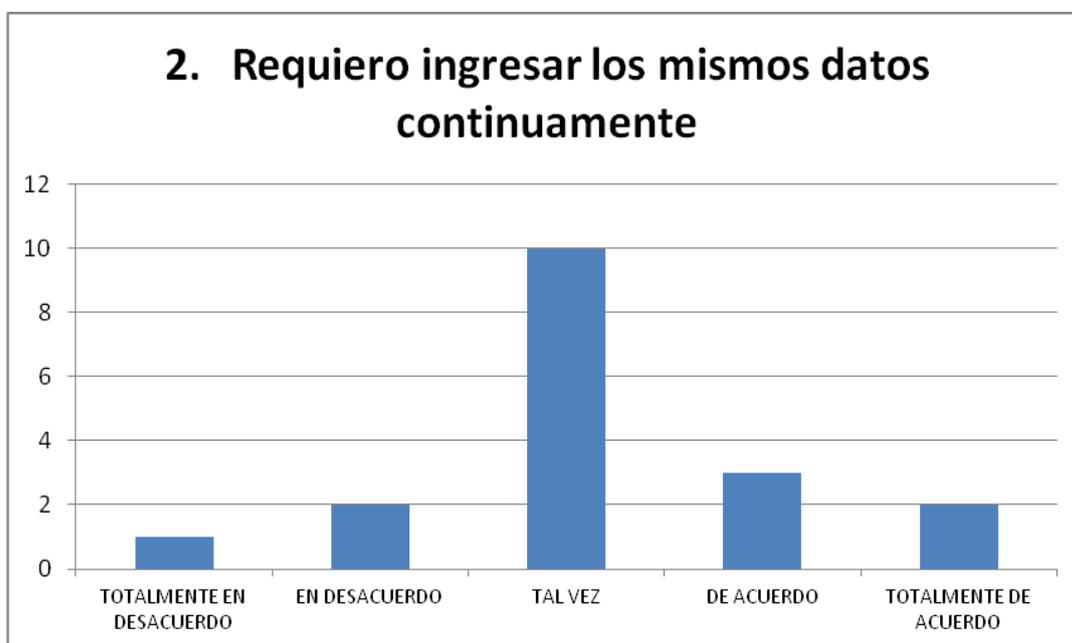
Gráfica 1 Frecuencias para Pregunta 1 del cuestionario (autoría propia)

El valor ideal para esta pregunta es pregunta es totalmente en desacuerdo, como se puede observar en la gráfica la tendencia es a generar ambigüedad en cuanto al conocimiento de funcionalidad del sistema.

A continuación se presentan los resultados de el Criterio *Visibilidad en el contexto*, que son las primeras 5 preguntas del cuestionario, los demás criterios se presentan resumidos en la siguiente sección, las nueve secciones restantes se presentan en el anexo C

2. Requero ingresar los mismos datos continuamente	
RANGO	FRECUENCIA
TOTALMENTE EN DESACUERDO	1
EN DESACUERDO	2
TAL VEZ	10
DE ACUERDO	3
TOTALMENTE DE ACUERDO	2

Tabla 10 Frecuencias para Pregunta 2 del cuestionario



Gráfica 2 Frecuencias para Pregunta 2 del cuestionario (autoría propia)

El valor ideal para esta pregunta es *totalmente en desacuerdo*, como se puede observar en la gráfica la tendencia es a generar ambigüedad en cuanto al acceso directo a las funciones.

3. A veces me pregunto si estoy utilizando la función correctamente	
RANGO	FRECUENCIA
TOTALMENTE EN DESACUERDO	0
EN DESACUERDO	1
TAL VEZ	10
DE ACUERDO	6
TOTALMENTE DE ACUERDO	1

Tabla 11 Frecuencias para Pregunta 3 del cuestionario

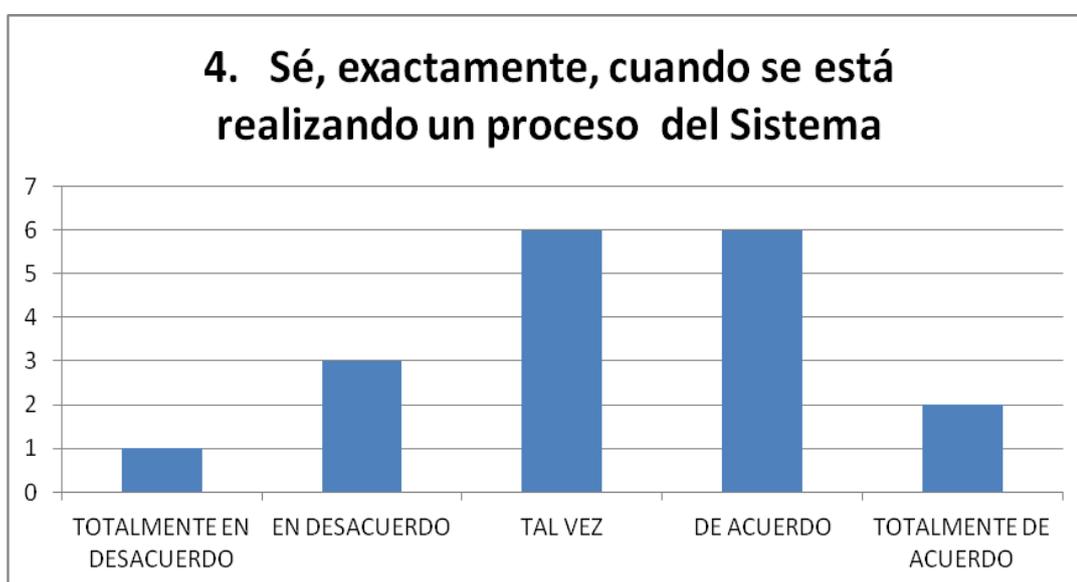


Gráfica 3 Frecuencias para Pregunta 3 del cuestionario (autoría propia)

El valor ideal para esta pregunta es *totalmente en desacuerdo*, como se puede observar en la gráfica la tendencia es a generar ambigüedad en cuanto al acceso directo a las funciones, que confirma las respuestas de la pregunta anterior.

4. Sé, exactamente, cuando se está realizando un proceso del Sistema	
RANGO	FRECUENCIA
TOTALMENTE EN DESACUERDO	1
EN DESACUERDO	3
TAL VEZ	6
DE ACUERDO	6
TOTALMENTE DE ACUERDO	2

Tabla 12 Frecuencias para Pregunta4 del cuestionario

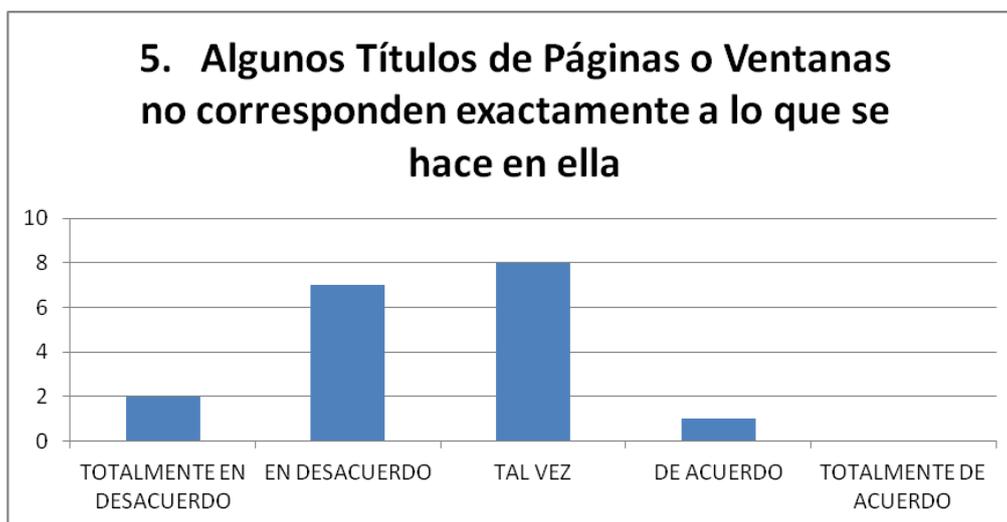


Gráfica 4 Frecuencias para Pregunta 4 del cuestionario (autoría propia)

El valor ideal para esta pregunta es pregunta es *totalmente en acuerdo*, como se puede observar en la gráfica la tendencia es a generar ambigüedad en cuanto lo que está haciendo el sistema en cierto momento, es decir retroalimentación, Para una usabilidad idónea la frecuencia alta debería estar concentrada en *Totalmente de acuerdo*.

5. Algunos Títulos de Páginas o Ventanas no corresponden exactamente a lo que se hace en ella	
RANGO	FRECUENCIA
TOTALMENTE EN DESACUERDO	2
EN DESACUERDO	7
TAL VEZ	8
DE ACUERDO	1
TOTALMENTE DE ACUERDO	0

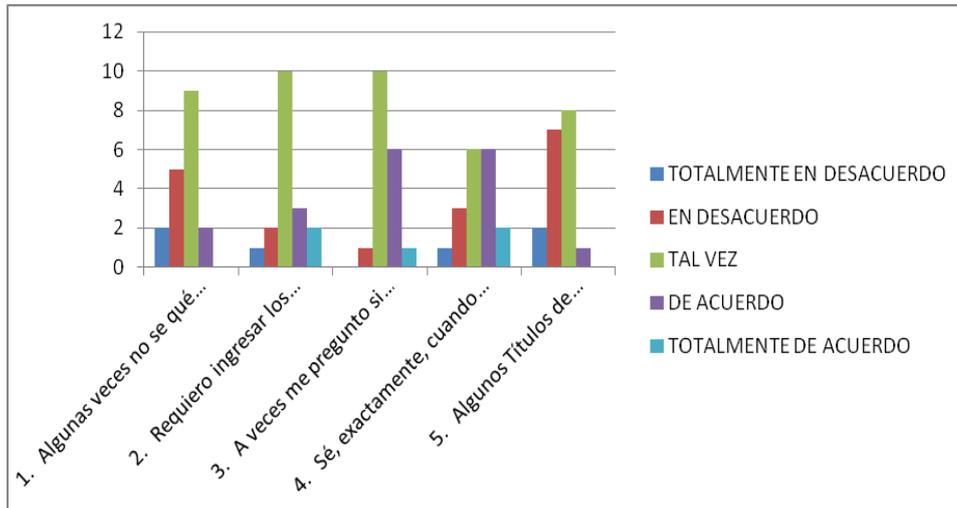
Tabla 13 Frecuencias para Pregunta4 del cuestionario



Gráfica 5 Frecuencias para Pregunta5 del cuestionario

El valor ideal para esta pregunta es pregunta es *totalmente en desacuerdo*, como se puede observar en la gráfica la tendencia es a generar ambigüedad en cuanto las descripciones o textos en el sistema, Para una usabilidad idónea la frecuencia alta debería estar concentrada en *Totalmente en desacuerdo*.

La gráfica que contempla las cinco preguntas de Visibilidad en el Contexto es:



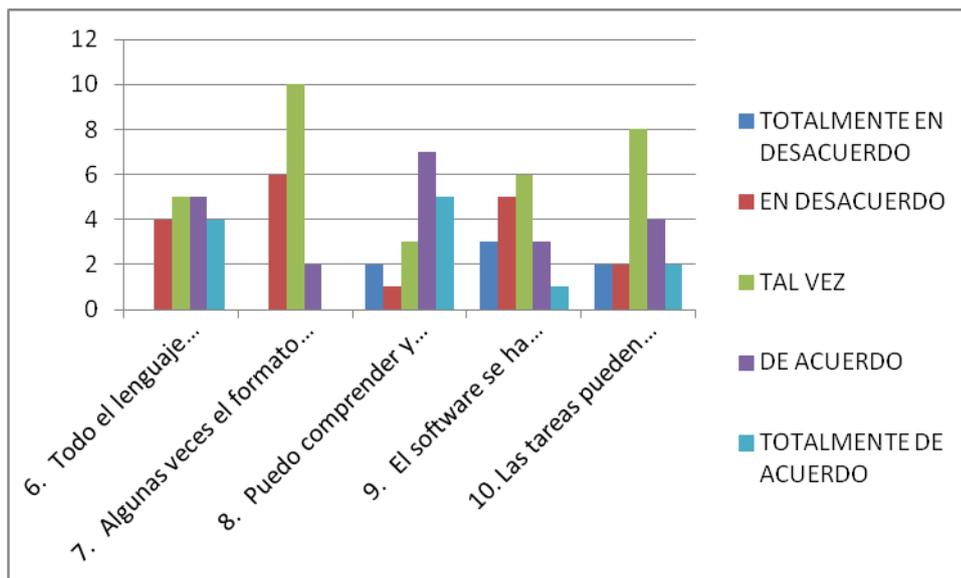
Gráfica 6 Frecuencias para las Preguntas de Visibilidad en el Contexto del cuestionario

Visibilidad del Contexto. El usuario debe saber exactamente qué es lo que está haciendo el sistema. La gráfica muestra el comportamiento de las respuestas proporcionadas para visibilidad en el contexto. Como se explicó anteriormente los valores idóneos de las preguntas caen en los extremos, por lo que la gráfica debería tener sus más altas frecuencias en éstos. Sin embargo prevalece una percepción muy ambigua acerca de lo que está haciendo el usuario en cierto momento.

PREGUNTA	VALOR IDÓNEO	RANGO
1. Algunas veces no se qué debo hacer después con éste software	Totalmente en desacuerdo	2
2. Requiero ingresar los mismos datos continuamente	Totalmente en desacuerdo	5
3. A veces me pregunto si estoy utilizando la función correcta	Totalmente de acuerdo	9
4. Sé, exactamente, cuando se está realizando un proceso del Sistema	Totalmente en desacuerdo	2
5. Algunos Títulos de Páginas o Ventanas no corresponden exactamente a lo que se hace en ella	Totalmente de acuerdo	0

Tabla 14 Visibilidad en el contexto

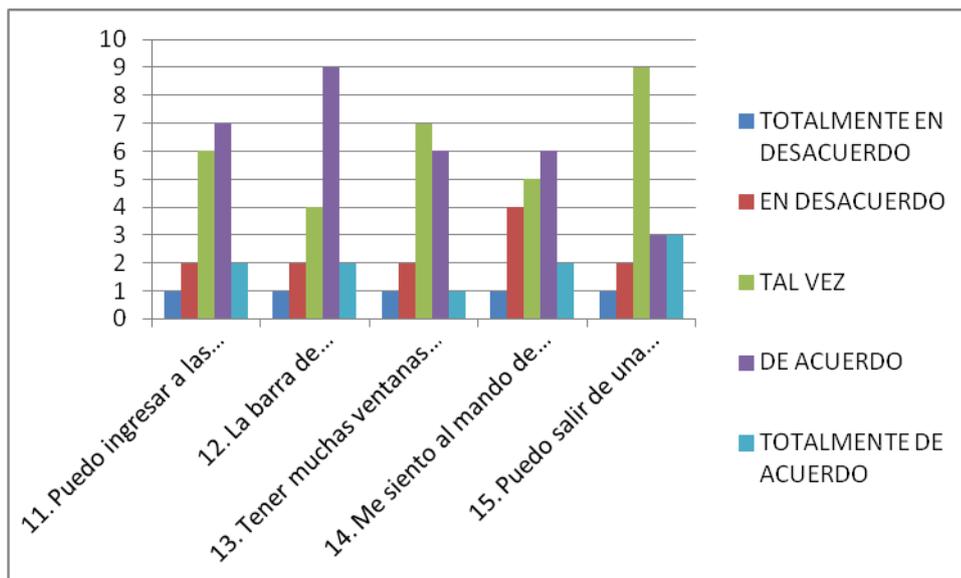
La gráfica que contempla las cinco preguntas de Coincidencia entre el sistema y el mundo real es:



Gráfica 7 de Coincidencia entre el Sistema y El Mundo Real del cuestionario

Coincidencia entre el sistema y el mundo real. Como se puede observar en la gráfica en cuanto al lenguaje utilizado en el mundo académico de la UAEM en el aspecto de tutorías no coincide al 100% con el lenguaje del Sistema, el que las barras de *tal vez* concentren la mayor parte de usuarios, se interpreta como no coincidencias en el lenguaje.

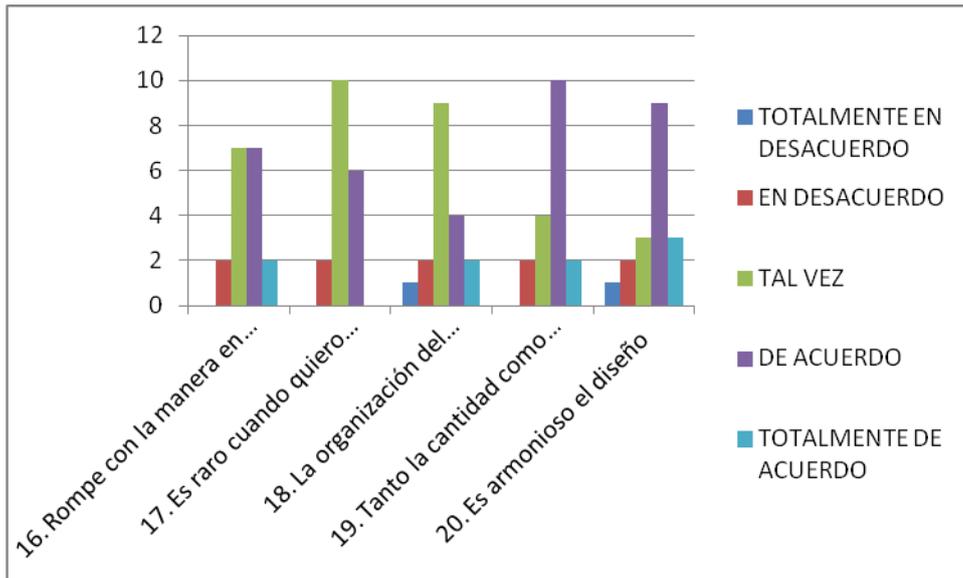
PREGUNTA	VALOR IDÓNEO	RANGO
6. Todo el lenguaje utilizado me es conocido, corresponde al lenguaje utilizado en los procesos manuales	Totalmente en desacuerdo	1
7. Algunas veces el formato utilizado tanto en entradas como salidas no es el mismo a lo que llevo en papel	Totalmente en desacuerdo	1
8. Puedo comprender y actuar con base en la información que me proporciona el sistema	Totalmente de acuerdo	5
9. El software se ha detenido alguna vez inesperadamente	Totalmente en desacuerdo	1
10. Las tareas pueden hacerse de una manera directa con el sistema	Totalmente de acuerdo	5



Gráfica 8 de Libertad y Control por parte del Usuario del cuestionario

Libertad y Control por parte del usuario. La navegación en el sistema a pesar de que es redundante, la tiene el usuario. Desde el momento de que puede controlar las funciones a las que entra y sale.

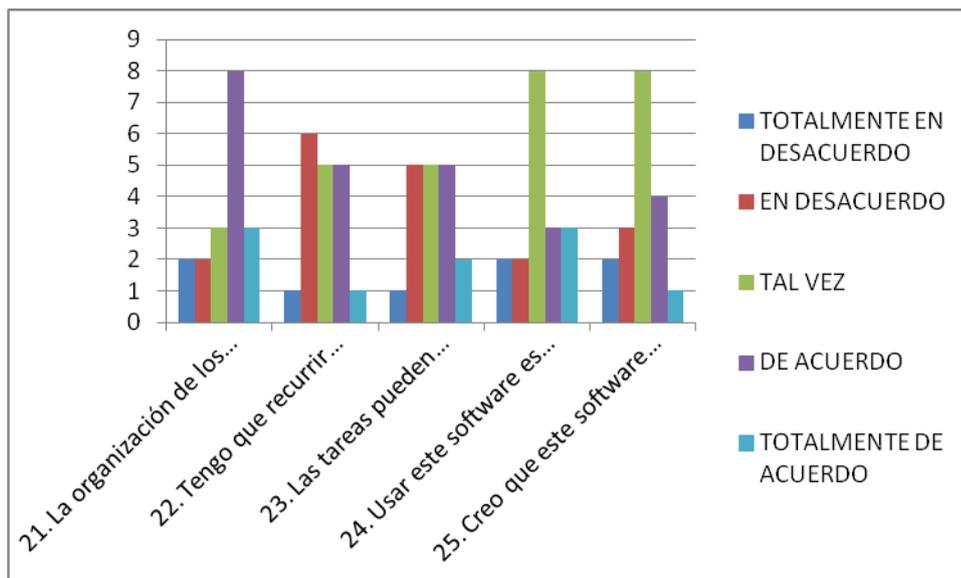
PREGUNTA	VALOR IDÓNEO	RANGO
11. Puedo ingresar a las funciones en cualquier momento	Totalmente en desacuerdo	1
12. La barra de desplazamiento existe y me es útil	Totalmente en desacuerdo	1
13. Tener muchas ventanas o páginas desplegadas en el sistema, no es un problema	Totalmente de acuerdo	5
14. Me siento al mando de éste software cuando lo uso, es fácil hacer lo que deseo	Totalmente en desacuerdo	1
15. Puedo salir de una Función a la que entré por error fácilmente	Totalmente de acuerdo	5



Gráfica 9 de Consistencia y Estándares del cuestionario

Consistencia y Estándares. Los usuarios no deben tener necesidad de discernir si palabras, situaciones o acciones distintas significan lo mismo. **Diseño de Interfaz, uso de estándares.** Se identifica que el sistema tiene un diseño armonioso, pero que la información varía de ventana en ventana, lo cual no es estándar, existe ambigüedad en la organización lógica y rompe con la funcionalidad común en las aplicaciones gráficas actuales.

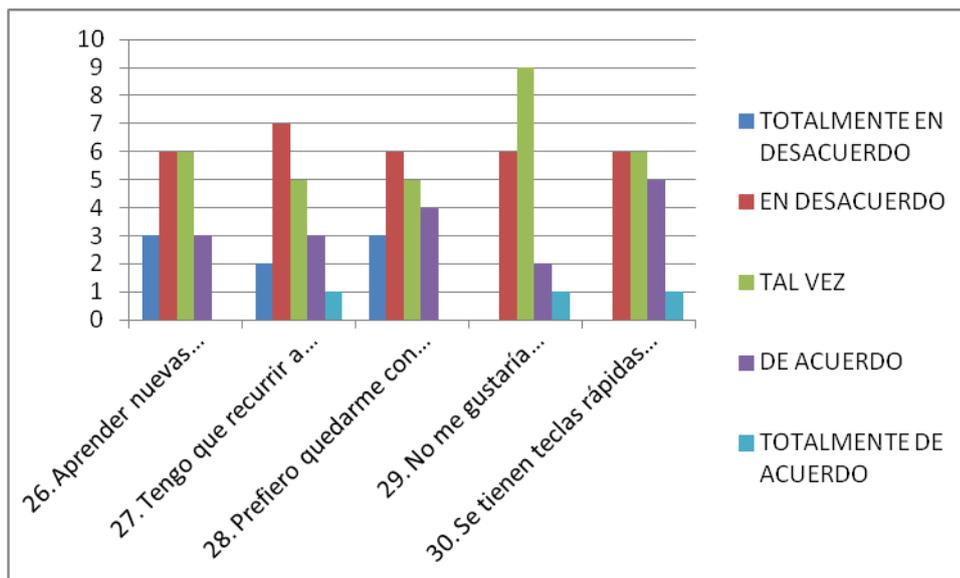
PREGUNTA	VALOR IDÓNEO	RANGO
16. Rompe con la manera en que trabajo en otras aplicaciones en la Computadora	Totalmente en desacuerdo	1
17. Es raro cuando quiero hacer algo estándar	Totalmente en desacuerdo	1
18. La organización del sistema es lógica, me lleva directamente a lo que quiero hacer	Totalmente de acuerdo	5
19. Tanto la cantidad como la calidad de la información de apoyo varían a través del sistema	Totalmente en desacuerdo	1
20. Es armonioso el diseño	Totalmente de acuerdo	5



Gráfica 10 de Prevención de Errores

Prevención de Errores. A pesar de que la organización de los menús es aceptablemente lógica, para realizar algunas tareas se recurre a apoyo y seguramente es por ello que no existe satisfacción por parte del usuario para el uso del SITA.

PREGUNTA	VALOR IDÓNEO	RANGO
21. La organización de los menús es lógica	Totalmente de acuerdo	5
22. Tengo que recurrir frecuentemente a las guías	Totalmente en desacuerdo	1
23. Las tareas pueden hacerse fácilmente	Totalmente de acuerdo	5
24. Usar este software es frustrante	Totalmente en desacuerdo	1
25. Creo que este software es inconsistente	Totalmente en desacuerdo	1



Gráfica 11 de Reconocer es mejor que recordar

Reconocer es mejor que recordar. Minimizar la carga en la memoria del usuario haciendo los objetos, acciones y opciones visibles. *Intuitivo, Predecible, Técnica paso a paso.*

PREGUNTA	VALOR IDÓNEO	RANGO
26. Aprender nuevas funciones es difícil	Totalmente en desacuerdo	1
27. Tengo que recurrir a ayuda algunas veces	Totalmente en desacuerdo	1
28. Prefiero quedarme con las funciones que conozco: Totalmente en desacuerdo	Totalmente en desacuerdo	1
29. No me gustaría recomendarlo: Totalmente en desacuerdo	Totalmente en desacuerdo	1
30. Se tienen teclas rápidas para acceder a las funciones: Totalmente de acuerdo	Totalmente de acuerdo	5

1. Flexibilidad y eficiencia de uso. Minimizar el esfuerzo requerido para alcanzar los objetivos, acortar páginas o ventanas y procesos innecesarios. *Intuitivo, Número de ventanas ó páginas visitadas para realizar una acción.*

2. Diseño minimalista y estética. Las páginas no deben contener información que sea irrelevante o remotamente necesaria. *Interfaz had hoc a la plataforma.*

3. Escribir para el Usuario. Los textos y descripciones deben estar optimizados para el sistema desde el punto de vista de los usuarios. *Lenguaje Común, Guías eficientes.*

4. Ayudar a los usuarios a reconocer, diagnosticar y recuperarse de los errores. Los mensajes de error deben expresarse en lenguaje habitual, indicar con precisión el problema y sugerir constructivamente una solución. *Prevención de errores, Recuperación de Errores, Mensajes de error, deshacer acciones.*

6 Conclusiones de la Investigación

6.1 Introducción

En orden de referencia, las secciones de éste capítulo muestran un breve resumen de los logros alcanzados, también explícitamente se responden las preguntas planteadas en el capítulo 1, posteriormente se mencionan futuras investigaciones en el mismo contexto de investigación y al final se presentan las un resumen de los resultados obtenidos.

6.2 Metas Alcanzadas

La propuesta desarrollada en cuanto al modelo de evaluación de usabilidad resultó adecuada, permitiendo el desarrollo sistemático del proceso de evaluación, ofreciendo en cada fase información útil para la aplicación de las técnicas. Pero es necesaria una mayor retención de las prácticas o métodos a utilizar en la evaluación; enfocadas a una jerarquización de los criterios de usabilidad de acuerdo al entorno del sistema. El modelo de evaluación resulta útil para diversos sistemas, ya que los criterios utilizados conducen a identificar claramente rendimientos de uso, independiente a la funcionalidad implementada, por lo que la hipótesis ha sido validada de manera metódica.

6.3 Respuesta a las Preguntas de Investigación

¿Es posible medir la usabilidad del producto software que desarrollan organizaciones no dedicadas a la fabricación de software, sin tener profesionales de la Ingeniería de la usabilidad?

Durante la investigación se hizo referencia a los temas de Calidad del software (capítulo 2) y métricas del software (capítulo 3), presentando información relevante para crear modelo de evaluación. Los principios de usabilidad, bien identificados en el capítulo 3, y sus métodos de evaluación proporcionaron la pauta para la creación de un modelo de evaluación de la usabilidad sistemático que permite su aplicación metódica con resultados concisos. Siguiendo las recomendaciones del sistema *MEDUSA*, se puede obtener fácilmente el grado de usabilidad del producto software.

¿Cómo se puede medir la usabilidad del producto software que producen organizaciones no dedicadas a la fabricación de software con métricas asistidas por computadora?

Efectivamente, bien conceptualizado la información que se deseó obtener, se implementó un sistema que de manera eficaz y eficiente pudiera proporcionar el grado de usabilidad que tiene un sistema.

Aún cuando existe software que utiliza métodos de evaluación de la usabilidad, en ellos se invierte demasiado, por ejemplo utilizando el método de evaluación observación, o bien el método de encuesta a usuarios en donde, los ingenieros tienen que procesar los datos para darle significado.

Con la propuesta señalada en ésta investigación sólo se requiere seguir las fases señaladas por el sistema *MEDUSA* de tal manera que sólo se debe hacer elección de usuarios representativos, 3 ingenieros de desarrollo con el perfil indicado y llevar a cabo la inspección, ya que el sistema proporciona los resultados tan pronto como sean requeridos.

¿Cuáles son los criterios apropiados para la evaluación de la usabilidad de productos software?

La respuesta a ésta pregunta fue dada en el capítulo 5, estudiando las características de la usabilidad, que son características cualitativas. Para la toma de decisiones con mayor fundamento se requiere información cuantitativa (de acuerdo a las métricas del software), para lo cual se crearon un conjunto de preguntas cerradas, enfocadas a investigar las características de la usabilidad, siguiendo las recomendaciones de los gurús de la manera más objetiva.

¿En qué medida los resultados de la evaluación heurística por expertos y la evaluación de la encuesta entre usuarios, corresponden?

Realizando una comparación de los resultados de ambas evaluaciones, en el capítulo 5, se observó congruencia en la información obtenida en las evaluaciones. Esto se logró gracias a la selección adecuada de los criterios de evaluación de la usabilidad, de tal manera que pudiera ser medible e impersonal. En la sección de resultados, del mismo capítulo constan los resultados de la conciliación de información.

6.4 Acerca de los Resultados obtenidos

Se definió un modelo de medición de la usabilidad, que es apoyado por el sistema *MEDUSA*, en donde la organización y presentación de datos es muy importante para su análisis, ya que los datos en bruto y desorganizados proporcionan una imagen significativa de la verdadera naturaleza de la muestra. Sin embargo obtener conclusiones de una gran cantidad de datos resulta difícil, por lo que se tomó como herramienta la gráfica de barra implementada en el sistema *MEDUSA*, a partir de éstas se pudieron identificar tendencias de manera más fácil y rápido. Cuando un sistema se construye, éste tiene priorizados algunos atributos de Calidad (de acuerdo a las necesidades de la organización), en lo que se refiere a usabilidad, siguiendo el método propuesto en este trabajo, como se ha comprobado, se pueden identificar claramente algunas características importantes como: Visibilidad en el contexto, Coincidencia entre el sistema y el mundo real, Libertad y Control por parte del Usuario, Consistencia y cumplimiento de estándares, etcétera. A continuación se proporcionan algunos análisis importantes.

De acuerdo a los resultados, en lo que respecta a la usabilidad, específicamente en el rubro de *Visibilidad en el Contexto*, a pesar de que el

control lo tiene el usuario, las etiquetas o textos utilizados son correctos; la navegabilidad es aceptable y la página principal refleja la idea del sitio y lo que se puede hacer en él. Además, presentan alta uniformidad y permanencia de controles.

Continuamente para el tipo de usuario *Tutor*, se requiere regresar a la página principal para cambiar de grupo y posteriormente elegir el alumno (a nivel de base de datos el alumno ésta identificado exclusivamente por su número de cuenta, por lo que es un paso adicional el elegir el grupo).

En lo que respecta a *Coincidencia entre el sistema y el mundo real* el sistema, la mayoría de las veces, se adapta, al lenguaje coloquial de tutorías. y presenta la información tal como el tutor ó alumno puede llevar a cabo sus procesos.

En *Libertad y Control por Parte del Usuario*, efectivamente éste puede navegar a través del SITA de manera libre, a pesar de tener pasos no necesarios para llegar a una función. La interfaz del sistema tiene consistencia en su presentación, etiquetas y formatos, por lo que se elimina como elemento de confusión hacía el usuario. En *Prevención de Errores* el sistema presenta estabilidad ya que nunca deja de funcionar, existe lógica en la información, por lo que no es una fuente de error, sin embargo se obtienen resultado de frustración al momento de utilizarlo.

En el proceso *Reconocer es mejor que recordar*, se tiene mucha ambigüedad, la interfaz del usuario tiene más etiquetas que imágenes que pudieran dar indicios de la navegabilidad del sistema. En *Flexibilidad y eficiencia de Uso*, presenta algunos problemas. El *Diseño minimalista y Estética* lo tiene de manera aceptable, en un punto donde no tiene buena aceptación es en su capacidad para reconocer, diagnosticar y recuperar errores.

7 Discusión

Algunos trabajos de evaluación de usabilidad, mencionan que debe considerarse desde la fase de análisis de requisitos. Es importante que el equipo de desarrollo reflexione acerca de la calidad del software, en medida de ello podrán insertar atributos de calidad a los sistemas. La usabilidad, indudablemente, trasciende a la productividad concebida con el uso del sistema, para ello se debe definir desde fases iniciales. Algunas veces los ingenieros del equipo de desarrollo se preocupan por que el sistema tenga la funcionalidad para la que fue creado; también debe cuestionarse acerca de cómo el software debe cumplir con la funcionalidad dada en los requerimientos.

El presente trabajo tiene continuidad, ya que sólo se ha medido la usabilidad en un producto software terminado. Se recomienda ampliar éste modelo añadiendo métodos de evaluación en fases iniciales del desarrollo del sistema, en el estudio de la usabilidad se señalaron diversos métodos que pudieran retomarse y formar parte de un modelo de evaluación más sólido. Por ejemplo, el ordenamiento de tarjetas (card sorting), que apoyaría para en la determinación de la cohesión y acoplamiento del sistema; prototipado, en apoyo a la satisfacción y aprendizaje del usuario (entre otras características) y la métrica de esfuerzo de usabilidad que da el grado de usabilidad insertado en el ciclo de vida del sistema.

Referencias

- 6Sigma. (2009). *6Sigma*. Recuperado el 10 de 2 de 2011, de &Sigma: <http://www.6sigma.us/>
- Abascal, J., Cañas, J., Gea, M., Gil, A. B., Ortega, M., & Valero, P. (2006). *Grupo de Investigación en Interacción Persona Ordenador e Integración de Datos*. Recuperado el 12 de Noviembre de 2009, de Curso Introducción a la Interacción Persona Ordenador: <http://griho.udl.es/ipo/ipo/libroe.html>
- AGESIC. (2008). *Agencia para el Desarrollo del Gobierno de Gestión Electrónica y la Sociedad de la Información y del Conocimiento*. Recuperado el 1 de 5 de 2011, de AGESIC: http://www.agesic.gub.uy/innovaportal/v/19/1/agesic/que_es_agesic.html
- BAC International Ban. (2010). *bac*. Recuperado el 3 de 2 de 2011, de Sobre la Norma ISO 9001:2008: <http://www.bac.net/regional/esp/banco/normaiso.html>
- Carego, C., Moraga, M. d., & Piattini, M. G. (2010). *Calidad del Producto y Proceso Software*. Madrid: Ra-Ma.
- Chrissis, M., Konrad, M., & Shrum, S. (2006). *Chrissis MB, Konrad M, Shrum S. 2006. CMMI: CMMI Guidelines for Process Integration and Product Improvement*. Boston Massachusetts: Addison Wesley.
- Cooper, N. (2007). *About face 3: the essentials of interaction design*. New York: Wiley.
- Ebert, C., Dumke, R. M., & Shmietendorf, A. (2005). *Best Practices in Software Measurement*. New York: Springer.
- EFQM. (2008). *EFQM*. Recuperado el 5 de 3 de 2011, de Excellence Model - EFQM 2010: http://www.efqm.org/en/PdfResources/EFQM_Ex_Mod_Teaser.pdf
- Ejiogu, L. O. (1991). *Software Enginnering with Metrics*. Boston: QED Technical Group.
- Fenton, N. E. (1991). *Software Metrics: A rigorous approach*. Virginia: Chapman & Hall/CRC Pres.
- Fenton, N. E., & Pfleeger, S. L. (1997). *Software metrics: a rigorous and practical approach*. Boston: Thomson.
- Ferré, G. X. (2005). *Marco de Integración de la Usabilidad en el Proceso de Desarrollo de Software*. Madrid: Universidad Politécnica de Madrid.
- GES. (2008). *Gestion Externa de Sistemas*. Recuperado el 5 de 4 de 2011, de Sistemas de Gestión, Normas y Estándares. Calidad ISO 9001: <http://gestionexterna.net/?lang=es&opcion=gestion&subopcion=calidad&norma=iso9001>
- Grady, R. B. (1992). *Practical software metrics for project management and process improvement*. Prentice Hall.
- Granollers, T., & Lorés, J. (2010). *Esfuerzo de Usabilidad: un nuevo concepto para medir la Usabilidad de un Sistema Interactivo basado en el Diseño Centrado en el Usuario*. Lleida España.

Granollers, T., & Lorés, J. P. (2008). *MODELO de PROCESO de la Ingeniería de la Usabilidad. Integración de la Ingeniería de Software y la Usabilidad*. Lleida, España.

GRIHO. (2009). *Grupo de Investigación en Interacción Persona Ordenador e Integración de Datos*. Recuperado el 8 de 12 de 2009, de <http://griho2.udl.cat/mpiua/wiki/index.php/Categor%C3%ADa:Accesibilidad#Introducci.C3.B3n>

Gutiérrez Gasca, E., Gutiérrez Tornés, A. F., Pérez, R. A., & Márquez López, L. F. (10 de Septiembre de 2008). *Revista Digital Universitaria*. Recuperado el 9 de 11 de 2009, de Acerca de la Implementación de los Modelos de Calidad en la Construcción de Software en México: <http://www.revista.unam.mx/vol.9/num9/art73/art73.pdf>

Hassan, M., Martín, F., & Hassan, D. (2004). Arquitectura de la Información en los entornos Virtuales de Aprendizaje. Aplicación de la Técnica Card Sorting y análisis cuantitativo de los resultados. *El Profesional de la Información*, 93-99.

Humphrey, W. S. (1995). *A discipline for software engineering*. Pittsburgh: Addison Wesley.

Information & Design. (4 de 2005). Recuperado el 07 de 05 de 2011, de Card Sorting: <http://www.infodesign.com.au/usabilityresources/cardsorting>

ISO. (2003). Recuperado el 15 de 12 de 2009, de International Standar Organization: http://www.iso.org/iso/catalogue_detail?csnumber=30858

Jacko, J. A., & Sears, A. (2003). *Human Computer Interaction HandBook: fundamentals, evolving technologies, and emerging applications*. Estados Unidos: Routledge.

La Torre, H. L. (2008). *Monografías*. Recuperado el 1 de 5 de 2011, de Propuesta de métrica de perfeccionamiento de gestión de la calidad en el proceso de desarrollo de Software: <http://www.monografias.com/trabajos55/proceso-de-desarrollo-software/proceso-de-desarrollo-software.shtml#resum>

Lecerof, A., & Paternó, F. (1998). *Automaitc Support for Usability Evaluation*. E.U.: IEEE.

MPIu+a. (3 de 11 de 2005). Recuperado el 9 de 11 de 2009, de Modelo de Proceso de la Ingeniería de la usabilidad y de la accesibilidad. MPIu+a: <http://griho2.udl.es/mpiua/index.htm>

Múria, A., & Gil, S. (1998). *Preparación, tabulación de Encuestas para Directivos*. Madrid: ESIC.

Nielsen, J. (1993). *Usability Engineering*. San Diego California: Academic Press.

Nielsen, J. (2009). *usable information technology*. Recuperado el 09 de 11 de 2009, de Alertbox: <http://www.useit.com/alertbox/>

Nielsen, J., & Molich, R. (2 de 1 de 2005). *Useit*. Recuperado el 5 de 5 de 2011, de Useit.com: http://www.useit.com/papers/heuristic/heuristic_list.html

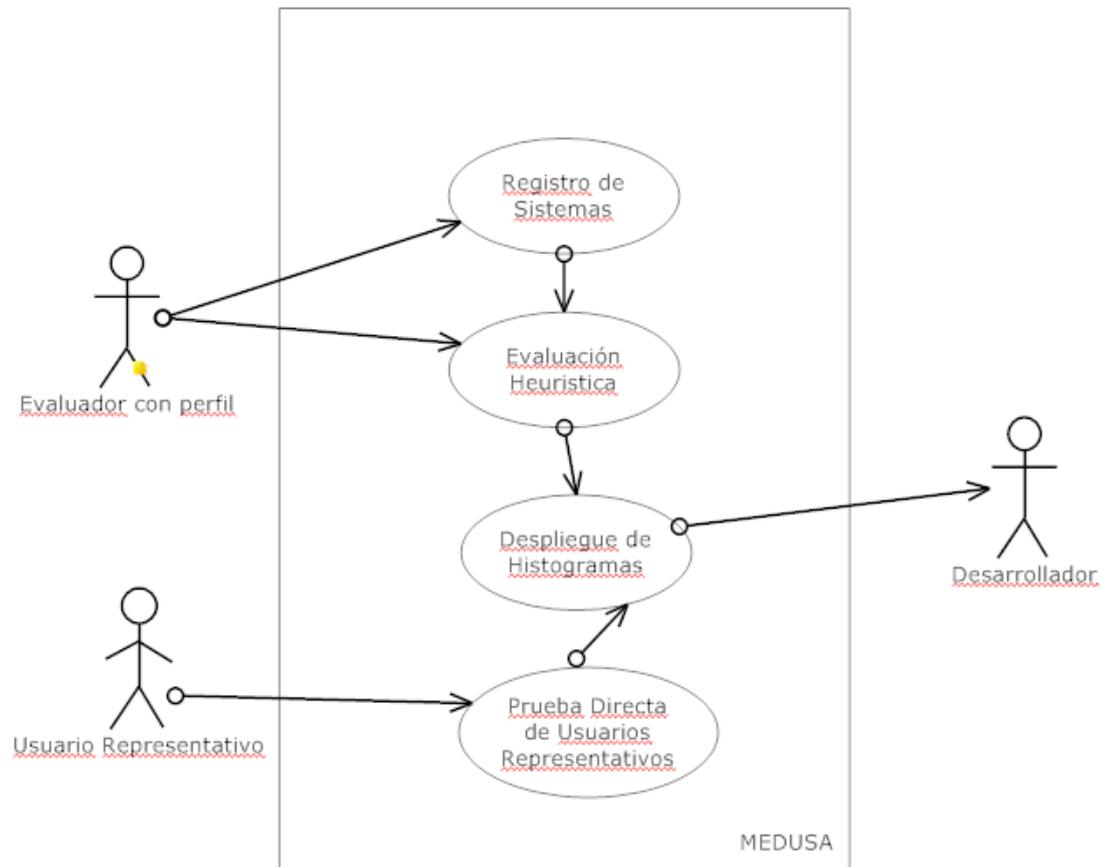
Peralta, C. E., & Hidalgo, G. C. (2007). *Agencia de Evaluación y Calidad*. Recuperado el 28 de 4 de 2011, de CAF Marco Común de evaluacion: http://www.aeval.es/comun/pdf/Guia_CAF_2006.pdf

Phillips, D. (1992). *The software project manager's handbook: principles that work at work Dwayne Phillips*. New Jersey: Jhon Wiley & Sons.

- Piattini Velthuis, M. G., García Rubio, F. O., & Caballero Muñoz, I. (2006). *Calidad de los Sistemas Informáticos*. Madrid, España: Alfaomega Ra-Ma.
- Preece, J. (2000). *Online communities: designing usability, supporting sociability*. Arizona E. U.: Wiley.
- Pressman, R. S. (2005). *Ingeniería de Software: un Enfoque Práctico*. McGraw Hill.
- Real Academia Española. (2001). *Diccionario de la Lengua Española*. Recuperado el 15 de Octubre de 2009, de http://buscon.rae.es/draeI/SrvltConsulta?TIPO_BUS=3&LEMA=calidad
- Rosenfeld, L., & Morville, P. (1998). *Information Architecture for the World Wide Web*. Cambridge: O'Really.
- Salking, N. J. (1999). *Métodos de Investigación*. Prentice Hall.
- Sánchez, S. M., & Moreno, G. A. (2003). Patrones de Usabilidad. En a. :. (JISBD'03), *Jornadas de Ingeniería del Software y Bases de Datos (JISBD)* (págs. 117-126). España: Universidad de Alicante.
- Schneiderman, B. (1998). *Designing the user interface*. New York: Addison Wesley.
- Sommerville, I. (2005). *Ingeniería del Software*. Madrid: Pearson Addison Wesley.
- Ssemugabi, S. (2006). *Usability Evaluation of a Web-Based E-Learning Application: a Study of Two Evaluation Methods*. South Africa: University of South Africa.
- Tracz, W. (2002). International Conference on Software Engineering. Orlando Florida.
- Tullis, T., & William, A. (2008). *Measuring the User Experience: Collecting, Analyzing and Presenting Usability Metrics*. Burlington: Morgan Kaufman.
- UTPL. (2008). Recuperado el 1 de 01 de 2010, de Control de Calidad: http://www.utpl.edu.ec/ecc/wiki/index.php/Control_de_Calidad

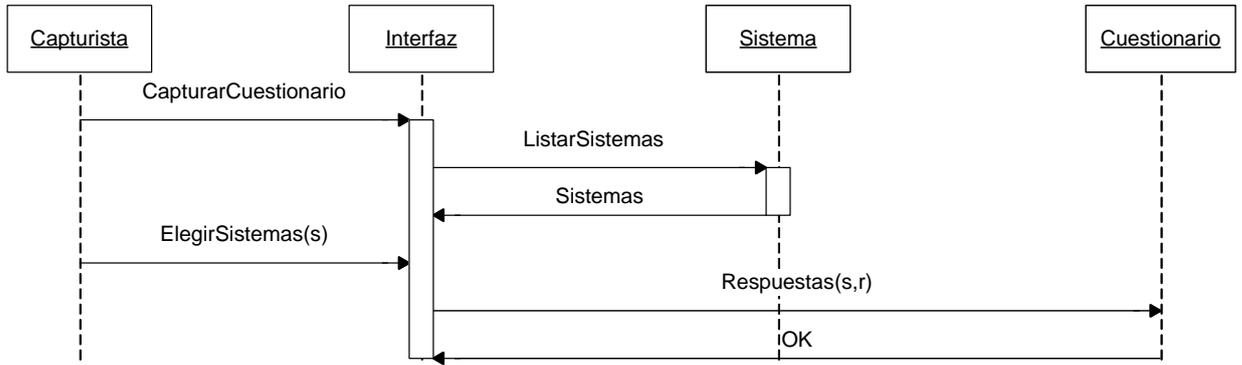
ANEXOS

Diagramas de caso de uso



Diagramas de Secuencia

Prueba de Usuarios



Evaluación Heurística

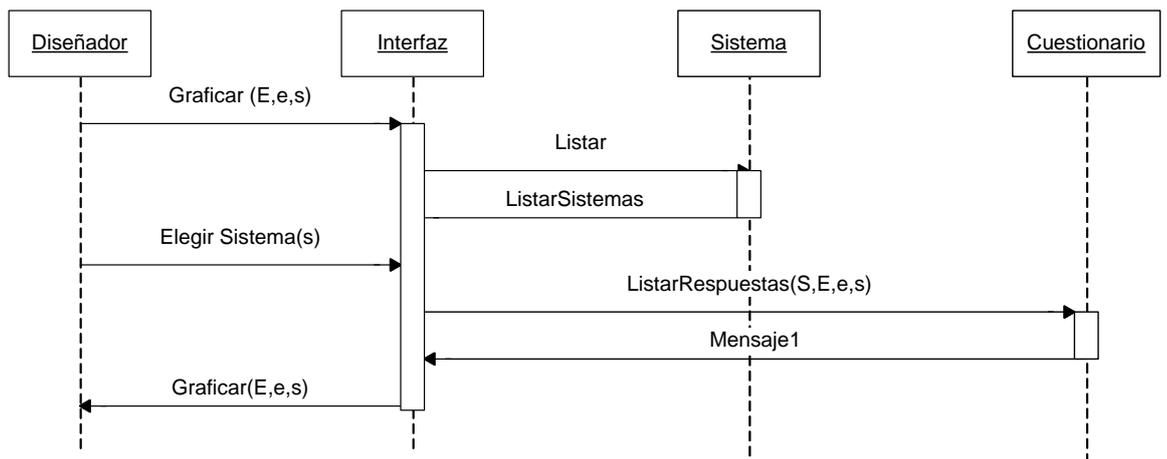


Diagrama de estado

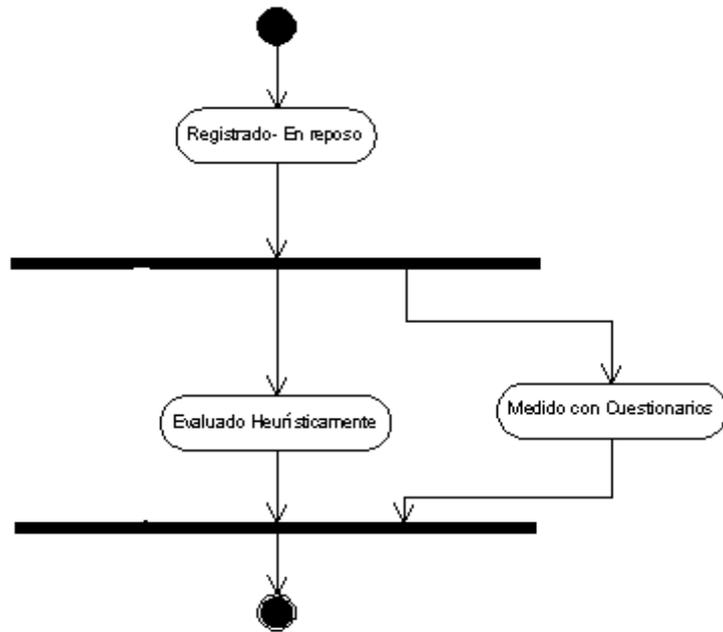
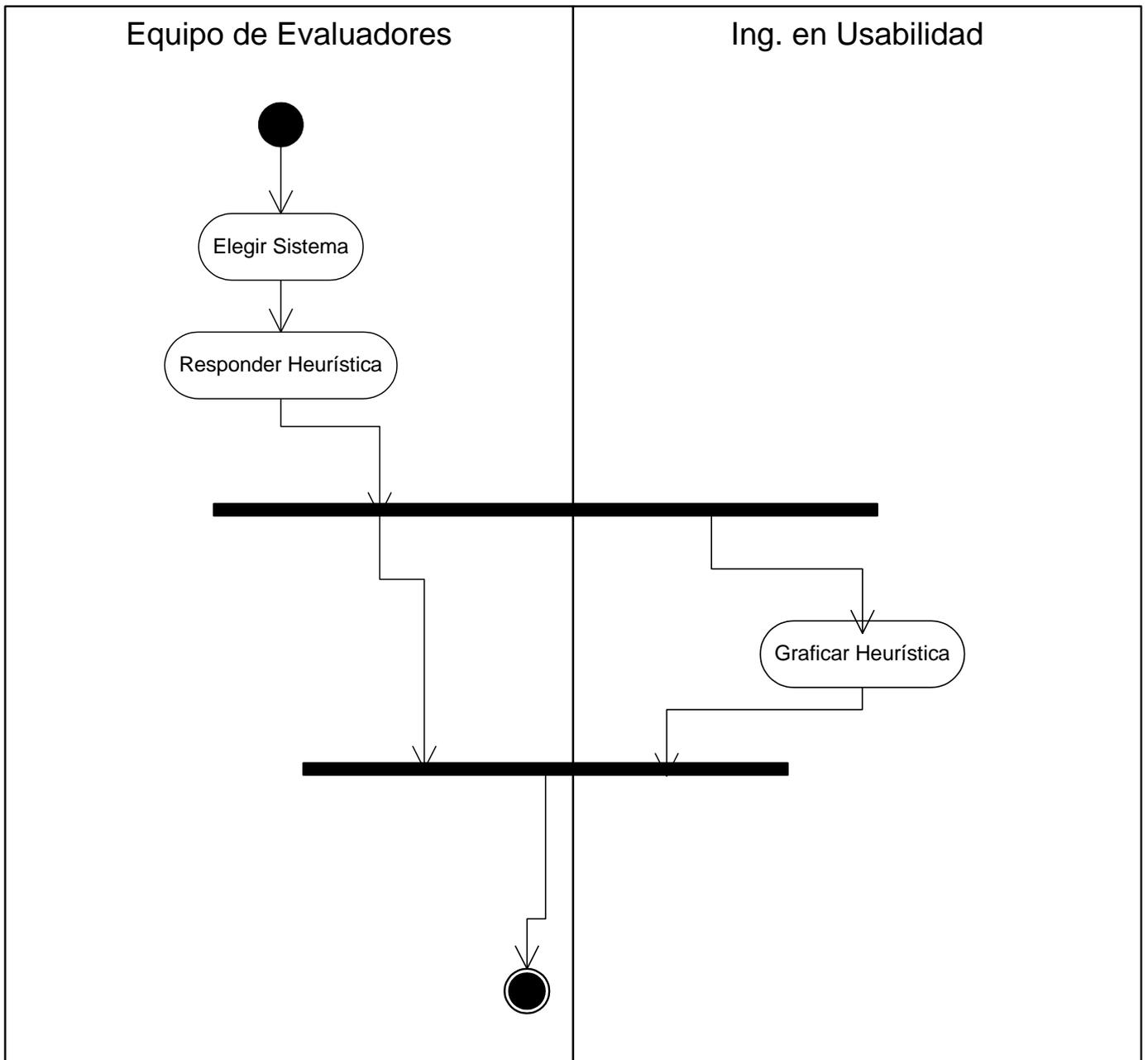
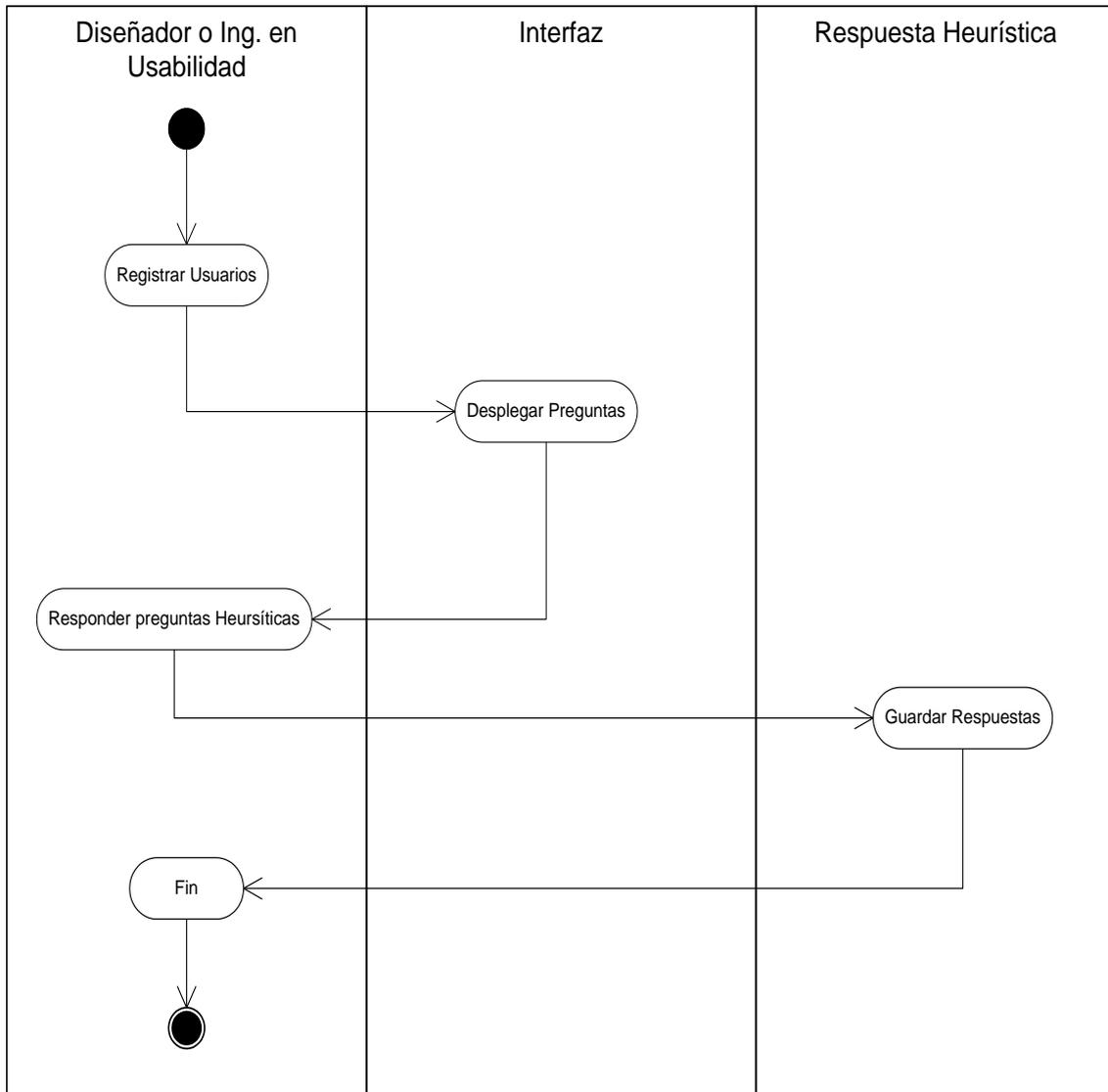


Diagrama de estado del objeto-sistema

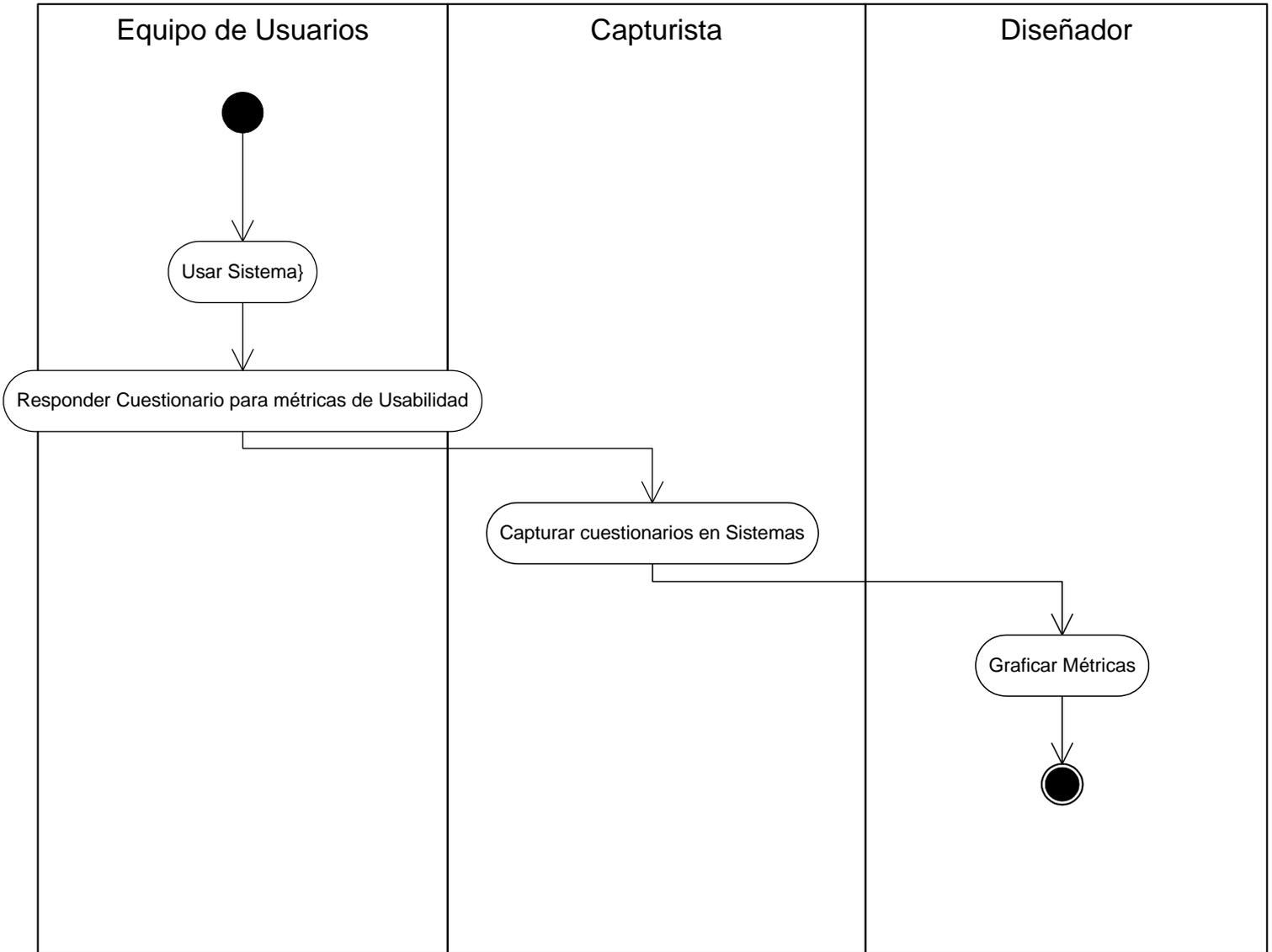
Diagrama De Colaboración



Evaluación Heurística

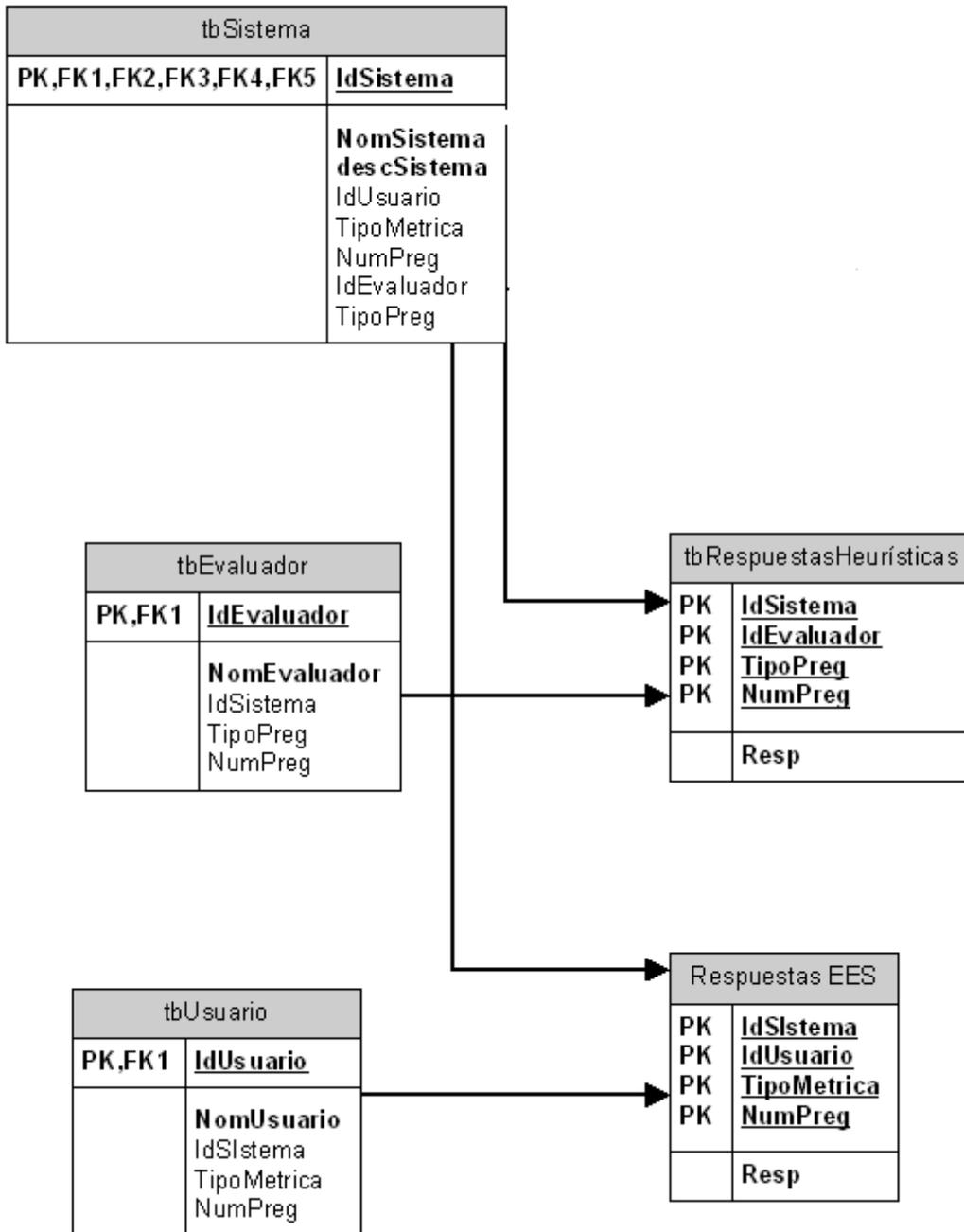


Evaluación Heurística



Métricas de Usabilidad

Modelo de Base de Datos



Script de Base de Datos

```

DROP TABLE tbEvaluador;
CREATE TABLE tbEvaluador(idEvaluador SMALLINT(2) NOT NULL,
    nomEvaluador VARCHAR(40),

```

```

        PRIMARY KEY (idEvaluador)) ENGINE=InnoDB;
DROP TABLE tbUsuario;
CREATE TABLE tbUsuario(idUsuario SMALLINT(2) NOT NULL,
        nomUsuario VARCHAR(40),
        PRIMARY KEY(idUsuario)) ENGINE=InnoDB;

DROP TABLE tbSistema;
CREATE TABLE tbSistema(idSistema SMALLINT(2) NOT NULL,
        nomSistema VARCHAR(40),
        descSistema VARCHAR(200),
        PRIMARY KEY (idSistema)) ENGINE=InnoDB;
(idSistema) ON DELETE NO ACTION) ENGINE=InnoDB;

DROP TABLE tbRespHeuristica;
CREATE TABLE tbRespHeuristica ( idSistema SMALLINT(2) NOT NULL,
        idEvaluador SMALLINT(2) NOT NULL,
        TipoHeuristica SMALLINT(2) NOT NULL,
        numPregunta SMALLINT(2) NOT NULL,
        Respuesta SMALLINT(2) NOT NULL,
        PRIMARY KEY(idSistema,idEvaluador,TipoHeuristica,numPregunta,),
        FOREIGN KEY (idSistema) REFERENCES tbSistema(idSistema) ON DELETE
        NO ACTION,
        FOREIGN KEY (idEvaluador) REFERENCES tbEvaluador(idEvaluador) ON DELETE
        NO ACTION
) ENGINE=InnoDB;

DROP TABLE tbRespuestasEES;
CREATE TABLE tbRespuestasEES ( idSistema SMALLINT(2) NOT NULL,
        idUsuario SMALLINT(2) NOT NULL,
        TipoMetrica SMALLINT(2) NOT NULL,
        numPregunta SMALLINT(2) NOT NULL,
        Respuesta SMALLINT(2) NOT NULL,
        PRIMARY KEY(idSistema,idUsuario,TipoMetrica,numPregunta,),
        FOREIGN KEY (idSistema) REFERENCES tbSistema(idSistema) ON DELETE
        NO ACTION,
        FOREIGN KEY (idUsuario) REFERENCES tbUsuario(idusuario) ON DELETE NO
        ACTION
) ENGINE=InnoDB;

```

Diagrama de Componentes

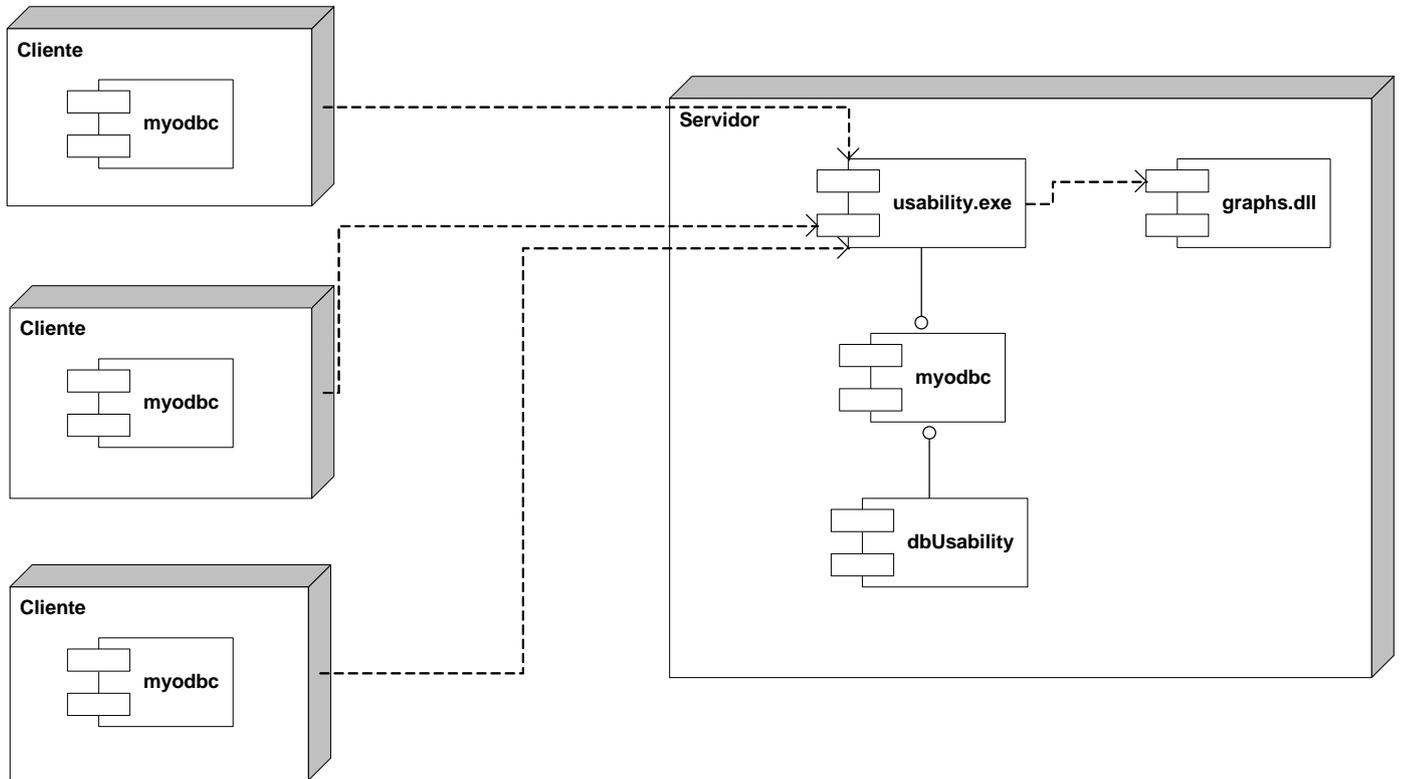
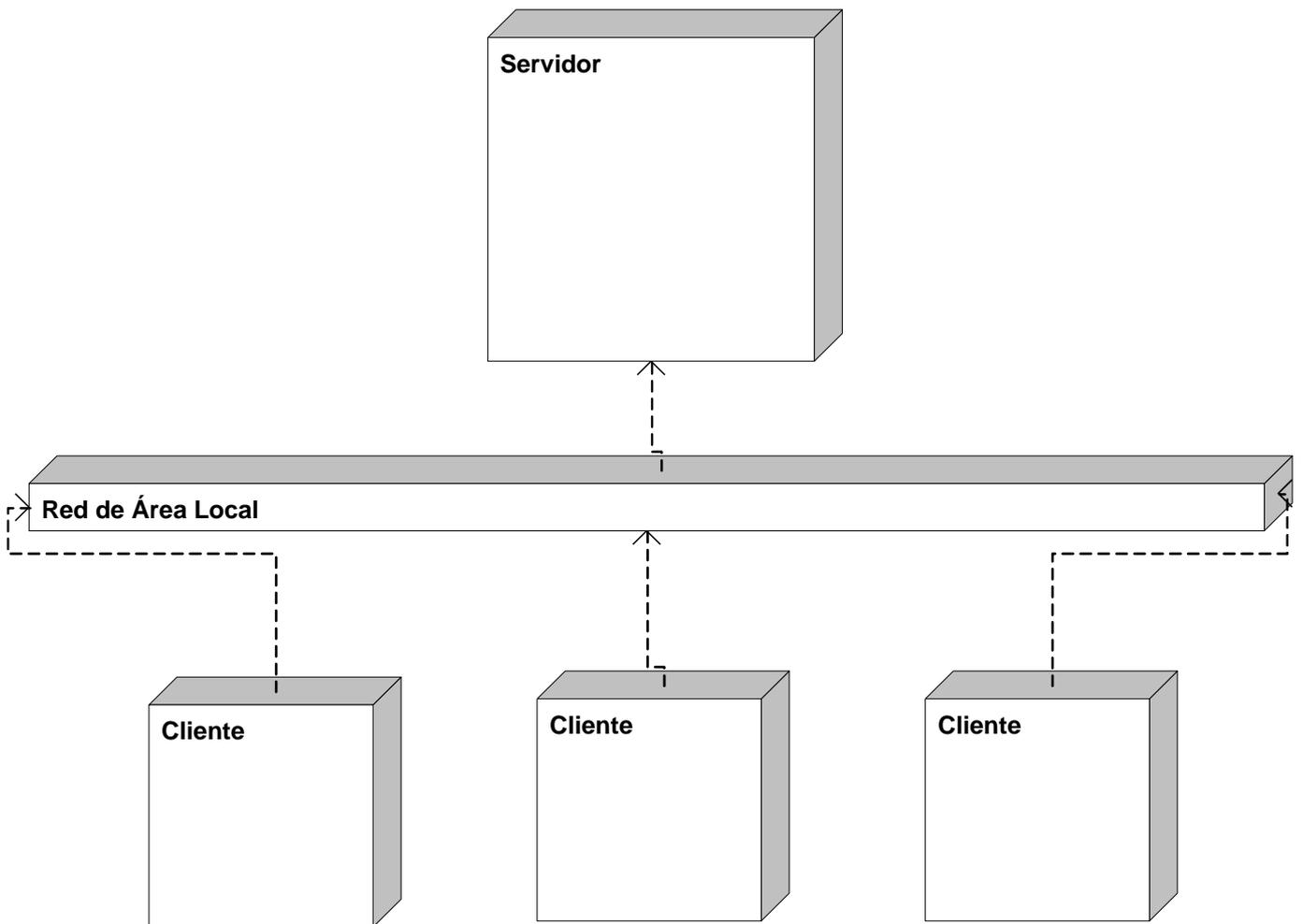


Diagrama de despliegado Nodos y LAN



Interfaz del Sistema

usabilidad Salir de esta encuesta

Visibilidad en el Contexto

1. 1. Algunas veces no se qué debo hacer después con éste software

Totalmente de Acuerdo

De acuerdo

Puede ser

En desacuerdo

Totalmente en desacuerdo

2. Requiero ingresar los mismos datos continuamente

Totalmente de Acuerdo

De acuerdo

Puede ser

En desacuerdo

Totalmente en desacuerdo

3. A veces me pregunto si estoy utilizando la función correcta

Totalmente de Acuerdo

De acuerdo

Puede ser

En desacuerdo

Totalmente en desacuerdo

4. 4. Sé, exactamente, cuando se está realizando un proceso del Sistema

Totalmente de Acuerdo

De acuerdo

Puede ser

En desacuerdo

Totalmente en desacuerdo

5. 5. Algunos Titulos de Páginas o Ventanas no corresponden exactamente a lo que se hace en ella

Totalmente de Acuerdo

De acuerdo

Puede ser

En desacuerdo

Totalmente en desacuerdo

Toda la interfaz tiene el mismo formato, lo que cambia es el tipo de pregunta y la huerísitca

Coincidencia entre el Sistema y el mundo Real

6. 6. Todo el lenguaje utilizado me es conocido, corresponde al lenguaje utilizado en los procesos manuales

- Totalmente de Acuerdo
 De acuerdo
 Puede ser
 En desacuerdo
 Totalmente en desacuerdo

7. 7. Algunas veces el formato utilizado tanto en entradas como salidas no es el mismo a lo que llevo en papel

- Totalmente de Acuerdo
 De acuerdo
 Puede ser
 En desacuerdo
 Totalmente en desacuerdo

8. 8. Puedo comprender y actuar con base en la información que me proporciona el sistema

- Totalmente de Acuerdo
 De acuerdo
 Puede ser

- En desacuerdo
 Totalmente en desacuerdo

9. 9. El software se ha detenido alguna vez inesperadamente

- Totalmente de Acuerdo
 De acuerdo
 Puede ser
 En desacuerdo
 Totalmente en desacuerdo

10. 10. Las tareas pueden hacerse de una manera directa con el sistema

- Totalmente de Acuerdo
 De acuerdo
 Puede ser
 En desacuerdo
 Totalmente en desacuerdo

Ant.

Sig.

Propiedad intelectual de Ma. Dolores Arévalo Zenteno, 2012

derechos reservados