



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

VICERRECTORADO DE INVESTIGACIÓN Y
VINCULACIÓN CON LA COLECTIVIDAD

MAESTRÍA EN INGENIERÍA DEL SOFTWARE
IV PROMOCIÓN

TESIS DE GRADO MAESTRÍA EN INGENIERÍA DEL SOFTWARE

**TEMA: DESARROLLO DE UN SISTEMA EXPERTO DE
EVALUACIÓN HEURÍSTICA QUE OPTIMICE LA MEDICIÓN
DE LA USABILIDAD DE APLICACIONES WEB EN LOS EVA
DE LA UTC.**

AUTORA

ING. VERÓNICA DEL CONSUELO TAPIA CERDA

DIRECTOR: ING. JOSÉ LUIS CARRILLO MEDINA MSc.

LATACUNGA

2015

UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE

MAESTRÍA EN INGENIERÍA DEL SOFTWARE

CERTIFICADO

ING. JOSÉ LUIS CARRILLO MEDINA MSc.

CERTIFICA

En mi calidad de director del trabajo de grado, titulado: DESARROLLO DE UN SISTEMA EXPERTO DE EVALUACIÓN HEURÍSTICA QUE OPTIMICE LA MEDICIÓN DE LA USABILIDAD DE APLICACIONES WEB EN LOS EVA DE LA UTC, presentado por la Ing. VERONICA DEL CONSUELO TAPIA CERDA, requisito previo a la obtención del título de MAGISTER EN INGENIERÍA DEL SOFTWARE, doy fe de que dicho trabajo reúne los requisitos y los méritos suficientes para ser sometido a presentación pública y evaluación por parte del jurado examinador que se designe.

En la ciudad de Latacunga, a los veinte días del mes de agosto del 2015.

Ing. José Luis Carrillo Medina MSc.

Director

UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE
MAESTRÍA EN INGENIERÍA DEL SOFTWARE
DECLARACIÓN DE RESPONSABILIDAD

ING. VERONICA DEL CONSUELO TAPIA CERDA

DECLARO QUE

El contenido e información que se encuentra en esta Tesis denominada “DESARROLLO DE UN SISTEMA EXPERTO DE EVALUACIÓN HEURÍSTICA QUE OPTIMICE LA MEDICIÓN DE LA USABILIDAD DE APLICACIONES WEB EN LOS EVA DE LA UTC”, es responsabilidad exclusiva del autor y ha respetado derechos intelectuales de terceros, conforme a las fuentes que se incorporan en la bibliografía.

En la ciudad de Latacunga, los veinte días del mes de agosto del 2015.

Ing. Verónica del Consuelo Tapia Cerda

C.C.: 0502053697

UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE

MAESTRÍA EN INGENIERÍA DEL SOFTWARE

AUTORIZACIÓN

ING. VERÓNICA DEL CONSUELO TAPIA CERDA

Autorizo a la Universidad de las Fuerzas Armadas – ESPE, la publicación, en la biblioteca virtual de la Institución del trabajo de grado denominado “DESARROLLO DE UN SISTEMA EXPERTO DE EVALUACIÓN HEURÍSTICA QUE OPTIMICE LA MEDICIÓN DE LA USABILIDAD DE APLICACIONES WEB EN LOS EVA DE LA UTC”, cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y autoría.

En la ciudad de Latacunga, los veinte días del mes de agosto del 2015.

Ing. Verónica del Consuelo Tapia Cerda

C.C.: 0502053697

DEDICATORIA

*A mi esposo Richard y a mis
hijos Richard Alexander y
Verónica Micaela, por su infinito
apoyo y paciencia en el caminar
de este nuevo reto cumplido.*

Verónica

AGRADECIMIENTO

*A la Universidad de las Fuerzas
Armadas ESPE, Extensión
Latacunga por brindarme la
oportunidad de crecer
profesionalmente y al Ing. José
Luis Carrillo por guiar este
trabajo de investigación.*

Verónica

RESUMEN

La presente investigación se plantea como objetivo general desarrollar un sistema experto de evaluación heurística que permita evaluar la usabilidad de aplicaciones web, es decir verificar en qué nivel estas aplicaciones son fáciles de usar y de comprender para los usuarios. La validación del sistema se realizó en los entornos virtuales de aprendizaje (EVA) de la Universidad Técnica de Cotopaxi (UTC), intervinieron como evaluadores cuatro profesionales de la ingeniería de software y del diseño gráfico, se aplicó la medición a una muestra correspondiente a los EVA de la carrera de Ingeniería en Informática y Sistemas Computacionales, ejecutándose en total siete proyectos que generaron treinta y cinco evaluaciones. Los resultados reflejan que los entornos virtuales de aprendizaje de la UTC tienen una calificación de 54.66%, esto significa un nivel medianamente usable; así mismo, la lista de problemas está compuesta por los 10 criterios medidos y de los 83 subcriterios, 75 generan recomendaciones de mejora; esto permite determinar que existen conflictos de usabilidad. Los criterios de mayor incidencia son: identidad e información, ayuda, búsqueda y elementos multimedia. Una vez obtenidos y analizados los resultados se diseñó un plan de mejoras, en dónde se proponen varias acciones que se deberían realizar, para optimizar la interacción de los usuarios con las aplicaciones evaluadas. Finalmente, como resultado general de toda esta investigación, se puede concluir que se han alcanzado todos los objetivos propuestos al inicio de la misma.

PALABRAS CLAVE:

- SOFTWARE
- APLICACIONES WEB
- ENTORNOS VIRTUALES - APRENDIZAJE
- USABILIDAD

ABSTRACT

The objective in this research is to develop an expert system heuristic to get the evaluation of the usability of web applications, in others words to verify the level of these applications in use facility and comprehension for users. System validation was performed in virtual learning environments (EVA) of the University Technical of Cotopaxi (UTC), to evaluate this software was necessary the intervened as evaluators of four professional engineering software and graphic design, so the measure applied was gotten from EVA career in Computer Engineering and Computer Systems, in total running 7 projects generating 35 evaluations. The finals results show that the UTC virtual learning environments have a rating of 54.66%, this means a level moderately usable; so, the problems list is composed of 10 criteria measured and 83 sub-criteria, 75 of these generated recommendations to improve; it lets to determines the existence of conflicts in usability. The criterias of highest incidence were: identity and information, help, search and media elements. Once gotten and analyzed the results obtained the next step was design an improvement plan, where it propose any actions to optimize the user interaction with applications evaluated. Finally, the result of this research, it can be concluded that it got all the goals planned at the beginning of it.

KEY WORDS:

- SOFTWARE
- WEB APPLICATIONS
- VIRTUAL ENVIRONMENTS - LEARNING
- USABILITY

ÍNDICE DE CONTENIDOS

CARÁTULA.....	i
CERTIFICADO.....	ii
DECLARACIÓN.....	iii
AUTORIZACIÓN.....	iv
DEDICATORIA.....	v
AGRADECIMIENTO.....	vi
RESUMEN.....	vii
ABSTRACT.....	viii
CAPÍTULO I.....	I
INTRODUCCIÓN	1
1.1. Antecedentes.....	1
1.2. Objetivos	3
1.2.1. Objetivo General.....	3
1.2.2. Objetivos Específicos	3
1.3. Justificación	3
1.4. Hipótesis.....	4
1.5. Variables de la Investigación	4
1.5.1. Variable Independiente	4
1.5.2. Variable Dependiente:	5
1.6. Métodos y técnicas	5
CAPÍTULO II.....	9
MARCO TEÓRICO	9
2.1. Introducción	9
2.2. Antecedentes históricos	9
2.3. Antecedentes Conceptuales y Referenciales	10

2.3.1. Gestión de la calidad del software-----	10
2.3.2. Ingeniería de la Usabilidad -----	16
2.3.3. Métodos de Evaluación de la Usabilidad-----	17
2.3.4. Evaluación Heurística -----	18
2.3.5. Sistemas Expertos -----	48
2.4. Antecedentes Contextuales -----	63
2.5. Conclusiones-----	66
CAPÍTULO III	68
DESARROLLO DEL SISTEMA EXPERTO DE EVALUACIÓN HEURÍSTICA.....	68
3.1. Introducción -----	68
3.2. Definición de la metodología de desarrollo del Sistema Experto -----	68
3.3. Desarrollo del sistema -----	69
3.3.1. Fase I: Identificación del problema -----	69
3.3.2. Fase II: Búsqueda de la fuente del conocimiento -----	78
3.3.3. Fase III: Identificación de conceptos y datos clave -----	80
3.3.4. Fase IV: Selección de las herramientas de software -----	81
3.3.5. Fase V: Adquisición del conocimiento I -----	82
3.3.6. Fase VI: Representación del conocimiento y formalización del razonamiento -----	110
3.3.7. Fase VII: Desarrollo del prototipo y pruebas -----	115
3.3.8. Fase VIII: Adquisición del conocimiento II -----	119
3.3.9. Fase IX: Mantenimiento y actualización -----	119
CAPÍTULO IV.....	120
IMPLANTACIÓN DEL SISTEMA EXPERTO EN EL DEPARTAMENTO DE SERVICIOS INFORMÁTICOS DE LA UTC.....	120

4.1. Implantación del SSEE.	120
4.2. Pruebas de aceptación	120
4.3. Formalización de la aceptación	126
CAPÍTULO V	127
VALIDACIÓN DEL SISTEMA EXPERTO EN LAS AULAS VIRTUALES DE LA UNIVERSIDAD TÉCNICA DE COTOPAXI	127
5.1. Organización de la recogida y procesamiento de datos	127
5.2. Corroboración de los resultados	131
5.3. Conclusiones	139
CAPÍTULO VI	140
CONCLUSIONES Y RECOMENDACIONES	140
6.1. Conclusiones	140
6.2. Recomendaciones	142
REFERENCIAS BIBLIOGRÁFICAS	143

ÍNDICE DE TABLAS

Tabla 2. 1: Heurísticas de usabilidad propuestas por (Nielsen, 1994)	20
Tabla 2.2: Principios de diseño de interfaces propuestos por (Tognazzini, 1993)	22
Tabla 2.3: Heurísticas de usabilidad propuestas por (Nielsen and Mack, 1994)	25
Tabla 2.4: Heurísticas de usabilidad propuestas por Shneiderman (1998)	26
Tabla 2.5: Principios propuestos por (UW Computing & Communications, 1999)...	27
Tabla 2.6: Heurísticas propuestas por (Pierotti, 2004).....	29
Tabla 2.7: MHET (Multiple Heuristics Evaluation Table) (Wheeler A., 2007)	31
Tabla 2.8: Heurísticas propuestas por (Sivaji et al., 2011)	32
Tabla 2.9: Métricas para evaluación heurística.....	41
Tabla 2.10: Aspectos Generales.....	42
Tabla 2.11: Identidad e Información.....	43
Tabla 2.12: Estructura y Navegación	44
Tabla 2.13: Rotulado.....	44
Tabla 2.14: Layout de la página.....	45
Tabla 2.15: Entendibilidad y Facilidad en la Interacción	46
Tabla 2.16: Control y Retroalimentación.....	46
Tabla 2.17: Elementos Multimedia.....	47
Tabla 2.18: Búsqueda.....	47
Tabla 2.19: Ayuda.....	48
Tabla 3.1: Plan de ejecución del SSEE	77
Tabla 3.2: Entrevistados – Gerentes y desarrolladores de software.....	80
Tabla 3.3: Tabla comparativa de lenguajes de sistemas expertos	81
Tabla 3.4: Escalas del nivel de usabilidad	90
Tabla 3.5: Escalas de valoración del cumplimiento de criterios	91
Tabla 3.6: Requisito Funcional # 001	93
Tabla 3.7: Requisito Funcional # 002	93
Tabla 3.8: Requisito Funcional # 003	94

Tabla 3.9: Requisito Funcional # 004	94
Tabla 3.10: Requisito Funcional # 005	95
Tabla 3.11: Requisito Funcional # 006	95
Tabla 3.12: Requisito Funcional # 007	96
Tabla 3.13: Requisito Funcional # 008	96
Tabla 3.14: Requisito Funcional # 009	97
Tabla 3.15: Requisito Funcional # 010	97
Tabla 3.16: Requisito Funcional # 011	98
Tabla 3.17: Caso de prueba # 001	104
Tabla 3.18: Caso de prueba # 002.....	105
Tabla 3.19: Caso de prueba # 003.....	106
Tabla 3.20: Caso de prueba # 004.....	107
Tabla 3.21: Caso de prueba # 005.....	107
Tabla 3.22: Caso de prueba # 006.....	108
Tabla 3.23: Caso de prueba # 007.....	109
Tabla 3.24: Caso de prueba # 008.....	116
Tabla 3.25: Caso de prueba # 009.....	117
Tabla 3.26: Caso de prueba # 010.....	117
Tabla 3.27: Caso de prueba # 011	118
Tabla 4.1: Lista de verificación de las pruebas de aceptación	126
Tabla 5.1: Entornos virtuales de aprendizaje que se van a evaluar.....	128
Tabla 5.2: Evaluadores de los entornos virtuales de aprendizaje.....	129
Tabla 5.3: Calificación de la usabilidad de los proyectos evaluados	131
Tabla 5.4: Problemas de usabilidad y su frecuencia en cada proyecto	132
Tabla 5.5: Recomendaciones generadas y su frecuencia en cada proyecto	133
Tabla 5.6: Plan de mejoras para la usabilidad de los EVA de la UTC	137

ÍNDICE DE GRÁFICOS

Gráfico 2.1. Modelo de calidad iso/iec 9126-1	13
Gráfico 2.2: Arquitectura de un ssee	53
Gráfico 2.3: Etapas de desarrollo de un SSEE	62
Gráfico 3.1: Modelo de Casos de Uso.....	100
Gráfico 3.2: Modelo de Clases	101
Gráfico 3.3: Diagrama de actividad - Ejecutar evaluación.....	102
Gráfico 4.1: Ventana de inicio de exshe	121
Gráfico 4.2: Ventana de administración de proyectos.....	121
Gráfico 4.3: Ventana de gestión de proyectos	122
Gráfico 4.4: Reporte consolidado de evaluación	122
Gráfico 4.5: Reporte de recomendaciones	123
Gráfico 4.6: Reporte histórico de evaluaciones.....	123
Gráfico 4.7: Ventana de mantenimiento de evaluadores	124
Gráfico 4.8: Ventana de calificación de proyecto.....	125
Gráfico 4.9: Informe de evaluación individual	125
Gráfico 5.1: Aulas virtuales de la carrera de Ingeniería en Informática y sistemas.	127

ÍNDICE DE ANEXOS

- Anexo 1: Lista de conceptos y datos clave de usabilidad y evaluación heurística
- Anexo 2: Cuestionario de entrevistas - Empresas de desarrollo de software
- Anexo 2.1: Resultados de las entrevistas a los desarrolladores y jefes de proyectos de las empresas de desarrollo de software
- Anexo 3: Manual técnico
- Anexo 4: Certificado de administrador de los Entornos Virtuales de Aprendizaje
- Anexo 5: Carta de aceptación del sistema
- Anexo 6: Instalador del SSEE
- Anexo 7: Certificados de los evaluadores

CAPÍTULO I

INTRODUCCIÓN

1.1. Antecedentes

En los últimos años el uso de aplicaciones web ha crecido significativamente, esto ha constituido a la vez, que la preocupación de desarrolladores y diseñadores por crear mejores diseños, más eficientes y de mayor calidad, confluyan en investigaciones dirigidas a mejorar la experiencia del usuario en su interacción con las aplicaciones.

A nivel mundial existen profesionales y empresas que se dedican al estudio de la usabilidad, se han diseñado propuestas que incluyen teorías, métodos y recursos que permiten medir la usabilidad de las aplicaciones, destinadas a mejorar su calidad y la experiencia de los usuarios con las mismas.

Las aplicaciones en general deben estar diseñadas para facilitar la vida de los seres humanos que las utilizan, de hecho, la usabilidad es transversal y aplicable a cualquier herramienta tecnológica destinada al uso de humanos. Las aplicaciones se deben adaptar a los usuarios, a sus habilidades, conocimientos y modelos mentales. De ahí que no se pretende que una aplicación sea usable universalmente para todos los usuarios, sino que al estar dirigida a un sector específico, se considere toda la diversidad de ese sector.

El Ecuador ya no puede seguir al margen de estas tendencias, es importante empezar a promover los estudios de usabilidad existentes, y sobre todo, hacer uso de las metodologías de evaluación de la usabilidad. En este medio, pocos desarrolladores consideran factores de usabilidad formales en los diseños y en menor porcentaje, hacen medición de usabilidad de las aplicaciones desarrolladas.

Así mismo, en la provincia de Cotopaxi es todavía más inusual el considerar factores de usabilidad en la producción de software; sin embargo, va en crecimiento porque los desarrolladores se han dado cuenta que la subsistencia de una aplicación web, depende en gran medida de su facilidad de uso.

En relación a la Universidad Técnica de Cotopaxi (UTC) institución de educación superior ubicada en la ciudad de Latacunga, tiene numerosas aplicaciones web al servicio de estudiantes, docentes, empleados y en general de toda la comunidad involucrada. Un conjunto de estas aplicaciones son los entornos virtuales de aprendizaje (EVA), pues cuenta con una plataforma MOODLE (Module Object-Oriented Dynamic Learning Environment) herramienta de distribución libre, que permite a los docentes gestionar cursos virtuales de sus asignaturas a través de las cuales brindan a los estudiantes la posibilidad de interactuar con una diversidad de elementos virtuales (contenidos multimedia, foros, chats, evaluaciones, entre otros) que a la vez, permiten que el proceso de enseñanza – aprendizaje se haga más dinámico e interesante al incorporar la tecnología al quehacer académico, sobre todo porque facilita el control automático de la participación y el aporte de cada estudiante en las diferentes asignaturas.

En conclusión se puede mencionar que existen razones valiosas que apoyan la necesidad de contar con una herramienta que evalúe la usabilidad de las aplicaciones web, la implantación de este tipo de herramientas aportaría a los desarrolladores y diseñadores porque al conocer cuáles son los factores que afectan el fácil manejo de las aplicaciones, se podrían implementar correcciones en el diseño, que habiliten a los usuarios a entender mejor el uso de estas aplicaciones y por lo tanto a optimizar la eficiencia y eficacia de los procesos que se ejecutan sobre las mismas, es decir, los beneficiarios directos serían los usuarios.

1.2. Objetivos

1.2.1. Objetivo General

- Desarrollar un sistema experto de evaluación heurística que optimice la medición de la usabilidad de los EVA en la UTC.

1.2.2. Objetivos Específicos

- Analizar la literatura publicada sobre gestión de la calidad del software y las metodologías de evaluación de la usabilidad, para diseñar el marco teórico de la investigación.
- Desarrollar el sistema experto de evaluación heurística, mediante el uso de metodologías y herramientas de ingeniería de software.
- Implantar el sistema experto de evaluación heurística en el Departamento de Servicios Informáticos de la UTC.
- Validar el sistema experto de evaluación heurística en las aulas virtuales de la UTC.

1.3. Justificación

Conocer los problemas de usabilidad de una aplicación web apoya indudablemente al mejoramiento de la calidad de la misma, es por esta razón que la idea fundamental del presente proyecto fue desarrollar un sistema experto que permita medir la usabilidad de aplicaciones web y generar reportes de las mediciones que ayuden en la toma de decisiones para mejorar la usabilidad de las mismas.

Significa entonces, que al estudiar y desarrollar el aplicativo de software propuesto, se benefician: i) los equipos de desarrollo, por cuanto se apoya con una herramienta de evaluación de la usabilidad; ii) los usuarios en general, ya que las aplicaciones van a ser desarrolladas con mayor calidad, al tener probada o mejorada la usabilidad.

Al aplicar la evaluación heurística en este trabajo, se está apoyando a que las aplicaciones que se desarrollan permitan a los usuarios adaptarse fácilmente al manejo de las mismas y por lo tanto mejoren el rendimiento de su trabajo a la hora de interactuar con las aplicaciones.

Es importante mencionar que por el hecho de validar la herramienta en los EVA de la UTC, también son beneficiarios los docentes y estudiantes que las manejan.

Otro elemento importante de rescatar en esta investigación, es que al ser considerada la industria de desarrollo de software como estratégica por el Gobierno Nacional, tiene correspondencia con el Plan Nacional del Buen Vivir 2013-2017 ya que plantea tácticamente mejorar la calidad del software que se desarrolla, articulando el objetivo de la investigación con las necesidades de esta sociedad y del país.

Finalmente, se puede concluir que la Evaluación Heurística es uno de los métodos de inspección más utilizados para la medición de usabilidad, sus ventajas en relación a otros métodos se deben principalmente según (Nielsen, 1999) a su bajo costo y al reducido esfuerzo y preparación que requiere su ejecución.

1.4. Hipótesis

Si se desarrolla un sistema experto de evaluación heurística, entonces se optimiza la medición de la usabilidad de los EVA de la UTC.

1.5. Variables de la Investigación

1.5.1. Variable Independiente

- Se desarrolla un sistema experto de evaluación heurística.

Conceptualización: Para la presente investigación el desarrollo de un sistema experto de evaluación heurística significa, la implementación de un sistema

informático a través de procesos metodológicos que guíen cada una de las etapas del desarrollo contemplando niveles de calidad. La funcionalidad del sistema estará concentrada en permitir a personas con poca experiencia en la evaluación heurística y sin conocimiento formal especializado, realizar mediciones de usabilidad a las aplicaciones web.

1.5.2. Variable Dependiente:

- Se optimiza la medición de la usabilidad de los EVA de la UTC.

Indicadores:

- Test de evaluación heurística automatizados
- Reportes de la evaluación
- Tiempos del proceso de medición
- Plan de mejoras de usabilidad para las aulas virtuales de la Universidad Técnica de Cotopaxi

1.6. Métodos y técnicas

En el presente proyecto de investigación se utilizarán métodos teóricos y empíricos.

Los Métodos Teóricos se describen a continuación:

- Método Histórico-Lógico: Permite conocer los antecedentes históricos de la evaluación de la usabilidad de las aplicaciones Web.
- Método Inductivo-Deductivo: El método inductivo al realizar el estudio de los métodos de evaluación de la usabilidad. Y el método deductivo en la conceptualización del proceso de medición de la usabilidad con la técnica de evaluación heurística.

- Método de Análisis-Síntesis: Para realizar juicios críticos sobre el análisis de los datos adquiridos durante la recopilación de información, se establece el marco de referencia sobre la evaluación de la usabilidad y la técnica heurística.
- Método Hipotético-Deductivo: Este método está presente en toda la investigación, ya que desde el inicio del estudio de la situación problemática se planteó la hipótesis de solución.
- El Método Sistémico: Este método permite desarrollar el sistema experto de evaluación heurística para medir la usabilidad de aplicaciones web.

Los Métodos Empíricos que sirven para recoger información, como la encuesta, la entrevista, análisis de documentos; elementos que van a permitir corroborar los resultados de la propuesta para su validación.

Así mismo, dentro del campo de la ingeniería de software, los métodos que tradicionalmente se han utilizado, son los cuantitativos, éstos se basan en la estadística y permiten interpretar la temática investigada. Además, al ser una disciplina aplicada, requiere de métodos de investigación que fundamentalmente le permitan conseguir dos propósitos:

1. Aumentar el conocimiento científico acerca de procesos, productos e interrelaciones en el contexto de un sistema de software, que sería el aporte teórico de la investigación.

2. Mejorar las prácticas, es decir basándose en la teoría investigada, utilizarla, que sería el aporte práctico de la investigación.

- La Investigación Bibliográfica porque se necesita conocer el estado del arte de la gestión de la calidad del software y las metodologías de evaluación de la

usabilidad; es decir, permitirá concretar el primer objetivo específico del proyecto.

- La Investigación – Acción porque luego de analizar toda la información recogida a través de la investigación bibliográfica, se necesita pasar a la práctica y desarrollar el sistema experto para medir la usabilidad de aplicaciones web a través del método de evaluación heurística.

Para la Investigación - Acción se tomará en cuenta sus etapas cíclicas: planificación, acción, observación y reflexión; a través de iteraciones que permitirán ir refinando el proceso y la búsqueda de soluciones para la problemática planteada. Para (Elliott, 1990) “la investigación-acción adopta una postura teórica según la cual la acción emprendida para cambiar la situación se suspende temporalmente hasta conseguir una comprensión más profunda del problema práctico en cuestión”. Es decir, se basa en profundizar el conocimiento sobre una situación problemática concreta para proponer en la práctica una alternativa de solución. Por lo tanto, es coherente con la investigación que se propone desarrollar en sus objetivos específicos segundo y tercero, porque se pretende estudiar a profundidad, cómo realizar estudios de usabilidad a través del método de evaluación heurística para luego en base a ese conocimiento y comprensión, desarrollar un sistema informático que permita automatizar el proceso de evaluación de la usabilidad.

De igual manera, para la construcción del Sistema Experto se utilizará una metodología de desarrollo específica, la misma que será definida en el capítulo III que corresponde a la propuesta del proyecto.

La modalidad que se ha escogido es el diagnóstico, ya que la investigadora está involucrada en la situación de estudio, es quién va a desarrollar el sistema y quién va a proponer un plan de mejoramiento. Sin embargo no existirá un seguimiento de los efectos producidos luego de la implementación de la propuesta.

La estrategia de recopilación de información se basa en los métodos teóricos y empíricos, sobre todo en la investigación bibliográfica, la misma que permitirá recoger información de fuentes primarias y secundarias, revisarla y organizarla según su calidad e impacto.

Para el análisis de la información recolectada además de las herramientas ofimáticas de Microsoft, se utilizará Zotero, una aplicación de código abierto que ayuda a los investigadores a organizar la información recogida de cualquier fuente y citarla.

Finalmente, se tomará la información correspondiente a las Aulas virtuales de la Universidad Técnica de Cotopaxi, institución donde se implantará y validará el sistema desarrollado.

CAPÍTULO II

MARCO TEÓRICO

2.1. Introducción

El presente capítulo describe toda la fundamentación teórica de la investigación, la cual abarca las siguientes secciones: Antecedentes históricos en donde se detalla el origen de la evaluación heurística. Antecedentes conceptuales y referenciales, en donde se habla de la gestión de la calidad del software, del proceso de evaluación de la usabilidad en general y de la evaluación heurística en particular, se detalla la evolución cronológica de los principios heurísticos a partir del estudio realizado por Piattini en el 2009, en ese estudio se presentan nueve propuestas con algunas variaciones en cuanto a las heurísticas a tomar en cuenta a la hora de realizar medición de usabilidad de aplicaciones. Además se incorpora la propuesta interesante de Hassan y Martín realizada en el 2003, la cual constituye una guía de aplicación de la evaluación heurística, en dónde además de considerar variables concretas de medición, también se incluyen los factores específicos que representan los indicadores para la medición, se finaliza la sección de los antecedentes conceptuales señalando la descripción de los Sistemas Expertos, su arquitectura y su proceso de desarrollo. Se concluye este capítulo, con los Antecedentes contextuales, en donde se ha recopilado una serie de estudios que permiten sustentar la aplicación de la evaluación heurística en el contexto particular de la medición de la usabilidad de las aplicaciones interactivas.

2.2. Antecedentes históricos

La evaluación heurística tiene su origen alrededor de la década de los 90, representa un método de la Ingeniería de la Usabilidad y su creación se la consigna al también considerado padre de la Usabilidad, Jakob Nielsen, el estudioso de este

campo más reconocido hasta la actualidad. Sin embargo, otros investigadores han aportado en su momento apreciaciones conceptuales que desde luego permiten verificar la evolución de la teoría al respecto, como aportes relevantes se pueden mencionar por ejemplo a (Nielsen y Molich, 1990; Nielsen 1994) quienes hablan de que “la evaluación heurística es un método para encontrar problemas de usabilidad en el diseño de una interfaz”, la cual por su complejidad requiere que cuatro o más evaluadores expertos en usabilidad apliquen un conjunto de heurísticas a un producto específico para identificar y evaluar esos problemas. Por su lado, (Piattini, 2009) dice que la evaluación heurística permite identificar los principales problemas de usabilidad de un producto de software en tiempo y costo razonable, y que es un tipo de evaluación que puede ser realizada para evaluar todo tipo de aplicaciones en cualquiera de las fases de desarrollo, aun cuando el resultado siempre será más eficaz, si se la utiliza en etapas tempranas de la producción del software.

Es importante señalar que la evaluación heurística es una técnica que ha adquirido popularidad precisamente por su eficacia y accesibilidad. Esto ha significado que se generen varias propuestas metodológicas y sobre todo diferentes criterios que se convierten en reglas generales para el diseño de aplicaciones, los mismos que son conocidos como principios heurísticos y representan la base de donde se derivan los diferentes factores de medición de la usabilidad. (Ver sección 2.3.4, literal a., pág. 20)

2.3. Antecedentes Conceptuales y Referenciales

2.3.1. Gestión de la calidad del software

Según (Pressman, 2010), la gestión de la calidad del software tiene el siguiente alcance:

- Asegurarse de que los planes de pruebas abarquen un alto porcentaje del software, ya que no solo se trata de hallar el número de defectos visibles, sino que los errores ocultos también se reduzcan de una entrega a otra.
- Trabajar por reducir la diferencia entre los recursos que se planificaron para la ejecución de un proyecto y los recursos que se utilizaron realmente, tomando en cuenta que cuando se hace referencia a recursos, estos incluyen los recursos de talento humano, de equipamiento y de tiempo.
- Educar las diferencias en velocidad y precisión en las respuestas de soporte a los problemas de los clientes.

En base a este concepto se podría definir a la Gestión de la Calidad del Software como un conjunto de actividades que son de responsabilidad directiva, se aplica a nivel de empresa o de cada proyecto, la gestión debe determinar las políticas de calidad de la empresa y cuáles van a ser las características de calidad de los productos que se generen, además debe especificar cómo garantizar el cumplimiento de estas características a lo largo de todo el proceso de desarrollo.

a. Calidad de Software

Según la definición del Instituto de Ingenieros Eléctricos y Electrónicos (IEEE, Std. 610-1990) “la calidad del software es el grado con el que un sistema, componente o proceso cumple los requerimientos especificados y las necesidades o expectativas del cliente o usuario”.

Según (Pressman, 2010), el American Heritage Dictionary, define la calidad como la característica o atributo de un producto o servicio. Es decir es una propiedad, se refiere a las características mensurables de un elemento. Para el software por su carácter intelectual y su complejidad de extensión, resulta difícil determinar los objetos mensurables. Sin embargo, sí existen formas de medir los atributos de un programa a través de indicadores como: la característica ciclométrica, la cohesión, el

número de puntos de función, las líneas de código, entre otras. Por esta razón, para medir las características del software, se plantean dos criterios de calidad: diseño y concordancia, la calidad del diseño representa características como la tolerancia y rendimiento, y la calidad de concordancia representa el nivel de cumplimiento de los detalles de diseño durante su desarrollo.

b. Estándares de calidad del software

En cuanto a estándares y normas internacionales para gestionar la calidad de software existen varias alternativas. A continuación, se van a mencionar algunos de esos estándares, tomando especial énfasis en la ISO 9126 porque define a la usabilidad como factor de calidad de un producto de software.

- ISO 9001 - Gestión de calidad. La Norma ISO 9001:2008 no es específica para la industria del software, es implementada para el desarrollo de diversos productos, incluido el software. Especifica los requisitos para un sistema de gestión de calidad, su objetivo central es la satisfacción del cliente a través de la aplicación eficaz del sistema, se incluyen procesos de mejora continua y de aseguramiento de la calidad.

Los requisitos de esta norma son genéricos porque se pretende que sean aplicables a todo tipo de organizaciones. En el caso de que exista algún requisito que no pueda ser aplicado, se considera la exclusión para la organización.

En conclusión, siendo la ISO 9001 una norma de calidad que valora y mide las diferentes etapas en la consecución de un producto de calidad, es perfectamente aplicable al desarrollo de un software ya que busca satisfacer las necesidades de calidad del usuario, así como requerimientos de calidad internos y externos.

- ISO / IEC 25000:2014 - Requisitos de calidad del software y evaluación. Esta norma proporciona una guía para el uso de la nueva serie de normas internacionales nombradas, sistemas y requisitos de calidad del software y

evaluación, presenta una visión general de los contenidos estructurados, modelos y definiciones comunes de referencia, así como la relación entre los documentos.

Es una de las normativas más completas para evaluar un software, ya que no solo evalúa la calidad de los procesos para el desarrollo del mismo, sino también, los sistemas y especificación de requisitos de calidad de un producto de software.

- ISO 9126 – 1: 2001- Software engineering - Product quality. Es un estándar para evaluar la calidad del software, presenta un modelo que clasifica la calidad en un conjunto estructurado de características: funcionalidad, fiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad.

El gráfico 2.1, destaca cada uno de los criterios de calidad de esta norma.

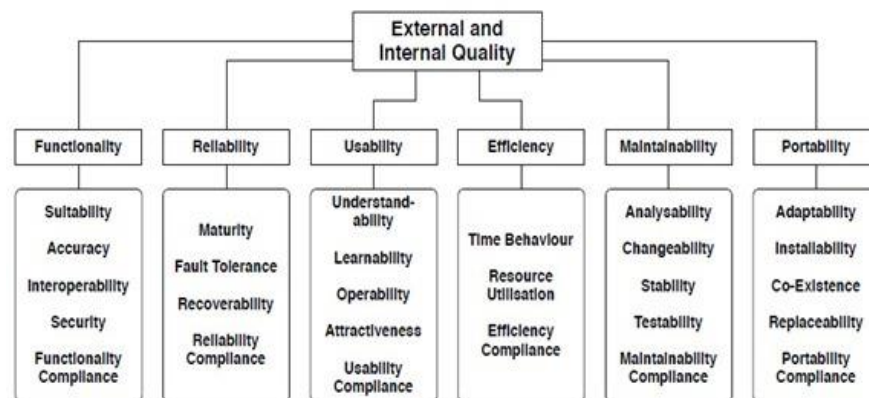


Gráfico 2.1. Modelo de calidad ISO/IEC 9126-1

Esta norma otorga a la Usabilidad la categoría de “atributo de la calidad del software”, y a la vez facilita su evaluación a partir de un conjunto de subcategorías:

- La aprendibilidad. Característica que otorga al software la capacidad de ser aprendido su manejo.
- La entendibilidad. Característica que permite que el producto sea entendido.

- La operabilidad. La capacidad que tiene el producto de ser operado y controlado por los usuarios.
- La atractividad. Capacidad del producto de ser atractivo para los usuarios.
- La consistencia. Capacidad del producto para seguir convenciones y estándares de usabilidad.

Todas estas características hacen referencia al esfuerzo en el proceso de diseño, que permite que un producto pueda ser usado con facilidad por un determinado grupo de usuarios, en contextos de uso específicos.

c. Medición de la calidad del software

Según (Pressman, 2010) “la medición puede aplicarse al proceso de software con la intención de mejorarlo de manera continua. Puede usarse a través de un proyecto de software para auxiliar en estimación, control de calidad, valoración de productividad y control de proyecto.”

La medición de la calidad de software ayuda a conocer cuantitativamente el estado de un producto en relación al cumplimiento de las expectativas del usuario, permite lograr cierto entendimiento acerca del proceso y del proyecto en general. El objetivo concreto de una medición, tal como lo dice Pressman, es el mejoramiento continuo del proceso y el producto, representa una actividad administrativa que brinda soporte al equipo de desarrollo para la toma de decisiones en cuanto a procesos de ejecución y perfeccionamiento. Es decir, la medición fundamentalmente permite determinar la medida cuantitativa de cumplimiento de un atributo específico dentro de un producto de software, para ello, es necesario determinar, qué atributos se va a medir y cuáles son las métricas que se van a utilizar.

d. Métricas para la calidad del software

De acuerdo al (IEEE Std 610.12-1990) métrica es “una medida cuantitativa del grado en el que un sistema, componente o proceso posee un atributo determinado”.

Para lograr producir un software de calidad es necesario aplicar modelos y metodologías durante todo el proceso de desarrollo de un proyecto; además medir continuamente, el nivel de calidad del producto que se está generando. Para (Pressman, 2010) “la calidad de un sistema, aplicación o producto sólo es tan buena como los requerimientos que describen el problema, el diseño que modela la solución, el código que conduce a un programa ejecutable y las pruebas que ejercitan el software para descubrir errores”. El autor además presenta las siguientes alternativas de medición: exactitud, capacidad de mantenimiento, integridad y usabilidad, de las cuales dice “proporcionan útiles indicadores para el equipo del proyecto”. (Gilb, 1988), establece la definición para cada una de estas características de medición:

- Exactitud. Grado en el cual el software realiza la función requerida. La medida más común son los defectos por miles de líneas de código (KLOC).
- Capacidad de mantenimiento. Representa la facilidad con la que un programa se puede corregir al momento de encontrarse un error y la capacidad con la que cuenta para adaptarse a los cambios que afecten en su funcionalidad, generalmente se usan medidas indirectas y la métrica más simple es el tiempo medio al cambio (TMC), el tiempo que tarda en la implementación del cambio y su validación.
- Integridad. Mide la habilidad de un sistema para resistirse a los ataques, se deben definir también los atributos de amenaza y seguridad, a través de la siguiente fórmula:

$$\text{Integridad} = \Sigma[1 - (\text{amenaza} \times (1 - \text{seguridad}))]$$

Ec. 2.1

En donde:

- Amenaza es la probabilidad (que puede estimarse o derivarse de evidencia empírica) de que un ataque de un tipo específico ocurrirá dentro de un tiempo dado y seguridad es la probabilidad (que puede estimarse o derivarse de evidencia empírica) de que el ataque de un tipo específico se repelerá.
- Usabilidad. Permite cuantificar la facilidad de uso y puede medirse en base a las sub-características que define la ISO 9126-1: 2001, (ver gráfico 2.1, acerca del modelo de calidad que plantea esta norma).

2.3.2. Ingeniería de la Usabilidad

El término usabilidad viene del inglés “usability”, se refiere al hecho de establecer si una aplicación informática es fácil de usar y es útil para los usuarios, su aparición data de finales de la década de los 80. En la actualidad la usabilidad es considerada como un atributo de los productos de software, cada vez adquiere mayor importancia entre los diseñadores y desarrolladores de software precisamente porque si una aplicación no es usable, simplemente el usuario la excluye.

Dada su importancia, varios teóricos y organizaciones de estandarización internacionales aportan con sus definiciones de Usabilidad, a continuación se detallan las más trascendentales:

Para la norma (ISO 9241-11, 1998) “la usabilidad es el grado por el cual un producto puede ser usado por unos usuarios específicos para alcanzar ciertas metas especificadas con efectividad, eficiencia y satisfacción en un contexto de uso especificado”.

Según la (ISO/IEC 9126: 2001) “la Usabilidad es la capacidad que tiene un producto software para ser entendido, aprendido, operable, atractivo para el usuario y conforme a estándares/guías, cuando es utilizado bajo unas condiciones específicas.” (Ref. Gráfico 2.1).

Jakob Nielsen (Rodríguez, Santamaría, Sáez, & Lluch, 2011), hace referencia acerca de la Usabilidad, como “...un atributo de calidad de las aplicaciones que mide la facilidad de su uso” y aporta 5 criterios para realizar esta medición: facilidad de aprendizaje, eficiencia de uso, facilidad de memorización, errores y satisfacción.

Como se ha planteado en los párrafos precedentes, la usabilidad es un factor de calidad, el hecho de que sea incluida en estándares internacionales que guían la medición de la calidad de un producto de software, confirma su importancia y la necesidad de mejorar continuamente los productos, enmarcados siempre, bajo la perspectiva de una experiencia positiva de los usuarios en su interacción con las aplicaciones.

2.3.3. Métodos de Evaluación de la Usabilidad

La evaluación de la usabilidad debe ser parte del proceso de desarrollo de software, su aplicación permitirá garantizar que el producto final cumpla con las características más relevantes de la usabilidad: fácil de usar, fácil de aprender y por lo tanto, satisfactorio para el usuario.

De acuerdo a los autores (Hernández & Cruz, 2009) que a su vez citan a (Granollers, 2004), evaluar la usabilidad de un software incluye aplicar de manera sistemática y práctica, métodos apropiados de recogida de requisitos, desarrollo y evaluación de prototipos, evaluación de alternativas, definición y evaluación de objetivos de usabilidad, analizar a los usuarios finales e implicarlos tanto en la evaluación de los avances producidos como con todas las fases en las que el proyecto

lo requiera, y constituir además equipos multidisciplinares en los que el peso de las responsabilidades parciales recaerá en la disciplina más adecuada.

En conclusión, la evaluación de la usabilidad es importante en todo el proceso de desarrollo, en sus primeras etapas porque permite establecer lo que hay que hacer. Posteriormente, en etapas de implementación y testeo porque permite determinar, si se ha hecho lo que se tenía que hacer.

En cuanto a los métodos de evaluación que se pueden aplicar, existen varias alternativas. Según (Rodríguez et al., 2011) para evaluar diseños existentes, es decir diseños de aplicaciones que se encuentran ya en producción, el tipo de evaluación que se debe usar, es la “evaluación sumativa”, ya que realiza una valoración absoluta o comparativa que permite deducir si los objetivos del usuario y del producto se han alcanzado. Otros métodos importantes que mencionan estos autores, son los métodos de evaluación sin usuarios, un tipo de evaluación en la que los evaluadores son un grupo de expertos con conocimiento especializado y formal sobre evaluación de la usabilidad, siendo la “evaluación heurística” la más conocida y aplicada.

En relación a la efectividad, los autores sustentan que estas alternativas de análisis permiten obtener buenos resultados, sin embargo en aquellos casos en los que los problemas no están focalizados, es recomendable, empezar con una evaluación sin usuarios y reservar métodos de evaluación con usuarios para fortalecer los resultados.

2.3.4. Evaluación Heurística

Para (Nielsen y Molich, 1990; Nielsen 1994) “la evaluación heurística es un método de la ingeniería de usabilidad que tiene el objetivo de encontrar los problemas de usabilidad en un diseño de interfaz de usuario para que puedan ser atendidos como parte de un proceso de diseño iterativo”. La evaluación heurística implica tener un pequeño conjunto de evaluadores, examinar la interfaz y juzgar su conformidad con los principios de usabilidad reconocidos (heurísticas).

(Montilla & Noguera, 2014) mencionan que la evaluación heurística es el método más popular entre los métodos de inspección de usabilidad, es fácil, rápido y barato, necesita de un grupo de tres a cinco evaluadores y utiliza un conjunto de heurísticas de diseño.

Según (Hassan Montero & Martín Fernández, 2004) “la evaluación heurística es un método de inspección en el que un grupo reducido de expertos, en base a su propia experiencia, fundamentándose en reconocidos principios de usabilidad (heurísticos), y apoyándose en guías elaboradas para tal fin, evalúan de forma independiente el producto, contrastando finalmente los resultados con el resto de evaluadores”.

El Diccionario de la Real Academia de la Lengua Española, define a la heurística como una “técnica de indagación y descubrimiento”. Si se traslada este concepto a la evaluación de la usabilidad, significa que la evaluación heurística es un método que permite descubrir problemas de usabilidad en el diseño de las interfaces de usuario, aplicando para ello, la medición de principios heurísticos establecidos por expertos en el campo del diseño centrado en el usuario.

El método desarrollado originalmente por Nielsen y Molich en 1990. Tiene una efectividad según el mismo (Nielsen, 1994) de “un 42% para los problemas graves de diseño y un 32% para los problemas menores, esto dependiendo del número de evaluadores que participen en la evaluación”.

Difícilmente un método de evaluación va a permitir encontrar todos los problemas de usabilidad en una aplicación, lo que se pretende es reducir los problemas para mejorar la usabilidad. El método de evaluación heurística tiene algunas ventajas en relación a otros métodos ya que permite realizar la evaluación desde etapas tempranas incluso sin la necesidad de involucrar a ningún usuario, esto representa que cualquier error puede ser corregido a tiempo sin tener que invertir mucho tiempo y dinero.

a. Principios Heurísticos

Los principios heurísticos representan reglas fundamentales que se deben tomar en cuenta a la hora de diseñar las interfaces gráficas de una aplicación de software, la consideración de estos principios o heurísticas inciden directamente en la facilidad de uso de las aplicaciones y por lo tanto en el nivel de satisfacción del usuario cuando interactúa con las mismas.

La evaluación heurística precisamente se vale de estos principios para concretar las mediciones de usabilidad de las aplicaciones.

Mario Piattini en el 2012, en su proyecto denominado “Medusas: Mejora y Evaluación del Diseño, Usabilidad, Seguridad y Mantenibilidad del Software”, un trabajo realizado a través del grupo Alarcos de la Universidad de Castilla-La Mancha de España, (grupo dedicado a investigar sobre temas de calidad y sostenibilidad de los sistemas de información), recoge varias propuestas de los principios heurísticos que se pueden utilizar para medir la usabilidad de productos interactivos en general. A continuación, se presenta cada una de ellas.

La primera propuesta es la planteada por Jakob Nielsen, ésta se basa en un análisis de problemas de usabilidad y el objetivo es proveer a los diseñadores de 10 reglas básicas que se deben aplicar en el diseño de interfaces de usuario, tal como se describe en la tabla 2.1.

Tabla 2. 1

Heurísticas de usabilidad propuestas por (Nielsen, 1994)

HEURÍSTICA	DESCRIPCIÓN
Visibilidad del estado del sistema	El sistema debe mantener al usuario siempre informado sobre lo que está ocurriendo.

CONTINÚA →

Relación entre el sistema y el mundo real	El sistema debe emplear el lenguaje del usuario
Control y libertad por parte del usuario	El usuario puede cometer un error por equivocación, por lo cual es necesario que el sistema cuente con métodos de escape, brindándole al usuario la posibilidad de volver atrás.
Consistencia y estándares	Se refiere al análisis de la coherencia en la utilización de componentes y diseños a lo largo de los distintos estados del sistema.
Prevención de errores	Mejor que disponer de buenos mensajes de error, es que el sistema tenga un diseño cuidadoso que evite la aparición de errores.
Reducción de la memoria inmediata	Analiza las cuestiones que tiene que ver con la necesidad de memorización del usuario ante la utilización de la aplicación.
Shortcuts o Accesos Rápidos	Analiza todos los aspectos que se relacionan con la incorporación o no de atajos (por ejemplo, utilización de combinación de teclas para realizar una función como guardar un documento o imprimir), para usuarios experimentados y en funcionalidades recurrentes de la operatoria del sistema.
Estética y diseño minimalista	Los diálogos no deben contener información irrelevante o raramente necesaria. El exceso de información disminuye la visibilidad del sistema.
Proporcionar ayuda a los usuarios para reconocer, diagnosticar y recuperarse de errores	Los mensajes de error deben ser expresados en un lenguaje sencillo, indicando con precisión el problema y sugiriendo una solución concreta.
Ayuda y documentación	Analiza todos los aspectos relacionados a la ayuda otorgada por el sistema al usuario. Se analizan parámetros como acceso a la ayuda, ilustraciones, profundidad y navegabilidad de la ayuda, etc.

La siguiente propuesta es la que plantea Tognazzini, quien presenta 16 principios que considera fundamentales en diseño de interfaces efectivas, el objetivo es mejorar la configuración y el aprendizaje del diseño de interfaces. El detalle de esta propuesta se describe en la tabla 2.2.

Tabla 2.2.

Principios de diseño de interfaces propuestos por (Tognazzini, 1993)

PRINCIPIO	DESCRIPCIÓN
Anticipación	Las aplicaciones deberían adelantarse a lo que el usuario quiere y necesita
Autonomía	<p>El computador, la interfaz y el entorno de trabajo "pertenecen" al usuario, pero dar autonomía al usuario no significa que se abandonen las reglas.</p> <p>Se deben utilizar mecanismos de estado para mantener a los usuarios informados.</p> <p>Además, se debe mantener la información sobre el estado, actualizada y visible.</p>
Imposibilidad de ver colores	Cada vez que se utilice el color para transmitir información en la interfaz, también se deben utilizar las señales claras para que las personas daltónicas puedan recibir esa información.
Consistencia	<p>La importancia de mantener la consistencia varía. A continuación se muestra una lista ordenada (de mayor a menor) de elementos que deben mantener consistencia:</p> <ul style="list-style-type: none"> • Interpretación del comportamiento del usuario, ej. las teclas de acceso directo deben mantener su significado. • Estructuras invisibles. • Pequeñas estructuras visibles. • La apariencia de una aplicación - pantallas de inicio, los elementos de diseño, etc.

CONTINÚA →

	<ul style="list-style-type: none"> • Consistencia de los desarrollos realizados dentro de la misma empresa u organización. • Consistencia con respecto a la plataforma. <p>Se deben evitar uniformidades. Los objetos deben ser consistentes con su comportamiento. Los objetos que actúen diferente deben tener distinta apariencia.</p> <p>Es fundamental que exista consistencia con respecto a lo que esperan los usuarios.</p>
Defecto	Los campos con valores por defecto deben aparecer seleccionados y los usuarios deben poder cambiar su valor de forma rápida y sencilla.
Eficiencia del usuario	<p>Se debe mirar la productividad del usuario y no la del ordenador.</p> <p>Se debe mantener al usuario ocupado y maximizar la eficiencia de todos los usuarios, no sólo de un grupo.</p> <p>Los mensajes de ayuda deben estar bien escritos y responder al problema.</p> <p>Los menús y las etiquetas deben contener en primer lugar las palabras claves.</p>
Interfaces explorables	<p>Es mejor dar a los usuarios caminos bien marcados y puntos de referencia.</p> <p>Las acciones deben ser reversibles y debe existir la posibilidad de deshacerlas. Además, los usuarios deberían poder salir de un determinado estado de forma sencilla.</p>
Ley de Fitt	El tiempo para alcanzar un objetivo es una función de la distancia y el tamaño del objetivo.
Objetos de la interfaz de usuario	Los objetos de las interfaces de usuarios no tienen que ser los mismos que los objetos de los sistemas orientados a objetos. En este caso existen carpetas, documentos y papelera.

CONTINÚA →

Reducción de la latencia	<p>Siempre que sea posible utilizar multihilo.</p> <p>Eliminar cualquier elemento de la aplicación que no sea de ayuda.</p>
Facilidad de aprendizaje	Se debe mostrar la curva de aprendizaje. Aunque lo ideal sería que no fuese necesario un aprendizaje de la aplicación y que el usuario pudiese directamente utilizarla, esto no es real.
Uso de metáforas	Se deben seleccionar buenas metáforas que permitan a los usuarios comprender todos los detalles.
Proteger el trabajo de los usuarios	Se debe asegurar que los usuarios nunca pierdan su trabajo, debido a un error por parte de los usuarios, un error de conexión con Internet, o cualquier otro motivo.
Facilidad de lectura	<p>El texto debe tener un alto contraste.</p> <p>Se deben utilizar tamaños de fuente lo suficientemente grandes como para que se puedan leer en monitores estándares.</p> <p>Se debe prestar especial atención a las necesidades de las personas mayores.</p>
Seguimiento del estado	La información de estado debe ser guardado en una cookie en el equipo del cliente durante la sesión y posteriormente, se debe almacenar en el servidor cuando se cierre la sesión.
Navegación visible	Se debe evitar la navegación invisible para evitar que los usuarios se pierdan.

A continuación en la tabla 2.3 se describe la propuesta de Nielsen y Mack cuyo objetivo es mejorar su propuesta anterior mediante una nueva de versión de los apartados heurísticos con el fin de estandarizar las normas de usabilidad.

Tabla 2.3.**Heurísticas de usabilidad propuestas por (Nielsen and Mack, 1994)**

HEURÍSTICA	DESCRIPCIÓN
Visibilidad del estado del sistema	El sistema debe mantener al usuario siempre informado sobre lo que está ocurriendo, mediante una retroalimentación apropiada y en un tiempo razonable.
Relación entre el sistema y el mundo real	El sistema debe emplear el lenguaje del usuario, utilizando palabras, frases, y conceptos que le sean familiares al usuario y no términos orientados al sistema.
Control y libertad por parte del usuario	El usuario puede cometer un error por equivocación, por lo cual es necesario que el sistema cuente con métodos de escape, brindándole al usuario la posibilidad de volver atrás. Por tanto, debe disponer de una "salida de emergencia" para poder salir de los estados no deseados sin necesidad de tener que navegar por multitud de cuadros de diálogo. Además, el sistema debe permitir las opciones de deshacer y rehacer.
Consistencia y estándares	Se refiere al análisis de la coherencia en la utilización de componentes y diseños a lo largo de los distintos estados del sistema.
Prevención de errores	Mejor que disponer de buenos mensajes de error, es que el sistema tenga un diseño cuidadoso que evite la aparición de errores.
Reconocimiento mejor que re-llamada	Los objetos, las acciones y las opciones deben estar visibles. El usuario no debería tener que recordar información desde una parte del diálogo a la otra.
Flexibilidad y Eficiencia de Uso	El uso de aceleradores, se debe permitir a los usuarios personalizar las acciones más frecuentes.
Estética y diseño minimalista	Los diálogos no deben contener información irrelevante o raramente necesaria. El exceso de información disminuye la visibilidad del sistema. "Menos es más"

CONTINÚA →

Proporcionar ayuda a los usuarios para reconocer, diagnosticar y recuperarse de errores	Los mensajes de error deben ser expresados en un lenguaje sencillo (sin códigos), indicando con precisión el problema y sugerir una solución constructiva. Los mensajes de error nunca deben culpar al usuario de lo sucedido. Los mensajes deben ser multinivel (en un primer nivel se proporciona información más general, la cual se profundiza en el siguiente nivel). Es bueno disponer de enlaces al sistema de ayuda.
Ayuda y documentación	Aunque es mejor que el sistema se pueda utilizar sin documentación, puede ser necesario proporcionar ayuda y documentación.

A continuación en la tabla 2.4 se presenta las heurísticas propuestas por Shneiderman, quien proporciona un enfoque flexible capaz de adaptarse a diversos escenarios y de ayudar a la comodidad de los usuarios en el uso de aplicaciones interactivas. Esta propuesta contiene 8 principios, es comúnmente conocida como “Reglas de oro del diseño de interfaces”.

Tabla 2.4.

Heurísticas de usabilidad propuestas por Shneiderman (1998)

HEURÍSTICA	DESCRIPCIÓN
Consistencia	Se refiere al análisis de la coherencia en la utilización de componentes y diseños a lo largo de la aplicación.
Puntos de Salida	Se corresponde con la evaluación de las características que posee la aplicación en relación a la salida de procesos, estados intermedios, o salida completa del sistema.
Shortcuts o Accesos Rápidos	Analiza todos los aspectos que se relacionan con la incorporación o no de atajos para usuarios experimentados a funcionalidades recurrentes de la operatoria del sistema.

CONTINÚA →

Reducción de la memoria inmediata	Corresponde con el análisis de las cuestiones que tiene que ver con la necesidad de memorización del usuario ante la utilización de la aplicación.
Errores	Se refiere a todo lo relativo al manejo de errores dentro de aplicación.
Simplicidad	Se corresponde con la evaluación de la complejidad del diseño de la aplicación analizada.
Control del sistema	Se corresponde con la evaluación de las características implicadas en el manejo de la aplicación.

A continuación en la tabla 2.5 se detalla la propuesta del departamento de Computación y Comunicación de la Universidad de Washington, una lista de principios de diseño web y gráfico, basada en el trabajo de Edward Tufte, profesor de la Universidad de Yale, especialista en diseño de información y de interfaces, autor de varios libros sobre visualización de información cuantitativa.

La universidad de Washington generó su sitio web aplicando los principios de diseño web y de imágenes, incluye heurísticas que describen las características de diseño que apoyan directamente la interacción usuario-sistema para ayudar en los aspectos de navegación de las interfaces.

Tabla 2.5.

Principios propuestos por (UW Computing & Communications, 1999)

PRINCIPIO	DESCRIPCIÓN
Integridad gráfica	Los gráficos pueden ser engañosos. Hoy en día es más fácil cometer errores gráficos debido a la amplia utilización de los ordenadores. Para evitar estos errores y asegurar la integridad de los gráficos, en este trabajo, proporcionan una serie de guías a utilizar.

CONTINÚA →

Densidad de datos	Los gráficos son una buena forma de representar grandes conjuntos de datos. Una baja densidad de datos es mala ya que obliga a ver los datos de manera secuencial, en lugar de verlos de forma espacial.
Compresión de datos	Se debe utilizar la compresión de datos, para poder mostrar una gran cantidad de información en un único gráfico.
Elementos gráficos multifuncionales	Las estructuras gráficas pueden servir para diferentes propósitos.
Maximizar la tinta de los datos	El ratio de tinta de los datos se mide dividiendo la tinta de los datos entre la tinta del plot. El objetivo es hacer esa relación tan grande como sea posible.
Múltiplos pequeños	Los múltiples pequeños son un conjunto de gráficos en miniatura que se representan en una única página y que representan aspectos de un mismo fenómeno.
Chartjunk	Son elementos decorativos que no representan datos y causan confusión.
Colores	El uso de colores puede aumentar la comprensión de los datos

Se detalla a continuación en la tabla 2.6 la propuesta de Denise Pierotti en el 2004, un conjunto de 13 heurísticas para cada una de las cuales identifica un conjunto de listas de comprobación. Esta propuesta ha sido ampliamente aplicada en la empresa Xerox para evaluar la usabilidad de las aplicaciones orientadas a la web. Se basa en los 10 principios heurísticos de Nielsen, pero añade habilidades, interacción placentera y respetuosa con el usuario y la privacidad.

Tabla 2.6.
Heurísticas propuestas por (Pierotti, 2004)

HEURÍSTICA	DESCRIPCIÓN
Visibilidad del estado del sistema.	El sistema debe mantener siempre informado al usuario sobre lo que está haciendo y proporcionarle retroalimentación en un tiempo razonable.
Relación entre el sistema y el mundo real	El sistema debe hablar el lenguaje del usuario, utilizando palabras, frases, y conceptos que le sean familiares al usuario y no términos orientados al sistema. Se deben seguir convenciones del mundo real y hacer que la información aparezca de forma natural y en un orden lógico
Control y libertad por parte del usuario	El usuario debería sentirse libre para seleccionar y secuenciar las tareas, en lugar de que sea el sistema quien lo realice por él. A menudo el usuario selecciona una función del sistema por equivocación por lo que necesita una "salida de emergencia" que esté claramente identificada para salir del estado no deseado sin tener que navegar por multitud de cuadros de diálogo. Además, el sistema debe permitir las opciones de deshacer y rehacer.
Consistencia y estándares	Los usuarios no tendrían que preguntarse si diferentes palabras, situaciones o acciones significan lo mismo. Además se deberían seguir las convenciones.
Proporcionar ayuda a los usuarios para reconocer, diagnosticar y recuperarse de errores	Los mensajes de error deben ser expresados en un lenguaje sencillo (sin códigos).
Prevención de errores	Mejor que disponer de buenos mensajes de error, es que el sistema tenga un diseño cuidadoso que evite la aparición de errores.

CONTINÚA →

Reconocimiento mejor que rellamada	Los objetos, las acciones y las opciones deben estar visibles. El usuario no debería tener que recordar información desde una parte del diálogo a la otra. Las instrucciones de utilización del sistema deberían estar visibles o deberían ser fácilmente recuperables cuando sean necesarias.
Flexibilidad y Diseño Minimalista	El uso de aceleradores (que no sean vistos por los usuarios inexpertos) puede hacer más rápida la interacción para el usuario experto, de tal forma, que el sistema pueda servir tanto para usuarios expertos como inexpertos. Se debe permitir a los usuarios personalizar las acciones más frecuentes. Se deben proporcionar formas alternativas de acceso y de operación para los usuarios que tienen determinadas características especiales (por ejemplo, capacidad física o cognitiva, cultural, de lenguaje, etc.)
Estética y diseño minimalista	Los diálogos no deben contener información irrelevante o raramente necesaria
Ayuda y documentación	Aunque es mejor que el sistema se pueda utilizar sin documentación, puede ser necesario proporcionar ayuda y documentación. Cualquier información debe ser fácil de buscar, centrada en las tareas del usuario, disponer de una lista de pasos concretos a realizar y no ser demasiado larga.
Habilidades	El sistema debería respaldar, extender, complementar o mejorar las habilidades del usuario, sus conocimientos y experiencia - no reemplazarlos.
Interacción placentera y respetuosa con el usuario	Las interacciones del usuario con el sistema debería mejorar la calidad de su vida relacionada con el trabajo. El usuario debe ser tratado con respeto. El diseño debe ser estéticamente agradable y debe tener valor funcional.
Privacidad	El sistema debería ayudar al usuario a proteger su información personal o privada- perteneciente al usuario o a sus clientes.

La siguiente propuesta detallada en la tabla 2.7 integra los trabajos de (Nielsen, 1993; Nielsen and Mack, 1994; Tognazzini, 1993; UW Computing & Communications), tiene la finalidad de facilitar el uso de las heurísticas agrupándolas en una tabla de evaluación de heurísticas múltiples denominada “Multiple Heuristic Evaluation Table” (MHET).

Este trabajo pretende actualizar y complementar las heurísticas, divide a las mismas en 3 grupos: factores relacionados con la experiencia del usuario, factores relacionados con el diseño de la interfaz de usuario y factores relacionados con la arquitectura del sistema.

Tabla 2.7.

MHET (Multiple Heuristics Evaluation Table) (Wheeler Atkinson et al., 2007)

GRUPO	HEURÍSTICA
Factores relacionados con la experiencia del usuario	Aprendizaje
	Facilidad de conocimiento
	Control del usuario
Factores relacionados con el diseño de la interfaz de usuario	Facilidad de conocimiento
	Control del usuario
	Relación entre el sistema y el mundo real
	Diseño gráfico
	Navegabilidad y puntos de salida
	Consistencia
	Valores por defecto

CONTINÚA →

Factores relacionados con la arquitectura del sistema	Navegabilidad y puntos de salida
	Consistencia
	Valores por defecto
	Interacción sistema software
	Ayuda y documentación
	Manejo de errores

La siguiente propuesta detallada en la tabla 2.8 es presentada por Ashok Sivaji, Azween Abdullah y Alan G Downe, publicada en el Modelling Symposium (AMS) Fifth Asia en el 2011; pretende mejorar las heurísticas de Nielsen. El estudio muestra la eficacia de la evaluación heurística como metodología de pruebas de usabilidad en la captura de defectos dando prioridad a los esfuerzos de desarrollo y de diseño. Los resultados también refuerzan la necesidad de la integración de las heurísticas tradicionales con heurísticas personalizadas según el dominio o campo del proyecto.

Tabla 2.8.

Heurísticas propuestas por (Sivaji et al., 2011)

HEURÍSTICA	DESCRIPCIÓN
Compatibilidad	La forma en que el sistema trabaja y su apariencia deberían ser compatibles con las convenciones y las expectativas de los usuarios.
Consistencia y Estándares	La forma en que el sistema trabaja y su apariencia deberían ser siempre consistentes.
Prevención de errores y corrección	El sistema debería ser diseñado para minimizar la posibilidad de que el usuario cometa errores, con funcionalidades para detectarlas y manejarlas; los usuarios deberían ser capaces de revisar las entradas introducidas y corregir

CONTINÚA →

	errores o situaciones de error potenciales antes de que las entradas sean procesadas.
Explicitación	La forma en que el sistema trabaja debe ser estructurada y clara para los usuarios
Flexibilidad y control	La interfaz debe ser lo suficientemente flexible en cuanto a su estructura, la forma en que la información se presenta y en términos de lo que el usuario puede hacer, para satisfacer las necesidades y los requisitos de todos los usuarios y permitirles que sientan que tienen el control del sistema.
Funcionalidad	El sistema debe satisfacer las necesidades y los requisitos de los usuarios, cuando estos llevan a cabo sus tareas.
Retroalimentación informativa	El sistema debe mantener siempre informado al usuario sobre lo que está realizando a través de retroalimentación y en un tiempo razonable.
Lenguaje y Contenido	La información transmitida debe ser entendible por los usuarios.
Navegación	El sistema de navegación debe estar estructurado de forma que se permita al usuario acceder a la ayuda para un objetivo determinado tan rápido como sea posible.
Privacidad	El sistema debe ayudar al usuario a proteger la información personal o privada que pertenece a él mismo o a sus clientes.
Guía y Ayuda al usuario	Se debe proporcionar una guía y una ayuda que sea relevante, fácil de usar e informativa para ayudar a que el usuario entienda y utilice el sistema.
Claridad visual	La información mostrada en la pantalla debe ser clara y estar bien organizada.

La sección anterior describe un conjunto de propuestas acerca de principios heurísticos, según (Piattini, 2012), son las propuestas que más se utilizan, sin embargo, no son las únicas.

El problema principal en la aplicación de las diferentes alternativas analizadas, es que los autores no proporcionan las características o atributos concretos que se deben evaluar para comprobar cuantitativamente el nivel de cumplimiento de cada uno de los principios en una interfaz de usuario. Es decir, es difícil concretar una evaluación de usabilidad basada en cualquiera de las propuestas anteriores, ya que no se relacionan las mismas con los parámetros de medición de cada principio.

Existen otras alternativas que a pesar de que se basan en los principios heurísticos originales, presentan listas de comprobaciones para cada una de las variables a medir. En la siguiente sección se describe una de las propuestas más completas aplicada a la evaluación de sitios web.

b. Modelo de evaluación de Hassan y Martín

El presente modelo de evaluación, es una propuesta planteada por (Hassan y Martín, 2003), está apoyada por su actividad profesional de consultores e investigadores en el campo de la Usabilidad y pretende ser una guía de aplicación para la evaluación heurística.

La guía proporciona un conjunto de criterios, cada uno de los cuales cuenta con factores específicos de evaluación, a su vez, éstos son comprobados mediante una lista de verificación.

Es importante aclarar que de acuerdo a lo que mencionan los autores, éste modelo es perfectamente adaptable a las necesidades del contexto del tipo de interfaz que se analiza.

Los criterios y factores que se toman en cuenta en este modelo de evaluación heurística son los siguientes:

Tomados de (Hassan y Martín, 2003)

- **Aspectos Generales**

- ¿Cuáles son los objetivos del sitio web? ¿Son concretos y bien definidos?
¿Los contenidos y servicios que ofrece se corresponden con esos objetivos?
- ¿Tiene una URL correcta, clara y fácil de recordar? ¿Y las URL de sus páginas internas? ¿Son claras y permanentes?
- ¿Muestra de forma precisa y completa qué contenidos o servicios ofrece realmente el sitio web?
- ¿La estructura general del sitio web está orientada al usuario?
- ¿El look & feel general se corresponde con los objetivos, características, contenidos y servicios del sitio web?
- ¿Es coherente el diseño general del sitio web?
- ¿Es reconocible el diseño general del sitio web?
- ¿El sitio web se actualiza periódicamente? ¿Indica cuándo se actualiza?

- **Identidad e Información**

- ¿Se muestra claramente la identidad de la empresa-sitio a través de todas las páginas?
- El Logotipo, ¿es significativo, identificable y suficientemente visible?
- El eslogan o tagline, ¿expresa realmente qué es la empresa y qué servicios ofrece?
- ¿Se ofrece algún enlace con información sobre la empresa, sitio web, 'webmaster',...?
- ¿Se proporciona mecanismos para ponerse en contacto con la empresa? (email, teléfono, dirección postal, fax...)

- ¿Se proporciona información sobre la protección de datos de carácter personal de los clientes o los derechos de autor de los contenidos del sitio web?
- En artículos, noticias, informes...¿Se muestra claramente información sobre el autor, fuentes y fechas de creación y revisión del documento?

- **Lenguaje y Redacción**

- ¿El sitio web habla el mismo lenguaje que sus usuarios?
- ¿Emplea un lenguaje claro y conciso?
- ¿Es amigable, familiar y cercano?
- ¿1 párrafo = 1 idea?

- **Rotulado**

- Los rótulos, ¿son significativos?
- ¿Usa rótulos estándar?
- ¿Usa un único sistema de organización, bien definido y claro?
- ¿Utiliza un sistema de rotulado controlado y preciso?
- El título de las páginas, ¿Es correcto? ¿Ha sido planificado?

- **Estructura y Navegación**

- La estructura de organización y navegación, ¿Es la más adecuada?
- En el caso de estructura jerárquica, ¿Mantiene un equilibrio entre Profundidad y Anchura?
- En el caso de ser puramente hipertextual, ¿Están todos los clusters de nodos comunicados?
- ¿Los enlaces son fácilmente reconocibles como tales? ¿su caracterización indica su estado (visitados, activos,...)?

- En menús de navegación, ¿Se ha controlado el número de elementos y de términos por elemento para no producir sobrecarga memorística?
- ¿Es predecible la respuesta del sistema antes de hacer clic sobre el enlace?
- ¿Se ha controlado que no haya enlaces que no lleven a ningún sitio?
- ¿Existen elementos de navegación que orienten al usuario acerca de dónde está y cómo deshacer su navegación?
- Las imágenes enlace, ¿se reconocen como clicables? ¿incluyen un atributo 'title' describiendo la página de destino?
- ¿Se ha evitado la redundancia de enlaces?
- ¿Se ha controlado que no haya páginas "huérfanas"?

- **Lay-Out de la Página**

- ¿Se aprovechan las zonas de alta jerarquía informativa de la página para contenidos de mayor relevancia?
- ¿Se ha evitado la sobrecarga informativa?
- ¿Es una interfaz limpia, sin ruido visual?
- ¿Existen zonas en "blanco" entre los objetos informativos de la página para poder descansar la vista?
- ¿Se hace un uso correcto del espacio visual de la página?
- ¿Se utiliza correctamente la jerarquía visual para expresar las relaciones del tipo "parte de" entre los elementos de la página?
- ¿Se ha controlado la longitud de página?

- **Búsqueda**

- ¿Se encuentra fácilmente accesible?
- ¿Es fácilmente reconocible como tal?
- ¿Permite la búsqueda avanzada?

- ¿Muestra los resultados de la búsqueda de forma comprensible para el usuario?
- ¿La caja de texto es lo suficientemente ancha?
- ¿Asiste al usuario en caso de no poder ofrecer resultados para una consultada dada?

- **Elementos Multimedia**

- ¿Las fotografías están bien recortadas? ¿son comprensibles? ¿se ha cuidado su resolución?
- ¿Las metáforas visuales son reconocibles y comprensibles por cualquier usuario?
- ¿El uso de imágenes o animaciones proporciona algún tipo de valor añadido?
- ¿Se ha evitado el uso de animaciones cíclicas?

- **Ayuda**

- Si posee una sección de Ayuda, ¿Es verdaderamente necesaria?
- En enlace a la sección de Ayuda, ¿Está colocado en una zona visible y "estándar"?
- ¿Se ofrece ayuda contextual en tareas complejas?
- Si posee FAQs, ¿es correcta tanto la elección como la redacción de las preguntas? ¿y las respuestas?

- **Accesibilidad**

- ¿El tamaño de fuente se ha definido de forma relativa, o por lo menos, la fuente es lo suficientemente grande como para no dificultar la legibilidad del texto?

- ¿El tipo de fuente, efectos tipográficos, ancho de línea y alineación empleadas facilitan la lectura?
- ¿Existe un alto contraste entre el color de fuente y el fondo?
- ¿Incluyen las imágenes atributos 'alt' que describan su contenido?
- ¿Es compatible el sitio web con los diferentes navegadores? ¿Se visualiza correctamente con diferentes resoluciones de pantalla?
- ¿Puede el usuario disfrutar de todos los contenidos del sitio web sin necesidad de tener que descargar e instalar plugins adicionales?
- ¿Se ha controlado el peso de la página?
- ¿Se puede imprimir la página sin problemas?

- **Control y Retroalimentación**

- ¿Tiene el usuario todo el control sobre el interfaz?
- ¿Se informa constantemente al usuario acerca de lo que está pasando?
- ¿Se informa al usuario de lo que ha pasado?
- Cuando se produce un error, ¿se informa de forma clara y no alarmista al usuario de lo ocurrido y de cómo solucionar el problema?
- ¿Posee el usuario libertad para actuar?
- ¿Se ha controlado el tiempo de respuesta?

De acuerdo a la revisión documental realizada para este trabajo se puede indicar que el modelo de Hassan y Martín es usado en mediciones de usabilidad, a continuación se presenta algunas evidencias de su aplicación en estudios de evaluación heurística de sitios web:

- “Análisis de Sitios Web Universitarios Mediante Evaluaciones Heurísticas”, un estudio realizado por (Candamil M., 2012) donde se examinaron los sitios web de las 10 universidades de Colombia mejor posicionadas a febrero de

2008, además del sitio de la universidad del Cauca que se encontraba vinculada al proyecto. El estudio permitió encontrar fallas de usabilidad y de arquitectura de la información e identificar las fortalezas y debilidades de los sitios analizados, se aplicaron los principios heurísticos de Hassan y Martín que se consideraron pertinentes al área y otros propuestos por Louis Rosenfeld en el 2004.

- “Evaluación de la usabilidad de un sitio web educativo y de promoción de la salud en el contexto universitario”, es un estudio realizado por (Fernández, et al., 2011). “El objetivo del estudio fue la evaluación de la usabilidad del sitio web www.unidadsalud.es, los resultados obtenidos muestran los errores detectados y su gravedad, proporcionan mejoras que pueden ayudar a la implementación y mejora de la web para que sea un lugar de referencia en materia de salud, a través de la actividad física”. Se aplicó la propuesta de Hassan y Martín para realizar la medición.
- “Evaluación Heurística de las sedes web de los medios digitales: El País y El Mundo”, este estudio realizado por (Roa M., y Caldera J, 2015) aplicó la propuesta de Hassan y Martín. Los resultados fueron óptimos proporcionando las debilidades de usabilidad que debían superarse en los portales web.
- “Evaluación de sitios web multilingües: metodología y herramienta heurística”, un estudio realizado por (Vall M., y Marcos M., 2012). Se propone una metodología de evaluación de sitios web dirigidos a un público multilingüe y multicultural. El resultado es una checklist para la evaluación de sitios web con público internacional, tanto monolingües (con o sin internacionalización) como multilingües (con o sin localización).

Se resalta del modelo de Hassan y Martín su flexibilidad y la capacidad de adaptarse a la evaluación de diferentes entornos de aplicaciones interactivas, según (Suárez, 2011) esta propuesta “es una de las más completas (recoge los aspectos más relevantes a considerar en la usabilidad) y exhaustiva (de cada uno de los aspectos

considerados, proporcionan varios elementos a evaluar)”, además al no proporcionar mecanismos de valoración cuantitativa de la usabilidad, posibilita a los propios evaluadores establecer métricas específicas según su necesidad de análisis y su experiencia.

Por todas las razones anteriormente expuestas, se considera al Modelo de Evaluación Heurística de Hassan y Martín, como la alternativa más adecuada en la implementación del Sistema Experto que se propone en este trabajo de investigación.

c. Métricas de la Evaluación Heurística

Según la propuesta de (Suárez, 2011) las métricas para una evaluación heurística son las siguientes:

Tabla 2.9.

Métricas para evaluación heurística

Tabla de valoración	
Valor	Significado
0	No se cumple en absoluto
10	Se cumple totalmente
NA	Criterio no aplicable en el sitio
NTS	No se cumple en todo el sitio
NEP	No se cumple en los enlaces principales
NPP	No se cumple en la página principal
NPI	No se cumple en alguna página interior
S	Se cumple el criterio

- Los valores de 0 a 10 indican el grado de cumplimiento de cada criterio cuando el mismo debe estar presente en todo el sitio.
- Los valores textuales indican si el criterio se cumple o no y su extensión o alcance en el sitio web, estos valores también tienen un cálculo numérico

dentro de la escala de 0 a 10, así: NTS equivale a 0, NEP equivale a 2.5, NPP equivale a 5, NPI equivale a 7.5 y S equivale a 10.

El cálculo general del **nivel de usabilidad** se determina en un rango de 0 a 100, en donde el significado respectivamente es de nula y máxima usabilidad.

Este tipo de valoración la determinó Suárez basándose en el hecho de que el alcance de la validación de un criterio está en uno de los siguientes dos casos:

- El cumplimiento de un criterio de manera global en todo el sitio.
- El cumplimiento de un criterio en cada una de las páginas del sitio.

c.1. Relación entre heurísticas, sub-criterios y valoraciones.

A continuación se presenta en forma detallada como (Suárez, 2011) propone realizar la valoración de cada uno de los sub-criterios.

La tabla 2.10 describe la heurística de **aspectos generales** y sus respectivos sub-criterios de evaluación, así como el rango de medición que puede tener cada uno al ser evaluado.

Tabla 2.10.

Aspectos Generales.

Aspectos Generales		
COD	SUB-CRITERIO	VALOR
AG.1	Objetivos del sitio web concretos y bien definidos	0 1 2 3 4 5 6 7 8 9 10 NA
AG.2	Contenidos o servicios ofrecidos precisos y completos	0 1 2 3 4 5 6 7 8 9 10 NA
AG.3	Estructura general del sitio web orientada al usuario	0 1 2 3 4 5 6 7 8 9 10 NA
AG.4	Look & Feel general se corresponde con los objetivos, características, contenidos y servicios del sitio web	0 1 2 3 4 5 6 7 8 9 10 NA
AG.5	Diseño general del sitio web reconocible	0 1 2 3 4 5 6 7 8 9 10 NA

CONTINÚA →

AG.6	Diseño general del sitio web coherente	0 1 2 3 4 5 6 7 8 9 10 NA
AG.7	Se utiliza el idioma del usuario	NTS NEP NPP NPI S NA
AG.8	Se da soporte a otro/s idioma/s	NTS NEP NPP NPI S NA
AG.9	Traducción del sitio completa y correcta	NTS NEP NPP NPI S NA
AG.10	Sitio web actualizado periódicamente	NTS NEP NPP NPI S NA

La tabla 2.11 describe la heurística de **identidad e información** y sus respectivos sub-criterios de evaluación, así como el rango de medición que puede tener cada uno al ser evaluado.

Tabla 2.11.

Identidad e Información

Identidad e Información		
COD	SUB-CRITERIO	VALOR
II.1	Identidad o logotipo significativo, identificable y suficientemente visible	0 1 2 3 4 5 6 7 8 9 10 NA
II.2	Identidad del sitio en todas las páginas	NTS NEP NPP NPI S NA
II.3	Eslogan o tagline adecuado al objetivo del sitio	0 1 2 3 4 5 6 7 8 9 10 NA
II.4	Se ofrece información sobre el sitio web, empresa	NTS NEP NPP NPI S NA
II.5	Existen mecanismos de contacto	NTS NEP NPP NPI S NA
II.6	Se ofrece información sobre la protección de datos de carácter personal o los derechos de autor de los contenidos del sitio web	NTS NEP NPP NPI S NA
II.7	Se ofrece información sobre el autor, fuentes y fechas de creación y revisión en artículos, noticias, informes	NTS NEP NPP NPI S NA

La tabla 2.12 describe la heurística de **estructura y navegación**, sus respectivos sub-criterios de evaluación, así como el rango de medición que puede tener cada uno al ser evaluado.

Tabla 2.12.**Estructura y Navegación**

Estructura y Navegación		
COD	SUB-CRITERIO	VALOR
EN.1	Se ha evitado pantalla de bienvenida	NTS NEP NPP NPI S NA
EN.2	Estructura de organización y navegación adecuada	0 1 2 3 4 5 6 7 8 9 10 NA
EN.3	Organización de elementos consistente con convenciones	0 1 2 3 4 5 6 7 8 9 10 NA
EN.4	Control del número de elementos y de términos por elemento en los menús de navegación	0 1 2 3 4 5 6 7 8 9 10 NA
EN.5	Equilibrio entre profundidad y anchura en el caso de estructura jerárquica	NTS NEP NPP NPI S NA
EN.6	Enlaces fácilmente reconocibles como tales	NTS NEP NPP NPI S NA
EN.7	La caracterización de los enlaces indica su estado (visitados, activos)	NTS NEP NPP NPI S NA
EN.8	No hay redundancia de enlaces	NTS NEP NPP NPI S NA
EN.9	No hay enlaces rotos	NTS NEP NPP NPI S NA
EN.10	No hay enlaces que lleven a la misma página que se está visualizando	NTS NEP NPP NPI S NA
EN.11	En las imágenes de enlace se indica el contenido al que se va a acceder	NTS NEP NPP NPI S NA
EN.12	Existe un enlace para volver al inicio en cada página	NTS NEP NPP NPI S NA
EN.13	Existen elementos de navegación que orienten al usuario acerca de dónde está y cómo deshacer su navegación (ej: migas)	NTS NEP NPP NPI S NA
EN.14	Existe mapa del sitio para acceder directamente a los contenidos sin navegar	NTS NEP NPP NPI S NA

La tabla 2.13 describe la heurística de **rotulado** y sus respectivos sub-criterios de evaluación, así como el rango de medición que puede tener cada uno al ser evaluado.

Tabla 2.13.**Rotulado**

Rotulado		
COD	SUB-CRITERIO	VALOR
RO.1	Rótulos significativos	NTS NEP NPP NPI S NA

CONTINÚA →

RO.2	Sistema de rotulado controlado y preciso	NTS NEP NPP NPI S NA
RO.3	Título de las páginas, correcto y planificado	NTS NEP NPP NPI S NA
RO.4	URL página principal correcta, clara y fácil de recordar	0 1 2 3 4 5 6 7 8 9 10 NA
RO.5	URLs de páginas internas claras	NTS NEP NPP NPI S NA
RO.6	URLs de páginas internas permanentes	NTS NEP NPP NPI S NA

La tabla 2.14 describe la heurística de **layout de la página** y sus respectivos sub-criterios de evaluación, así como el rango de medición que puede tener cada uno al ser evaluado.

Tabla 2.14.

Layout de la página

Layout de la Página		
COD	SUB-CRITERIO	VALOR
LA.1	Se aprovechan las zonas de alta jerarquía informativa de la página para contenidos de mayor relevancia	NTS NEP NPP NPI S NA
LA.2	Se ha evitado la sobrecarga informativa	NTS NEP NPP NPI S NA
LA.3	Es una interfaz limpia, sin ruido visual	NTS NEP NPP NPI S NA
LA.4	Existen zonas en blanco entre los objetos informativos de la página para poder descansar la vista	NTS NEP NPP NPI S NA
LA.5	Uso correcto del espacio visual de la página	0 1 2 3 4 5 6 7 8 9 10 NA
LA.6	Se utiliza correctamente la jerarquía visual para expresar las relaciones del tipo "parte de" entre los elementos de la página	0 1 2 3 4 5 6 7 8 9 10 NA
LA.7	Se ha controlado la longitud de página	NTS NEP NPP NPI S NA
LA.8	La versión impresa de la página es correcta	NTS NEP NPP NPI S NA
LA.9	El texto de la página se lee sin dificultad	NTS NEP NPP NPI S NA
LA.10	Se ha evitado el texto parpadeante / deslizando	NTS NEP NPP NPI S NA

La tabla 2.15 describe la heurística de **entendibilidad y facilidad** en la Interacción, sus respectivos sub-criterios de evaluación, así como el rango de medición que puede tener cada uno al ser evaluado.

Tabla 2.15.**Entendibilidad y Facilidad en la Interacción**

Entendibilidad y Facilidad en la Interacción		
COD	SUB-CRITERIO	VALOR
EF.1	Se emplea un lenguaje claro y conciso	NTS NEP NPP NPI S NA
EF.2	Lenguaje amigable, familiar y cercano	NTS NEP NPP NPI S NA
EF.3	Cada párrafo expresa una idea	NTS NEP NPP NPI S NA
EF.4	Uso consistente de los controles de la interfaz	NTS NEP NPP NPI S NA
EF.5	Metáforas visibles reconocibles y comprensibles por cualquier usuario (ej.: iconos)	NTS NEP NPP NPI S NA
EF.6	Si se usan menús desplegables, orden coherente o alfabético	NTS NEP NPP NPI S NA
EF.7	Si el usuario tiene que rellenar un campo, las opciones disponibles se pueden seleccionar en vez de tener que escribirlas	NTS NEP NPP NPI S NA

La tabla 2.16 describe la heurística de **control y retroalimentación**, sus respectivos sub-criterios de evaluación, así como el rango de medición que puede tener cada uno al ser evaluado.

Tabla 2.16.**Control y Retroalimentación**

Control y Retroalimentación		
COD	SUB-CRITERIO	VALOR
CR.1	El usuario tiene todo el control sobre la interfaz	NTS NEP NPP NPI S NA
CR.2	Se informa al usuario acerca de lo que está pasando	NTS NEP NPP NPI S NA
CR.3	Se informa al usuario de lo que ha pasado	NTS NEP NPP NPI S NA
CR.4	Existen sistemas de validación antes de que el usuario envíe información para tratar de evitar errores	NTS NEP NPP NPI S NA
CR.5	Cuando se produce un error, se informa de forma clara y no alarmista al usuario de lo ocurrido y de cómo solucionar el problema	NTS NEP NPP NPI S NA
CR.6	Se ha controlado el tiempo de respuesta	NTS NEP NPP NPI S NA
CR.7	Se ha evitado que las ventanas del sitio anulen o se superpongan a la del navegador	NTS NEP NPP NPI S NA

CONTINÚA →

CR.8	Se ha evitado la proliferación de ventanas en la pantalla del usuario	NTS NEP NPP NPI S NA
CR.9	Se ha evitado la descarga por parte del usuario de plugins adicionales	NTS NEP NPP NPI S NA
CR.10	Si existen tareas de varios pasos, se indica al usuario en cual está y cuantos faltan para completar la tarea	NTS NEP NPP NPI S NA

La tabla 2.17 describe la heurística de **elementos multimedia** y sus respectivos sub-criterios de evaluación, así como el rango de medición que puede tener cada uno al ser evaluado.

Tabla 2.17.

Elementos Multimedia

Elementos Multimedia		
COD	SUB-CRITERIO	VALOR
EM.1	Fotografías bien recortadas	NTS NEP NPP NPI S NA
EM.2	Fotografías comprensibles	NTS NEP NPP NPI S NA
EM.3	Fotografías con correcta resolución	NTS NEP NPP NPI S NA
EM.4	El uso de imágenes o animaciones proporciona algún tipo de valor añadido	NTS NEP NPP NPI S NA
EM.5	Se ha evitado el uso de animaciones cíclicas	NTS NEP NPP NPI S NA
EM.6	El uso de sonido proporciona algún tipo de valor añadido	NTS NEP NPP NPI S NA

La tabla 2.18 describe la heurística de **búsqueda** y sus respectivos sub-criterios de evaluación, así como el rango de medición que puede tener cada uno al ser evaluado.

Tabla 2.18.

Búsqueda

Búsqueda		
COD	SUB-CRITERIO	VALOR
BU.1	Si es necesaria, se encuentra accesible desde todas las páginas del sitio	NTS NEP NPP NPI S NA
BU.2	Es fácilmente reconocible como tal	0 1 2 3 4 5 6 7 8 9 10 NA

CONTINÚA →

BU.3	Se encuentra fácilmente accesible	0 1 2 3 4 5 6 7 8 9 10 NA
BU.4	La caja de texto es lo suficientemente ancha	0 1 2 3 4 5 6 7 8 9 10 NA
BU.5	Sistema de búsqueda simple y claro	0 1 2 3 4 5 6 7 8 9 10 NA
BU.6	Permite la búsqueda avanzada	0 1 2 3 4 5 6 7 8 9 10 NA
BU.7	Muestra los resultados de la búsqueda de forma comprensible para el usuario	0 1 2 3 4 5 6 7 8 9 10 NA
BU.8	Asiste al usuario en caso de no poder ofrecer resultados para una consultada dada	0 1 2 3 4 5 6 7 8 9 10 NA

La tabla 2.19 describe la heurística de **ayuda** y sus respectivos sub-criterios de evaluación, así como el rango de medición que puede tener cada uno al ser evaluado.

Tabla 2.19.

Ayuda

Ayuda		
COD	SUB-CRITERIO	VALOR
AY.1	El enlace a la sección de ayuda, colocado en una zona visible y estándar	NTS NEP NPP NPI S NA
AY.2	Fácil acceso y retorno al/del sistema de ayuda	0 1 2 3 4 5 6 7 8 9 10 NA
AY.3	Se ofrece ayuda contextual en tareas complejas	NTS NEP NPP NPI S NA
AY.4	FAQs (si la hay), correcta la elección como la redacción de las preguntas	0 1 2 3 4 5 6 7 8 9 10 NA
AY.5	FAQs (si la hay), correcta la redacción de las respuestas	0 1 2 3 4 5 6 7 8 9 10 NA

Tal como se planteó al inicio de esta sección, las tablas descritas son parte de la propuesta de (Suárez, 2011) representan para la presente investigación el modelo de evaluación que se va a aplicar para el desarrollo del Sistema Experto ya que complementa el trabajo de Hassan y Martín al plantear las métricas específicas para cada sub-criterio a evaluar.

2.3.5. Sistemas Expertos

Según (Feigenbaum, 1980) un sistema experto es “un programa inteligente de ordenador que utiliza conocimiento y procedimientos inferenciales para solventar

problemas lo suficientemente difíciles que exigen la competencia de un experto humano en su resolución”.

Para (Carrillo, 1987) los sistemas expertos “cuentan con el apoyo del almacenamiento del conocimiento relativo a un campo muy concreto y que, mediante una serie de inferencias, producen la respuesta que un experto en la materia daría”.

Para (Hidalgo, 1998) un sistema experto constituye una tipología de lo que se conoce como Sistema Basado en el Conocimiento, dado que por lo general los sistemas expertos no están en capacidad de adquirir conocimiento por sí solos, suelen incorporar un cuerpo de conocimientos considerable sobre una determinada área del saber que fusionado con otros elementos permiten reproducir la conducta seguida por cualquier experto en dicha área y más concretamente en su actuación habitual a la hora de solventar problemas de su campo de acción.

Se puede mencionar por lo tanto que un sistema experto es un tipo de software que permite realizar el trabajo de un especialista en un campo específico sin que éste se encuentre físicamente presente. Para ello se requiere entonces, almacenar la base del conocimiento del especialista, procesarla mediante hechos, procesos y reglas que permitan inferir respuestas a acontecimientos reales y concretos, tal cual lo haría el especialista.

Desde sus inicios los sistemas expertos han tenido mayor aplicación en el campo de la medicina y la industria en general, sin embargo, actualmente su uso se ha extendido a muchas otras áreas precisamente por su eficacia en la resolución de problemas que de forma ordinaria requieren siempre de la presencia de un perito en el campo de aplicación concreto, lo que implica además mayores riesgos y costos de operación para las organizaciones.

Así mismo de acuerdo con la propuesta de (Hidalgo, 1998) el comportamiento de un sistema experto debe tener ciertas características que lo definan como tal, a continuación se presentan las más notables en relación a este caso de estudio:

- Adquisición y dominio de conocimientos debido al carácter evolutivo y dinámico de la ciencia y de todas las ramas del saber. Sin embargo, es importante aclarar que los sistemas expertos, no tienen aún la capacidad de adquirir conocimiento por sí mismos.
- Resolución de problemas, es decir debe tener la capacidad de explorar, buscar y determinar la posibilidad más apropiada para resolver un problema concreto.
- Ejecución eficaz y eficiente porque debe estar en capacidad de responder a cualquier contingencia con un nivel equivalente al de un experto humano en competencia y tiempo.
- Capacidad de usar un motor de inferencia para realizar sus deducciones, es decir con este elemento el sistema podrá emitir un dictamen final sobre las cuestiones que se le han presentado al usuario en un caso concreto, en base al conocimiento que posee.
- Versátil porque puede ser utilizado incluso por personas que no posean un conocimiento profundo sobre informática ni sobre el campo al que está relacionado. Debe estar diseñado amigablemente de tal forma que cualquier persona que lo use pueda comunicarse adecuadamente con él y viceversa.

a. Arquitectura de un sistema experto

Para determinar la estructura de un sistema experto se ha recogido la propuesta de (Hidalgo, 1998) según la cual, un sistema experto tiene la siguiente arquitectura:

1. Base de conocimiento. Todo el conocimiento específico (experiencia) a disposición del sistema, es de vital importancia para la resolución de problemas en un contexto dado, se concreta en hechos, sucesos, objetos, situaciones y relaciones de un determinado campo del saber. Generalmente se

expresa en forma de reglas expresadas en un lenguaje propio de una herramienta de desarrollo. El conocimiento debe ser representado de la forma más sencilla posible, debe permitir la justificación y la explicación de los procesos desarrollados y de las soluciones intermedias y definitivas y las unidades de conocimiento deben ser totalmente independientes unas con otras.

Se recomienda para estructurar una buena base de conocimiento, responder las siguientes preguntas: a) qué hechos, sucesos, objetos, estrategias y situaciones serán definidas, b) cómo se formularán y procesarán las reglas, y, c) la base de conocimiento hace totalmente referencia a la solución del problema. Debe además, permitir un posterior desarrollo continuado y fácil mantenimiento, es decir, adición, eliminación y modificación de reglas. La potencia de un sistema experto depende del número de reglas que contiene.

2. Base de datos global (memoria de trabajo). Contiene los resultados intermedios basados en el razonamiento efectuado y conclusiones alcanzadas, puede servir para hacer un seguimiento a la evolución de un problema específico, es decir corresponde a la particularidad de un solo caso.
3. Motor de inferencia. Es el componente encargado de controlar y gestionar en forma lógica toda la base de conocimiento. Las funciones más importantes son: a) Controlar cómo las estrategias de búsqueda y razonamiento son utilizadas, determina la secuencia en la que se examinan, eligen y disparan las reglas que componen la base de conocimiento. b) Inferencia, poner en acción un conjunto de estrategias de búsqueda y razonamiento para derivar nuevos hechos a partir de los ya existentes en el programa, de esta manera posibilita el razonamiento.
4. Componente de adquisición. Permite la creación y posterior edición de la base de conocimiento, ayuda al experto a incorporar la base de conocimiento sin la ayuda del ingeniero del conocimiento, permite además supervisar la sintaxis incorporada a las unidades en las que se ha estructurado el conocimiento.

5. Componente explicativo. Brinda la capacidad de explicar, tras un análisis de los procesos seguidos por el motor de inferencia una justificación de cuál ha sido la secuencia que el programa ha seguido hasta alcanzar una determinada solución para el problema planteado. La mayoría de sistemas expertos carecen de mecanismos incorporados para justificar y aclarar de manera adecuada.
6. Interfaz módulo experto. Permite la configuración del sistema, es decir determinar el formato y la estructura de otros componentes de sistema. Se encarga de la adquisición, el mantenimiento, la validación y depuración del conocimiento.
7. Interfaz de usuario. Es el modelo de interacción o el medio de comunicación entre el sistema y las personas encargadas de su explotación.

Esta propuesta es concreta y ajustada a los requerimientos de un sistema experto (SSEE), presenta una estructura base que no deja suelto ningún componente importante, y por otro lado no extiende la estructura dividiendo elementos que por su naturaleza pueden mantenerse juntos o no son necesarios para el SSEE.



Gráfico 2.2: Arquitectura de un SSEE

Fuente: (Hidalgo, 1998)

Otro factor importante de resaltar es que durante el proceso de desarrollo de un SSEE, intervienen tres roles fundamentales:

1. El experto, quien posee el conocimiento especializado en un campo específico.
2. El ingeniero de conocimiento, quien extrae el conocimiento, ayuda al experto a divulgar el conocimiento y generar la base de conocimiento.
3. El usuario, quien usará el SSEE.

b. Metodología para el desarrollo de Sistemas Expertos

Tomando en cuenta que el desarrollo de un SSEE puede ser similar al desarrollo de cualquier otro producto de software, incluso en el hecho de que ciertos problemas suelen surgir durante cualquier etapa del proceso, es concluyente que el desarrollo de un SSEE depende de su contexto particular.

En base a la propuesta presentada por (Hidalgo, 1998) se definen las siguientes fases de desarrollo para un sistema experto, las mismas se apoyan en un esquema común que puede variar en contenido y ordenación dependiendo del tipo, la necesidad y el tamaño del sistema a desarrollar.

- Fase I. Definición del problema.

Antes del desarrollo de cualquier programa es imprescindible detallar la problemática que se pretende resolver, se debe comprobar si efectivamente el SSEE contribuirá de manera eficiente en la resolución del problema alcanzando todas las metas programadas. No siempre la forma más eficiente de resolver un problema es a través de un sistema experto, es por esto que muchos especialistas manifiestan que antes de intentar abordar extensamente un proyecto de esta magnitud es mejor plantear otras soluciones alternativas.

Es importante definir si la complejidad del problema amerita de una técnica inteligente para su resolución, según (Hidalgo, 1998) entre los casos de esta naturaleza que se pueden citar están los problemas en los que existe la necesidad de contar con un experto humano, además cuando el problema está bien definido y el conocimiento que se requiere para la solución es finito y es posible plantear diversas alternativas de solución.

En esta etapa se necesita determinar lo siguiente:

- Las tareas específicas que el software deberá ejecutar.

- Las modalidades y cantidad de conocimiento
- Los recursos físicos y humanos con lo que se cuenta
- Conceptos principales con los que se debe trabajar en el contexto concreto
- Viabilidad y coste del proyecto
- Plan completo de ejecución

- **Fase II. Búsqueda de la fuente de conocimiento**

En esta etapa se requiere extraer la información con la cual se va a concretar la base de conocimientos del SSEE, generalmente la fuente que ayuda en la consecución de este objetivo es un experto o especialista en el área específica sobre la cual se va implementar el sistema. Es importante entender que la fuente debe garantizar la acumulación de conocimiento en cantidad y calidad, de tal manera que el sistema pueda ser desarrollado apropiadamente, en otras palabras el experto debe conocer sobre el tema, tanto el fundamento teórico como su aplicación en la práctica, y debe entender además la dimensión de las actividades necesarias para el desarrollo del SSEE y de sus dificultades sobre todo en aquellas actividades que tengan que ver con la adquisición del conocimiento, la representación del mismo y el procesamiento de los datos obtenidos.

Además del conocimiento y la experiencia, las personas involucradas como expertos deben tener la habilidad de articularlo y transmitirlo de forma adecuada; es decir, ser capaces de definir con claridad todos los métodos, técnicas y procedimientos a utilizar en la resolución de todos los problemas que se puedan presentar en su especialidad.

- **Fase III. Identificación de conceptos y datos clave**

En esta etapa es necesario profundizar el conocimiento de la problemática a resolver, el ingeniero de conocimiento y el experto debe realizar un análisis conjunto sobre la temática concreta de estudio, el alcance de este trabajo es llegar a explicar las

relaciones entre los objetos y procesos más relevantes dentro del contexto del problema tratado y entre los datos disponibles, su carácter de volatilidad o de permanencia, su consistencia, precisión y fiabilidad.

Una de las técnicas más apropiadas para conseguir identificar los conceptos y datos clave es la división del problema en varios sub-problemas de tal manera que el análisis sea menos complejo y más fácil y concreto.

Este proceso termina cuando en conjunto el ingeniero del conocimiento y el experto, deciden que se ha conseguido identificar todos los principios, conceptos, hechos y relaciones que conforman el tema tratado. Y por otra parte cuando han logrado definir la estructura del sistema a través del diseño de su arquitectura.

- **Fase IV. Selección de las herramientas de hardware y software**

Esta es una de las etapas más importantes del ciclo de desarrollo de un SSEE. Desde el inicio se deben tomar en cuenta las tareas que el sistema debe desarrollar además de la estructura genérica del conocimiento y el tipo de inferencia que el experto utiliza.

La selección de las herramientas de hardware y de software implica el lenguaje de programación a utilizar para el desarrollo y otros programas que se utilizarán como apoyo o ayuda para el ingeniero del conocimiento, además del soporte físico requerido para el funcionamiento del sistema; en ambos casos es recomendable especificar a detalle los requisitos técnicos de las herramientas a utilizar, tomando en cuenta sobre todo, la compatibilidad entre estos dos tipos de recursos.

Comprobación del Sistema e integración del mismo en Producción:

- Integración a mayor escala.
- Uso regular y evaluación del Sistema Experto
- Mantenimiento y mejora del mismo.

Adquisición del conocimiento.

- Representación del conocimiento.
- Desarrollo de un prototipo.
- Uso y mejora del prototipo.
- Desarrollo del diseño detallado -final.
- Implantación del núcleo del Sistema.
- Expansión de la base de conocimiento.
- Adecuación de la interfaz del usuario.
- Monitorización de las prestaciones y evaluación del Sistema.
- Mantenimiento de la Base del Conocimiento.

Las herramientas de software que tienen que elegir entonces dependen de factores como la complejidad del problema, del equipamiento físico que se haya dispuesto o con el que se cuente, el dominio de las mismas que tenga el ingeniero del conocimiento pues si no es una persona con experiencia en el desarrollo de SSEE puede ayudarse de sistemas vacíos o shells para aprovechar la estructura y componentes que poseen estos programas para la creación de programas de inteligencia artificial y si no se cuenta con este tipo de herramienta se debe utilizar un lenguaje de programación de alto nivel.

En todo caso siempre es necesario evaluar las bondades y restricciones que tienen las herramientas que se van a utilizar y contrastar los resultados para definir las mejores alternativas. Las siguientes preguntas que propone (Hidalgo, 1998) se pueden realizar a la hora de buscar el software a utilizar.

- ¿Cuál es el límite máximo de reglas que acepta?
- ¿Qué estrategias de inferencia utiliza?
- ¿Qué tipo de representación del conocimiento será necesario utilizar en la resolución del caso planteado?

Las respuestas a estas preguntas planteadas guiarán la selección de las herramientas adecuadas. Sin embargo se debe recordar que lo importante es desarrollar un SSEE que satisfaga las necesidades de los usuarios y este criterio es completamente independiente de las herramientas que se utilicen para la implementación del mismo.

- **Fase V. Adquisición del conocimiento I.**

Esta etapa está enfocada a la adquisición del conocimiento y su representación, se empieza a realizar el primer prototipo del SSEE, es lo que se conoce como ingeniería del conocimiento. Tanto la adquisición y la representación de éste son dos etapas diferentes pero que sin embargo están intrínsecamente unidas debido a que para iniciar el desarrollo del prototipo inicial ya se debe haber recogido parte del conocimiento sustancial del experto, haberlo organizado y sistematizado para determinar las posibilidades de representación automatizada.

Esta etapa determina la base de conocimiento del SSEE, tiene un carácter de repetitiva y cíclica ya que se trata de adquirir todo el conocimiento posible que posee el experto sobre el campo temático específico en forma evolutiva, es decir gradualmente, para con ella ir mejorando el prototipo hasta obtener el sistema final. Los pasos a seguir serían de la siguiente manera:

- Se crea el primer prototipo
- Se evalúan los resultados sobre las técnicas utilizadas para resolver las problemáticas planteadas
- Si las técnicas son las adecuadas se inician las pruebas respectivas, caso contrario se debe regresar al prototipo.

Como se ve, no es lógico en un primer momento integrar toda la base de conocimiento, es importante hacerlo solo con lo que se considere oportuno en esta etapa.

Así mismo el papel que cumple el ingeniero del conocimiento en esta etapa es fundamental ya que le puede resultar tremendamente complicado entender la estructura del conocimiento que debe sistematizar, por ello es importante que aparte de contar con el apoyo del experto también se capacite sobre el tema de tal manera que conozca con cierto nivel de profundidad sobre el ámbito de análisis.

- **Fase VI: Representación del conocimiento y formalización del razonamiento.**

Esta es una de las etapas que mayor tiempo y esfuerzo requiere pues se trata de entender y luego sistematizar el conocimiento del experto, que consiga a través de la transformación de los datos iniciales de un problema en una solución apropiada para el mismo. Según (Hidalgo, 1998) la mejor forma de representar el conocimiento es a través de hechos y fenómenos, y de procedimientos o de reglas; para manipular los hechos se utilizan estructuras de control que determinan el cómo y cuándo aplicar las reglas y/o procedimientos.

La representación del conocimiento se centra en encontrar un esquema que permita determinar los rasgos esenciales del ámbito correspondiente y hacer accesible esta información a un procedimiento de resolución específico, es importante entender que cada tipo de problema requiere un tipo diferente de razonamiento y éste a su vez requiere una representación específica.

El proceso básico sería: El tipo de problema determina el tipo de razonamiento y éste a su vez, determina la representación específica a plantearse.

Existen diversas alternativas de representación del conocimiento, cada una de ellas ofrece un potencial diferente de eficiencia y complejidades según el problema que se pretende resolver y desde luego, proporcionan efectos desiguales a la operatividad del sistema, los mismos que se pueden apreciar en la rapidez de procesamiento y la facilidad de uso del mismo.

Los parámetros a tomar en cuenta a la hora de seleccionar un esquema de representación del conocimiento son:

- Adecuación lógica
- Poder heurístico
- Que la notación no sea excesivamente compleja

La representación del conocimiento pretende imitar los comportamientos generalmente utilizados por el ser humano cuando se enfrenta a la solución de un problema específico y la solución depende en gran medida del tipo de razonamiento que se haya implementado. En el caso de los SSEE el proceso tiene los siguientes pasos:

- La clasificación y organización de los conocimientos recogidos, es importante utilizar estrategias como la jerarquización y los árboles de decisión para estructurar adecuadamente esa información.
- El conocimiento debe ser segmentado en trozos, cada uno de los cuales se convierten en reglas, redes semánticas o cualquier otra estrategia de representación que se haya seleccionado.

Como se puede apreciar, el trabajo en esta etapa es completamente colaborativo entre el ingeniero del conocimiento y el experto. El experto es quién va revisando a detalle la representación del conocimiento para identificar y establecer las modificaciones correspondientes, el ingeniero de conocimiento en cambio, tiene la responsabilidad de clasificar el conocimiento y el razonamiento realizado por el experto, en donde se pone en evidencia cómo éste utiliza los conceptos, cómo infiere los hechos, cómo plantea hipótesis y cómo comprueba esas hipótesis, cómo razona en el caso de incertidumbre o en el caso de contar con la información incompleta, imprecisa e inconsistente. El hecho es encontrar la estrategia de solución de los problemas no sólo en situaciones normales o comunes sino además cuando tiene que escoger de forma inteligente, entre varias alternativas, la mejor para el caso tratado.

- **Fase VII: Desarrollo del prototipo, pruebas y validación.**

Es importante que el trabajo inicial sea una versión elemental, que maneje solo una porción reducida del conocimiento, de tal manera que cuando sea validada se convierta en parte de otras versiones más completas y complejas, esta versión podrá estar constituida por el motor de inferencia, una parte de la base del conocimiento y la interfaz de usuario.

Se plantea de esta forma el inicio del desarrollo, debido a que las diferentes pruebas que se realizarán sobre este prototipo, determinarán si el diseño levantado, el nivel de tratamiento y la representación del conocimiento están siendo adecuados para el problema. Así mismo, permitirá la depuración y contrastación de la terminología, los conceptos empleados y las estrategias de resolución implementadas; sobre todo, será posible la detección temprana de errores. Luego de esta etapa si aún no se ha alcanzado la eficacia resolutoria esperada, será más fácil continuar con otros ensayos y simulaciones.

Este proceso es continuo e iterativo, la idea fundamental es que a medida que el sistema sea capaz de resolver los casos más simples, poco a poco se irá incrementando el nivel de complejidad de los casos a resolver.

- **Fase VIII: Adquisición del conocimiento II**

En esta etapa se procederá en forma incremental a completar todas las funcionalidades del SSEE, se pueden manejar varias estrategias que ayudarán a nutrir la base de conocimiento como por ejemplo: entrevistas, observaciones directas, exámenes y pruebas, informes del experto, análisis inductivos, entre otras.

De las estrategias mencionadas, las más apropiadas para el caso específico que se va a tratar en esta investigación son:

- Informes de Expertos porque van a permitir sustentar conocimiento basado en estudios referidos a la evaluación de la usabilidad de aplicaciones web.
- Análisis inductivos porque a partir de casos resueltos por los especialistas, se extrae el conocimiento empleado por ellos para la resolución de los mismos.

Adicionalmente es necesario contar con la suficiente literatura científica y técnica que permita fundamentar la investigación y el desarrollo del SSEE, así como el acceso a las bases de datos y las memorias de los resultados empíricos de los casos previos relacionados.

- Fase IX: Mantenimiento y actualización

Al igual que un especialista, es importante que un SSEE tenga la posibilidad de ser actualizado para mantener su efectividad, por lo tanto la características de acumular y asimilar nuevo conocimiento es imprescindible. Esto significa concretamente, que la base de conocimiento de un SSEE puede ser incrementada o modificada cuando sea necesario y por lo tanto no es procedente hablar de una versión terminada o fina del sistema.

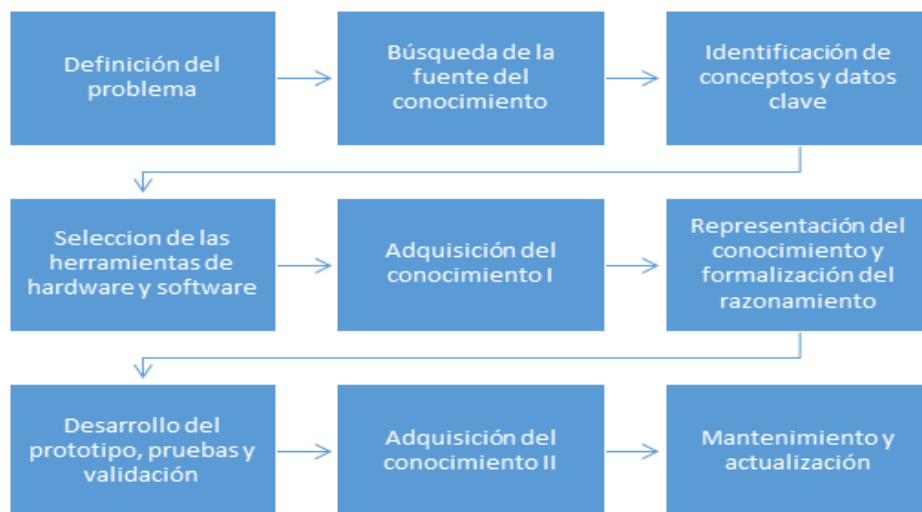


Gráfico 2.3: Etapas de desarrollo de un SSEE

Fuente: (Hidalgo, 1998)

2.4. Antecedentes Contextuales

Aproximadamente desde la década de los 80 se empieza a pensar en usabilidad, es una época en la que la industria del software crece obligada por el surgimiento de la industria del hardware, por lo tanto surge la demanda de los usuarios por utilizar sistemas más eficientes con la capacidad de ejecutar tareas con mayor facilidad, esto promueve la creación de normas de diseño para las interfaces y para mejorar la usabilidad. Las aplicaciones se vuelven más intuitivas y participativas para el usuario, porque se formaliza la utilización de modelos y métodos de desarrollo que incluyen desde ya ciertas consideraciones de usabilidad.

Aplicar técnicas de usabilidad en el diseño de sistemas web significa según estadísticas de (Mycrosystems; Villa, 2003) lo siguiente:

- “Cambios en el diseño realizados en IBM a través de métodos de usabilidad resultan en una reducción de una media de 9.6 minutos por tarea, generando unos ahorros internos de 6,8 millones de dólares”. (Karat 1990).
- “La usabilidad demuestra reducciones del ciclo de desarrollo de los productos de 33-50%”. (Bosert 1991).
- “La empresa Ricoh ha descubierto que el 95% de los usuarios encuestados nunca utilizan las tres características claves diseñadas para hacer más atractivo el producto bien por desconocer su existencia, por no saber cómo utilizarlas o no entenderlas”. (Nussbaum and Neff 1991).
- “El 63% de todos los proyectos de software sobrepasan su presupuesto, siendo las cuatro causas más importantes relacionadas con usabilidad”. Estas causas según (Lederer and Prasad, 1992) son: cambios frecuentes en los requerimientos, tareas olvidadas, falta de comprensión de los requisitos por parte de los propios usuarios, falta de comunicación y comprensión entre el usuario y el analista.

A partir del año 2000 gracias a las aportaciones de Jacob Nielsen la ingeniería de la usabilidad ha adquirido especial importancia en el desarrollo de proyectos de software, hoy se toma en cuenta al usuario en todas las etapas del proceso porque se entiende que es la única forma de crear productos que satisfagan sus necesidades y sus expectativas a la hora de interactuar con las aplicaciones.

Según (Nielsen, 2012) las mejores prácticas actuales requieren gasto de alrededor de 10% del presupuesto de un proyecto de diseño, en la usabilidad, en promedio, esto será más que duplicar las métricas de calidad deseados de un sitio web. Para el software y los productos físicos, las mejoras son más pequeñas, pero sigue siendo considerable, cuando se enfatiza el proceso de diseño en la facilidad de uso.

Todo esto significa que el interés por tener aplicaciones más usables crece entre todos los involucrados, las exigencias desde el lado de los usuarios es evidente cuando prefieren productos que no exijan complicaciones de eficacia, eficiencia y satisfacción, y desde el lado de la industria del software por generar aplicaciones cada vez más fáciles de usar, de comprender y de recordar; esta es la razón por la que se han creado varios estándares, normas y métodos que ayudan en la aplicación de la ingeniería de la usabilidad en el desarrollo de software.

La evaluación heurística representa un método de evaluación de la usabilidad, tal como se ha mencionado en párrafos anteriores es uno de los métodos sin usuarios más utilizados debido a su eficiencia, ya que permite realizar evaluaciones a sitios específicos en corto tiempo y bajo costo. Las únicas condiciones de uso serían: 1) contar con un experto en la materia y 2) la adquisición de una herramienta tecnológica que permita la automatización del proceso de evaluación. En cuanto a la primera condición no representa un problema ya que existen varias propuestas de heurísticas generadas por expertos en usabilidad y que actualmente se aplican en evaluaciones.

El objetivo de la mayoría de estudios de usabilidad, están orientados a encontrar fallas en el diseño de interfaces que involucren dificultad para los usuarios. Se ha investigado a nivel del Ecuador sobre qué tipo de trabajos se han realizado y se puede mencionar al respecto que los pocos que se han hecho no se enfocan a la evaluación heurística sino a la evaluación de la usabilidad en general. A continuación se presentan algunas de estas investigaciones, como argumento a lo mencionado en esta sección:

El estudio realizado por (Acuña Herrera & Vasco Amores, 2012) “Análisis de Usabilidad y Accesibilidad en Sistemas de Información Utilizados por Personas con Capacidades Especiales”, es una investigación hecha por dos tesis de la Maestría en Ingeniería de Software de la ESPE, que se orienta en el desarrollo de un test de usabilidad sobre un aplicativo de realidad virtual que está dirigido a usuarios con discapacidades. Tal como se aprecia, se refiere a un tipo de evaluación con usuarios y no desarrolla ninguna aplicación informática para realizar la medición. Por lo tanto no tiene similitud con el presente proyecto.

El estudio realizado por (Salazar Jácome, 2013) “Marco de integración de la usabilidad en el proceso de desarrollo de software en las principales empresas desarrolladoras del Ecuador (Quito, Guayaquil y Cuenca). LATACUNGA/ESPE/2013”. Hace un análisis de todas las técnicas de Interacción Persona-Ordenador (IPO) y propone un marco de integración de la usabilidad en el proceso de desarrollo, para ello el ingeniero de software tendrá que elegir según el tipo de proyecto, la más adecuada de entre 16 técnicas IPO. Esta investigación tampoco es similar al presente trabajo, sin embargo aporta con datos estadísticos importantes para la fundamentación teórica del mismo.

Otro ejemplo es la investigación “Metodología de medición y evaluación de la usabilidad en sitios web educativos”, realizada por (Alva Obeso, 2009), este estudio se centra en el desarrollo de una metodología de evaluación de la usabilidad de sitios

web educativos, incluye la perspectiva de los expertos y la de los usuarios, se puede realizar tanto en la etapa de prototipo como en la versión final del producto. De igual manera, esta investigación tampoco guarda similitud con el presente estudio, ya que se enfoca en dos tipos de evaluación y además involucra a los usuarios en la medición de la usabilidad de las aplicaciones.

2.5. Conclusiones

- La usabilidad representa un factor de calidad del software, es incorporada al desarrollo de aplicaciones informáticas a partir de la década de los 80 cuando la industria de software experimenta un marcado despliegue precisamente porque aparece la necesidad de mejorar la experiencia de los usuarios en la interacción con las aplicaciones.
- En el afán de los desarrolladores y diseñadores de mejorar esta experiencia de los usuarios y crear aplicaciones más fáciles de usar y de comprender, nacen métodos que permiten evaluar estas características, entre estos métodos, está la evaluación heurística como un método de inspección que no requiere la participación de los usuarios en la evaluación, sino la de expertos en el campo de usabilidad que realizan la inspección y determinan los resultados correspondientes.
- De toda la bibliografía analizada se puede concluir que la evaluación heurística es uno de los métodos de evaluación más utilizados, ha demostrado ser efectivo para la medición de la usabilidad de todo tipo de aplicaciones. En relación a su aplicación existen varias propuestas que guían el proceso, sin embargo no es fácil contar con la participación de expertos en este campo, sobre todo en este medio (nuestro país) en el que poco se conoce acerca de profesionales especialistas en usabilidad, por esta razón es importante recoger las propuestas generadas, tratar ese conocimiento y desarrollar una aplicación

que automatice el proceso de evaluación, sin la necesidad de contar con un experto en el área.

CAPÍTULO III

DESARROLLO DEL SISTEMA EXPERTO DE EVALUACIÓN HEURÍSTICA

3.1. Introducción

El presente capítulo describe el proceso de construcción del sistema experto, inicia con la definición de la metodología a utilizar y finaliza con el conjunto de pruebas que se realizarán al prototipo final del sistema. El ciclo de desarrollo cuenta con 9 fases, cada una de las cuales tiene objetivos precisos que se van concretando a través de la ejecución de una serie de actividades que generan artefactos verificables que permiten la evolución del sistema hasta su versión en producción. Una de las fases intermedias es la generación del primer prototipo, representa una etapa esencial porque permite valorar si las estrategias utilizadas para la solución del problema correspondiente, son adecuadas; en el caso de no serlo, se puede proceder con otro tipo de solución, a fin de encontrar la estrategia apropiada. Si por el contrario, la solución es idónea, el proceso continúa siguiendo la misma estrategia.

3.2. Definición de la metodología de desarrollo del Sistema Experto (SSEE)

La metodología que se va utilizar en el desarrollo de SSEE es la definida por (Hidalgo, 1998), esta metodología plantea un esquema genérico adaptable y flexible de acuerdo al alcance del problema y sus características particulares.

La descripción de esta metodología se encuentra en el Capítulo II de este trabajo, sección 2.3.5, literal b.

3.3. Desarrollo del sistema

Nombre del Sistema: “EXSHE” (Expert System Of Heuristic Evaluation)

El nombre del Sistema proviene del acrónimo que significa “Sistema Experto de Evaluación Heurística”, expresado en el idioma Inglés, fue seleccionado de varias alternativas porque es corto, preciso y de fácil memorización.

3.3.1. Fase I: Identificación del problema

a. Detalle de la problemática a resolver y justificación

Tal como se analizó en la problematización y se fundamentó en el marco teórico del presente trabajo, la usabilidad de las aplicaciones en esta época es un parámetro imprescindible de considerar no solo para los desarrolladores y diseñadores de interfaces gráficas, sino además para los usuarios quienes cada vez exigen productos de software más fáciles de usar y de comprender. En tal virtud, el sistema que se propone desarrollar va a permitir evaluar la usabilidad de todo tipo de aplicaciones web durante cualquier fase de su ciclo de vida, esta medición se realizará a través del método de evaluación heurística que plantea la verificación de un conjunto de reglas tales como: aspectos generales, identidad e información, estructura y navegación, rotulado, entre otras; que una vez comprobadas, el sistema generará el informe correspondiente en el cual se inferirá el nivel de usabilidad del sitio y un detalle con la lista de problemas y recomendaciones de los aspectos de diseño que se deben mejorar.

Es una necesidad el contar con este tipo de herramientas de apoyo para la creación de aplicaciones que satisfagan las necesidades de los clientes, si bien estas herramientas existen, generalmente están en línea y son propiedad de empresas que se dedican a realizar asesoría tecnológica en general, lo que implica que si se utilizan las mismas, en algún momento se deberá devolver el servicio bajo cualquier condición.

Además existen otras herramientas disponibles que son creadas en aplicaciones ofimáticas como Excel (Por ejemplo: ExpertReviewCheckpoints-Spanish.xls, disponible en la web: <http://www.userfocus.co.uk/resources/guidelines.html>) que no son completamente automatizadas y no ofrecen resultados eficientes en su totalidad.

Se debe también analizar el ámbito local, en este medio es poco lo que se conoce de usabilidad y no existen especialistas en este campo específico, es decir, son escasos o nulos los servicios de asesoría a los cuales se puede acudir. Además que de contratar los servicios de asesoría, los costos de producción se incrementan, razón por la cual, los equipos de desarrollo prefieren no contratarlos.

Tomando en consideración todo lo expuesto en las secciones anteriores, el desarrollo de un sistema experto, es una alternativa completamente viable, en primera instancia porque se recogerá el conocimiento de expertos en usabilidad y se aplicará uno de los métodos más utilizados a nivel mundial como es la Evaluación Heurística; este método requiere un proceso que no necesita de usuarios ni de un laboratorio dedicado, pues solo a través de la indagación de especialistas, se generan los informes y las recomendaciones respectivas, razón por la que los costos de implementación son bajos.

El SSEE además brindará la posibilidad de ser usado por cualquier profesional del desarrollo de software, no requiere de conocimiento específico ni especializado, brindará soporte en el proceso de desarrollo porque permitirá medir la usabilidad de las aplicaciones rápidamente y podrá ser utilizado para evaluar productos de software desde las fases tempranas, es decir en los primeros prototipos, así como podrá ser utilizado para evaluar productos ya en producción.

b. Dominio del problema

EXSHE, será un sistema encargado de realizar la evaluación de la usabilidad de aplicaciones web, su objetivo se centra en identificar fallos de diseño en las interfaces que ocasionen dificultad en su uso y comprensión.

Tal como se ha mencionado en secciones anteriores del presente trabajo, la validación del sistema se realizará en los EVA de la Universidad Técnica de Cotopaxi, esta entidad pública que tiene como misión fundamental la formación de profesionales en 18 carreras diferentes, cuenta con una plataforma MOODLE sobre la cual funcionan los entornos virtuales de aprendizaje, esta plataforma está instalada en el Departamento de Servicios Informáticos y actualmente es administrada por el Ing. Gonzalo Borja Almeida, docente de dicha universidad.

Por otra parte, los EVA son aplicaciones tipo web cuya funcionalidad específica es servir de apoyo al proceso de enseñanza – aprendizaje, en donde los docentes tienen la posibilidad de compartir variedad de elementos académicos y colocar tareas y otras actividades para que los estudiantes las desarrollen, las aplicaciones permiten recoger todas las evidencias de la participación de los estudiantes durante el proceso, lo que a la vez facilita que los docentes valoren cada aporte que realizan.

El sistema ofrecerá la posibilidad de ser manipulado por cualquier profesional del desarrollo de software y del diseño de interfaces, la evaluación consistirá en la verificación de un conjunto de principios (heurísticas), cada uno de los cuáles cuenta a su vez, con factores que permitirán determinar su nivel de cumplimiento en la interfaz evaluada.

El sistema será flexible en cuanto al número de evaluadores para un proyecto, número de sesiones por proyecto y número de interfaces evaluadas.

Es necesario aclarar que no se pretende con este trabajo medir procesos metodológicos de enseñanza – aprendizaje, esto implicaría un estudio adicional sobre

pedagogía y metodología de la educación, caso que no tiene que ver con el objetivo perseguido por esta investigación.

c. Tareas del Sistema

La especificación de las tareas del sistema, están basadas en la publicación de Jakob Nielsen acerca de cómo se realiza una evaluación heurística, realizada en el año 1995.

- T001: El sistema debe permitir el registro y administración de los datos informativos de la aplicación web a ser evaluada, (proyecto de evaluación)
- T002: El sistema debe permitir que el proyecto tenga al menos 4 evaluadores.
- T003: El sistema debe permitir que un evaluador pueda tener varias sesiones de evaluación.
- T004: El sistema debe inferir el nivel de usabilidad de la aplicación evaluada.
- T005: El sistema debe generar un informe de evaluación por evaluador.
- T006: El sistema debe generar el informe consolidado de la evaluación realizada, tomando en cuenta todas las sesiones de un evaluador y todas las evaluaciones del proyecto.
- T007: El sistema debe generar recomendaciones para mejorar la usabilidad de la aplicación evaluada.
- T008: El sistema debe permitir obtener reportes históricos de las evaluaciones realizadas.
- T009: El sistema debe permitir obtener la memoria de trabajo de una evaluación específica.

T010: El sistema debe permitir la gestión de la información del usuario evaluador.

d. Modalidades de conocimiento a utilizar

De acuerdo a la propuesta realizada por (Carrillo, 1998), las modalidades de conocimiento que se van a utilizar en el desarrollo de EXSHE, son: conceptos y relaciones, procedimientos de rutina, hechos y heurísticas y conocimiento clasificatorio.

A continuación se detalla, cómo se puntualizan cada una de las modalidades de conocimiento en este trabajo:

- Conceptos y relaciones: Glosario de los principales conceptos de la evaluación heurística y sus relaciones.
- Procedimientos de rutina: Técnicas y procedimientos de trabajo a partir de la experiencia de los expertos en el campo de la evaluación heurística.
- Hechos y Heurísticas: Las Reglas prácticas que se aplican en la solución de tareas, esto es, la lista de principios heurísticos y sub-heurísticos que se van a aplicar en la medición de la usabilidad de las aplicaciones.
- Conocimiento clasificatorio: Proceso de refinamiento de conceptos que permiten distinguir entre percepciones parecidos.

e. Recursos físicos y humanos

Recursos Humanos:

- Expertos en usabilidad: Jakob Nielsen, Yusef Hassan Montero, Francisco Martín Fernández, María del Carmen Suárez Torrente.

- Profesionales en el campo del desarrollo de software del medio. (Ver Tabla 3.2)
- Ingeniera del conocimiento: Ing. Ms. C. Verónica Tapia
- Director del proyecto: Ing. Ms. C. José Luis Carrillo

Recursos Físicos:

- Computador HP, procesador Intel Core 2, 2.13 GHz, RAM 2.00 GB, HD 500GB.
- Laptop HP Pavilion g4-1085la Notebook PC, Core i5, 8GB RAM, HDD 640GB.
- Impresora

f. Definición de acrónimos

- EXSHE: Expert System Of Heuristic Evaluation
- EVA: Entornos de evaluación de aprendizajes
- SSEE: Sistema experto
- EH: Evaluación heurística
- ISO: Organización Internacional de Normalización
- IEC: Comisión Electrotécnica Internacional
- IEEE: Instituto de Ingenieros Eléctricos y Electrónicos
- UML: Lenguaje de Modelado Unificado
- UTC: Universidad Técnica de Cotopaxi
- NA: Criterio no aplicable en el sitio
- NTS: No se cumple en todo el sitio
- NEP: No se cumple en los enlaces principales
- NPP: No se cumple en la página principal

- NPI: No se cumple en alguna página interior
- S: Se cumple el criterio
- NA: No aplicable
- R: Restricciones
- RQF: Requisito funcional
- RNF: Requisitos no funcionales
- T: Tarea
- CP: Caso de prueba

g. Viabilidad y coste del proyecto

- Viabilidad del proyecto: El proyecto es viable porque reúne las condiciones técnicas y operativas que van a permitir el cumplimiento de los objetivos. Cuenta con el recurso humano necesario para la ejecución del mismo, y la información técnica y de soporte se encuentra disponible y al alcance de la investigadora. Una vez terminado el proyecto, éste será validado en una institución pública que ha brindado todo el apoyo e interés requeridos para la consolidación de la presente investigación.
- Viabilidad técnica para la construcción de un sistema experto: Según el argumento de Silvia Acuña, en su libro "MPAS: Metodología participativa para un sistema basado en conocimiento aplicado al cultivo de algodón", publicado en el 2003. Los sistemas convencionales "...tratan con problemas "bien-estructurados", es decir aquellos problemas que tienen una solución algorítmica, requisitos objetivos que no necesitan una interpretación subjetiva, y donde la incertidumbre del éxito del sistema en la solución del problema es baja". Por el contrario, "cuando los problemas a ser solucionados están mal estructurados y tienen requisitos subjetivos, donde la incertidumbre, la complejidad, la inconsistencia y la imprecisión

conjunta o individual, impregnan los elementos con los cuales una persona tiene que trabajar para desarrollar un sistema a fin de solucionar estos problemas y cuando no hay soluciones algorítmicas y/o de investigación operativa, se requiere diseñar un sistema basado en conocimiento, en general, y sistemas expertos, en particular”. Por lo tanto, siendo el problema que se pretende corregir un ejemplo concreto de situaciones para las cuales las soluciones están dadas en relación a los criterios subjetivos de los involucrados (evaluadores y usuarios), además porque no existen procedimientos comprobados totalmente que certifiquen la validez de resultados en forma concreta y objetiva, y sobre todo por la complejidad que representa el desarrollo de un sistema que deduzca el conjunto de problemas y recomendaciones específicas acerca de la usabilidad de una aplicación como lo hiciera un experto humano; se puede concluir que es viable el desarrollo del software que se propone como un sistema experto, pues cumple con las características necesarias para la implementación de un sistema de este tipo.

- El costo de la implementación del proyecto se ha estimado en \$ 21.260 USD, dentro de este valor se incluyen los siguientes gastos: papelería \$ 300, internet \$ 200, fotocopias \$ 200, recolección y análisis de la Información \$ 2.100, desarrollo del software \$ 16.200, recursos tecnológicos de hardware \$ 1.000, gastos administrativos \$ 1.260. Como se puede observar, el rubro más alto representa el cálculo del total de horas de trabajo de la autora de la investigación; este valor no implica inversión económica efectiva, razón por la que se concluye que el proyecto es viable económicamente.

h. Plan de ejecución del SSEE

A continuación en la tabla 3.1 se describe el plan de ejecución para el desarrollo del sistema experto.

Tabla 3.1.

Plan de ejecución del SSEE

Semanas	FECHA		Horas / Semana	Número / Días	Horas / Día	Descripción de tareas a realizar	Resultados
	Inicio	Fin					
1	26 de Mayo	31 de Mayo	10	3	3,33	Preparación/Instalación de herramientas (Java, Eclipse, Window Builder, JavaDb), entorno de desarrollo, Pruebas con JESS dentro de eclipse.	Herramientas de software requeridas para el desarrollo instaladas y configuradas
2	1 de Junio	7 de Junio	20	7	2,86	Modelado de Persistencia, Jerarquización de la aplicación, Conexión(Presentación- Aplicación- Datos), Interfaces Gráficas (Preliminares/Navegabilidad), Scripts de Bdd.	Prototipo de las interface de usuario y modelo físico de la base de datos
3	8 de Junio	14 de Junio	20	7	2,86	Clase Evaluador (Registro: Validación, Redundancia, Verificación de password, Autenticación, Pruebas). Incorporación de Heurísticos y Sub-heurísticos en la capa de Persistencia.	Funcionalidades: Añadir evaluador e iniciar sesión. Tablas de heurísticas y sub-heurísticas implementadas
4	15 de Junio	21 de Junio	20	7	2,86	Clase Evaluación(Registro: Validación de Campos, Persistencia, Relación con el Evaluador, Visualización, Modificación, Eliminación)	Clase evaluación implementada
5	22 de Junio	28 de Junio	20	7	2,86	Ejecución de la Evaluación(Presentación de Heurísticos/sub-heurísticos para Calificarlos, Almacenamiento de Resultados , Manejo de Sesiones)	Interfaz de ejecución de evaluación y calificación. Funcionalidad de editar evaluaciones
6	29 de Junio	5 de Julio	20	7	2,86	Calificación de la Evaluación en función de la Base de Conocimiento, Conexión de la Capa de Persistencia con JESS (Análisis de las calificaciones de cada evaluación para obtener Resultados)	Funcionalidad de informe de evaluación individual
7	6 de Julio	12 de Julio	20	7	2,86		Funcionalidad de consolidar evaluación general del proyecto
8	13 de Julio	19 de Julio	20	7	2,86	Generación de Reportes (Por definir cuantos reportes).	
9 - 12	20 de Julio	14 de Agosto	80	5	4,00	Ajustes Finales (Completar requisitos funcionales, generación del instalador). Por definir (Embeber la capa de persistencia por facilidad de configuración en la instalación)	Sistema experto funcional, instalador de la aplicación.

3.3.2. Fase II: Búsqueda de la fuente del conocimiento

La fuente de conocimiento en el caso de EXSHE la constituyen:

- Jakob Nielsen. Ingeniero de Interfaces, doctor en Diseño de Interfaces de Usuario y Ciencias de la Computación en la Universidad Técnica de Dinamarca, fundador de Nielsen Norman Group, se hará referencia específicamente al artículo “How to Conduct a Heuristic Evaluation” (Cómo llevar a cabo una evaluación heurística), publicado en la dirección Web: <http://www.nngroup.com/articles/how-to-conduct-a-heuristic-evaluation/>, el 01 de enero del 1995.

El aporte de Nielsen es el conocimiento acerca de cómo hacer evaluación heurística.

- Yusef Hassan Montero, consultor sobre Experiencia de Usuario y Visualización de Información, diseñador de Interacción en Scimago Lab, doctor en Documentación de Universidad de Granada, editor de la revista “No Solo Usabilidad”. Coautor de la Guía de evaluación Heurística, publicada en <http://www.nosolousabilidad.com/articulos/heuristica.htm>, el 30 de marzo del 2003.
- Francisco Jesús Martín Fernández, licenciado en Documentación, investigador sobre Usabilidad y Accesibilidad del Centro de Enseñanzas Virtuales de la Universidad de Granada, editor de la revista “NoSoloUsabilidad”. Coautor de la Guía de evaluación Heurística, publicada en <http://www.nosolousabilidad.com/articulos/heuristica.htm>, el 30 de marzo del 2003.

El aporte de Hassan y Martín es el conocimiento acerca de las heurísticas y los factores que se deben medir en una evaluación de usabilidad.

- María del Carmen Suárez Torrente, doctora en Informática, autora de la tesis doctoral “Sirius: Sistema de Evaluación de la Usabilidad Web Orientado al Usuario y basado en la Determinación de Tareas Críticas”, Universidad de Oviedo-Departamento de Informática, publicado en Febrero del 2011 en la dirección: <http://www.di.uniovi.es/~cueva/investigacion/tesis/Sirius.pdf>

El aporte de Suárez representa el conocimiento acerca de las métricas que se van a utilizar en la evaluación heurística.

- Desarrolladores y jefes de proyectos de empresas dedicadas a la industria de software en la Región 3 del Ecuador y la ciudad de Quito. Tomando como referencia la información publicada por la Asociación Ecuatoriana de Software (AESOFT) en el 2014 acerca de las empresas afiliadas que se encuentran ubicadas en este sector; se visitaron 9 empresas de la Región 3 y 9 empresas de la ciudad de Quito. (Ver tabla 3.2)

El conocimiento que se pretendía obtener en relación a este trabajo tenía que ver con la forma de gestionar la usabilidad en el desarrollo de los productos de software, lamentablemente no se concretó el objetivo por cuanto en la mayor parte de empresas visitadas no se considera formalmente este parámetro. (Ver Anexos 2 y 2.1)

La tabla 3.2 detalla la lista de empresas en donde se realizaron las entrevistas.

Tabla 3.2.**Entrevistados – Gerentes y desarrolladores de empresas de la industria del software**

N°	Empresa	Ciudad	Nombres	Cargo	Teléfono
1	BUPARTECH	Quito	Espinoza Juan Carlos	Gerente	02 2247587
2	SODITEC	Quito	Hidalgo José Luis	Gerente General	02 2555269
3	SMARTWORK	Quito	Bastidas Andrés	Gerente General	02 2557352
4	GIGASYSTEM	Quito	Lalama Alex	Gerente	02 2923982
5	SIFIZSOFT S. A.	Quito	Rodríguez Paulina	Desarrollador	02 2448034
6	TATA	Quito	López Hernán	Desarrollador	981398544
7	KRUGER	Quito	Recalde Luis	Desarrollador	
8	CIMAIT	Quito	Revelasco Johnny	Desarrollador	
9	CREATIVE WORKS	Quito	Quiroga Fabián	Desarrollador	02 2463343
10	FENIX	Latacunga	Través Silvia	Programadora	998548169
11	BABEL	Latacunga	Merlo Santiago	Gerente Técnico y Ventas	03 2292431
12	CONNECT AMBATO	Ambato	Eraza Pablo	Director Comercial	992652407
13	ICONO SISTEMAS	Ambato	Vascones David	Jefe de Producción	03 2424708
14	BESIXPLUS	Ambato	Lazcano Klever	Gerente	03 2848748
15	CITY TECH	Riobamba	Aguirre Freddy	Líder del proyecto	03 2944928
16	RIOAXIS	Riobamba	Paca Henry	Representante legal	992619889
17	SYSTEMARKET	Riobamba	Veloz Hítalo	Gerente General	03 2951205
18	KOOPER	Riobamba	Rodríguez Carlos	C.T.O	03 2948137

3.3.3. Fase III: Identificación de conceptos y datos clave

En esta fase se consideran los conceptos generales de la usabilidad y los conceptos específicos de la evaluación heurística, se ha tratado de incluir la mayoría de principios, hechos y relaciones de este campo de estudio.

Para poder identificar los diferentes conceptos se han tomado en cuenta los discursos expuestos por las fuentes de conocimiento descritas anteriormente, la información se recopiló a través de las técnicas de análisis bibliográfico y entrevistas. (Ver Anexo 1)

3.3.4. Fase IV: Selección de las herramientas de software

La selección de las herramientas se realizó a través de un análisis comparativo entre dos lenguajes utilizados para la implementación de sistemas expertos.

A continuación en la tabla 3.3 se describe el mencionado análisis basado en las publicaciones realizadas por Miriam Ayala en el 2011 y Alexander Curiel en el 2013, acerca del desarrollo de sistemas expertos, estas publicaciones permitieron resumir las características más importantes de los lenguajes comparados y proporcionar una calificación entre 1 y 2 que representa el menor o mayor cumplimiento de cada característica respectivamente.

Tabla 3.3.

Tabla comparativa de lenguajes de sistemas expertos

Características	Clips	Jess
Portabilidad	2	2
Flexibilidad	2	2
Integración con Java	2	2
Facilidad de aprendizaje	1	1
Soporte e información	1	2

Como se puede precisar el lenguaje mejor valorado es Jess por lo que es el lenguaje a utilizar en el desarrollo de este sistema, para la selección de las demás herramientas se tomó en consideración específicamente la posibilidad de integración con Jess y su disposición de uso.

A continuación se detalla cada una de las herramientas a utilizar:

- Jess v. 7.1. Base de conocimientos para construir sistemas expertos escritos en Java y basados en reglas.
- JAVA Standard Edition v. 7. Lenguaje de programación orientado a objetos, se integra fácilmente con Jess.
- Entorno de Desarrollo Eclipse Luna. Entorno de desarrollo para Java.
- Window Builder for Eclipse Luna. Herramienta de desarrollo de aplicaciones visuales en Java.
- JavaDB v. JDK 7. Distribución compatible de la base de datos de código abierto Apache Derby de Oracle.
- StarUml v. 5.0.2.1570. Plataforma Open Source UML para el diseño de los modelos del sistema.

3.3.5. Fase V: Adquisición del conocimiento I

En esta fase se distinguen tres etapas:

a. Creación del primer prototipo (prototipo de prueba).

El conocimiento ha sido obtenido a través del análisis de la documentación publicada por las fuentes de conocimiento referidas en la Fase II.

De las entrevistas realizadas al grupo de profesionales de software no se obtuvo ningún conocimiento concreto que apoye a la presente investigación. (Ver anexos 2 y 2.1: Cuestionario de entrevista y resultados relacionados a usabilidad respectivamente)

a.1. Técnicas de adquisición de conocimientos

Las técnicas de adquisición de conocimientos utilizadas son:

- Análisis de tareas. Para conceptualizar y esquematizar el proceso de evaluación heurística definida por los expertos.

- Análisis de protocolos. Para la interpretación de cada una de las reglas heurísticas y sus respectivas sub-heurísticas, y para determinar las métricas de evaluación a utilizar en las mediciones.
- Clasificación de conceptos. Para la sistematización y codificación de los conceptos recogidos de los expertos.

a.2. Conocimiento adquirido

La técnica de análisis de tareas se aplicó al proceso de evaluación heurística que describe Nielsen en 1995, en su artículo “How to Conduct a Heuristic Evaluation”, del cual se obtuvo el siguiente conocimiento:

Conocimiento sobre el proceso para realizar una evaluación heurística:

- La evaluación heurística implica tener un grupo de evaluadores que van a examinar cada interfaz interactiva y a juzgar su conformidad con los principios heurísticos definidos.
- Se recomienda utilizar de cuatro y siete evaluadores.
- La evaluación se lleva a cabo haciendo que cada evaluador inspeccione la interfaz individualmente. Sólo después de que todas las evaluaciones se han completado los evaluadores están autorizados para comunicarse acerca de los resultados de la evaluación. Este procedimiento es importante a fin de asegurar en las evaluaciones independencia e imparcialidad de cada evaluador.
- Los resultados de la evaluación se pueden grabar, ya sea como informes escritos de cada evaluador o conforme avanza en la inspección ir comentando los resultados a un observador asignado para cada proyecto.
- Los informes escritos tienen la ventaja de ser un registro formal de la evaluación, pero requieren de ser leídos y agregados por un gerente de evaluación.

- Los resultados de la evaluación están disponibles muy pronto después de la última sesión de evaluación, el observador sólo tiene que entender y organizar un conjunto de notas personales, no un conjunto de informes escritos por otros.
- Durante la sesión de evaluación, el evaluador navega por la interfaz varias veces e inspecciona los diversos elementos de diálogo y los compara con una lista de principios de usabilidad o heurísticas (ver sección 2.3.4, literal b., pág. 34) que tienen el objetivo de describir las propiedades comunes de las interfaces usables.
- Existe una lista de verificación de heurísticas generales a tener en cuenta para todos los elementos de diálogo, el evaluador obviamente, también se permite considerar que existe algún principio o resultados de usabilidad adicionales que pueden ser relevantes para cualquier elemento de diálogo específico.
- Es posible desarrollar la heurística de categorías concretas que se aplican a una clase específica de productos como un suplemento a los heurísticos generales.
- En principio, los evaluadores deciden por sí mismos cómo quieren proceder con la evaluación de la interfaz. Una recomendación general sería que van a través de la interfaz al menos dos veces, sin embargo. El primer paso sería realizar una observación general que le permita entender el flujo de la interacción y el alcance general del sistema. El segundo paso entonces, permitirá que el evaluador pueda concentrarse en elementos específicos de la interfaz, sabiendo cómo encajan cada uno de ellos en la interfaz en general.
- Dado que los evaluadores no están usando el sistema como tal (para realizar una tarea real), es posible llevar a cabo la evaluación heurística de interfaces de usuario sólo en papel. Esto hace que la evaluación heurística sea adecuada para el uso temprano en el ciclo de vida de la ingeniería de la usabilidad.

- Si el sistema está concebido como una interfaz para la población en general o si los evaluadores son expertos en el dominio, será posible dejar que los evaluadores utilicen el sistema sin más ayuda.
- Si el sistema se encuentra en un dominio dependiente y los evaluadores son novatos con respecto al dominio del sistema, será necesario ayudar a los evaluadores para que puedan utilizar la interfaz. Un enfoque que se ha aplicado con éxito es suministrar a los evaluadores un uso típico (escenario).
- La salida utilizando el método de evaluación heurística es una lista de los problemas de usabilidad de la interfaz con referencia a los principios de usabilidad que fueron vulnerados por el diseño en cada caso en la opinión del evaluador.
- Los evaluadores deben explicar los resultados de su evaluación haciendo referencia a las heurísticas vulneradas en el diseño. Deben tratar de ser lo más específicos posible y deben enumerar cada problema de usabilidad por separado.

La técnica de análisis de protocolos se aplicó a la “Guía de Evaluación Heurística de Sitios Web”, publicada por Hassan y Martín en el 2003, de esta técnica se obtuvo el siguiente conocimiento:

Conocimiento sobre los criterios y sub-criterios a medir en una evaluación heurística:

1. Aspectos generales
 - 1.1. Objetivos del sitio web concretos y bien definidos
 - 1.2. Contenidos y servicios ofrecidos precisos y completos
 - 1.3. Estructura general del sitio web orientada al usuario
 - 1.4. Look & Feel general se corresponde con los objetivos, características, contenidos y servicios del sitio web

- 1.5. Diseño general del sitio web reconocible
- 1.6. Diseño general del sitio web coherente
- 1.7. Se utiliza el idioma del usuario
- 1.8. Se da soporte a otro/s idioma/s
- 1.9. Traducción del sitio completa y correcta
- 1.10. Sitio web actualizado periódicamente
2. Identidad e información
 - 2.1. Identidad o logotipo significativo, identificable y suficientemente visible
 - 2.2. Identidad del sitio en todas las páginas
 - 2.3. Eslogan o tagline adecuado al objetivo del sitio
 - 2.4. Se ofrece información sobre el sitio web, empresa
 - 2.5. Existen mecanismos de contacto
 - 2.6. "Se ofrece información sobre la protección de datos de carácter personal o los
 - 2.7. derechos de autor de los contenidos del sitio web"
 - 2.8. "Se ofrece información sobre el autor, fuentes y fechas de creación y revisión
 - 2.9. en artículos, noticias, informes"
3. Estructura y navegación
 - 3.1. Se ha evitado pantalla de bienvenida
 - 3.2. Estructura de organización y navegación adecuada
 - 3.3. Organización de elementos consistente con las convenciones
 - 3.4. Control del número de elementos y de términos por elemento en los menús de navegación
 - 3.5. Equilibrio entre profundidad y anchura en el caso de estructura jerárquica
 - 3.6. Enlaces fácilmente reconocibles como tales
 - 3.7. La caracterización de los enlaces indica su estado (visitados, activos)
 - 3.8. No hay redundancia de enlaces
 - 3.9. No hay enlaces rotos
 - 3.10. No hay enlaces que lleven a la misma página que se está visualizando
 - 3.11. En las imágenes de enlace se indica el contenido al que se va a acceder

- 3.12. Existe un enlace para volver al inicio, en cada página existen elementos de navegación que orienten al usuario acerca de dónde está y cómo deshacer su navegación
- 3.13. Existe mapa del sitio para acceder directamente a los contenidos sin navegar
4. Rotulado
 - 4.1. Rótulos significativos
 - 4.2. Sistema de rotulado controlado y preciso
 - 4.3. Título de las páginas, correcto y planificado
 - 4.4. URL página principal correcta, clara y fácil de recordar
 - 4.5. URLs de páginas internas claras
 - 4.6. URLs de páginas internas permanentes
5. Layout de la página
 - 5.1. Se aprovechan las zonas de alta jerarquía informativa de la página para contenidos de mayor relevancia
 - 5.2. Se ha evitado la sobrecarga informativa
 - 5.3. Es una interfaz limpia, sin ruido visual
 - 5.4. Existen zonas en blanco entre los objetos informativos de la página para poder descansar la vista
 - 5.5. Uso correcto del espacio visual de la página
 - 5.6. Se utiliza correctamente la jerarquía visual para expresar las relaciones del tipo "parte de" entre los elementos de la página
 - 5.7. Se ha controlado la longitud de página
 - 5.8. La versión impresa de la página es correcta
 - 5.9. El texto de la página se lee sin dificultad
 - 5.10. Se ha evitado el texto parpadeante / deslizante
6. Entendibilidad y facilidad en la interacción
 - 6.1. Se emplea un lenguaje claro y conciso
 - 6.2. Lenguaje amigable, familiar y cercano

- 6.3. Cada párrafo expresa una idea
- 6.4. Uso consistente de los controles de la interfaz
- 6.5. Metáforas visuales reconocibles y comprensibles por cualquier usuario (ej.: iconos)
- 6.6. Si se usan menús desplegables, orden coherente o alfabético
- 6.7. Si el usuario tiene que rellenar un campo, las opciones disponibles se pueden seleccionar en vez de tener que escribirlas
7. Control y retroalimentación
 - 7.1. El usuario tiene todo el control sobre la interfaz
 - 7.2. Se informa al usuario acerca de lo que está pasando
 - 7.3. Se informa al usuario de lo que ha pasado
 - 7.4. Existen sistemas de validación antes de que el usuario envíe información para tratar de evitar errores
 - 7.5. Cuando se produce un error, se informa de forma clara y no alarmista al usuario de lo ocurrido y de cómo solucionar el problema
 - 7.6. Se ha controlado el tiempo de respuesta
 - 7.7. Se ha evitado que las ventanas del sitio anulen o se superpongan a la del navegador
 - 7.8. Se ha evitado la proliferación de ventanas en la pantalla del usuario
 - 7.9. Se ha evitado la descarga por parte del usuario de plugins adicionales
 - 7.10. Si existen tareas de varios pasos, se indica al usuario en cual está y cuantos faltan para completar la tarea
8. Elementos multimedia
 - 8.1. Fotografías bien recortadas
 - 8.2. Fotografías comprensibles
 - 8.3. Fotografías con correcta resolución
 - 8.4. El uso de imágenes o animaciones proporciona algún tipo de valor añadido
 - 8.5. Se ha evitado el uso de animaciones cíclicas
 - 8.6. El uso de sonido proporciona algún tipo de valor añadido

9. Búsqueda

- 9.1. La búsqueda, si es necesaria, se encuentra accesible desde todas las páginas del sitio
- 9.2. Es fácilmente reconocible como tal
- 9.3. Se encuentra fácilmente accesible
- 9.4. La caja de texto es lo suficientemente ancha
- 9.5. Sistema de búsqueda simple y claro
- 9.6. Permite la búsqueda avanzada
- 9.7. Muestra los resultados de la búsqueda de forma comprensible para el usuario
- 9.8. Asiste al usuario en caso de no poder ofrecer resultados para una consultada dada

10. Ayuda

- 10.1. El enlace a la sección de Ayuda está colocado en una zona visible y estándar
- 10.2. Fácil acceso y retorno al/del sistema de ayuda
- 10.3. Se ofrece ayuda contextual en tareas complejas
- 10.4. FAQs (si las hay) correcta la elección como la redacción de las preguntas
- 10.5. FAQs (si las hay) correcta la redacción de las respuestas

La técnica de análisis de protocolos se aplicó también para determinar las métricas a utilizar en la evaluación heurística tomando como referencia la propuesta realizada por (Suárez, 2011) de la cual se obtuvo el siguiente conocimiento:

Conocimiento sobre el modelo de evaluación

- Métricas de evaluación heurística. (ver sección 2.3.4, literal c, pág. 41)
- Relación entre heurísticas, sub-criterios y valoración. (ver sección 2.3.4, literal c1, pág. 42)

La técnica de clasificación de los conceptos analizados en la literatura, permitió obtener una lista de expresiones con sus respectivas relaciones acerca de la usabilidad y de la evaluación heurística. (Ver Anexo 1)

Finalmente se presenta el conocimiento adquirido para determinar los niveles de usabilidad, el cual también ha sido definido a través de la técnica de clasificación de conceptos analizados en la literatura.

Conocimiento para determinar el nivel de usabilidad

La tabla 3.4 detalla las escalas definidas para determinar el nivel de usabilidad.

Tabla 3.4.

Escalas del nivel de usabilidad

Porcentaje de Calificación	Nivel de Usabilidad
85,77 - 100,00	Completamente usable
71,48 - 85,76	Muy usable
57,18 - 71,47	Usable
42,89 - 57,17	Medianamente usable
28,59 - 42,88	Poco usable
14,30 - 28,58	Casi no usable
0,00 - 14,29	No usable

Este conocimiento ha sido determinado a través del siguiente análisis:

- Es necesario inferir el nivel general de usabilidad de cada sitio web evaluado y en vista de que no se plantea por parte de las fuentes del conocimiento ningún modelo aplicable a este estudio, se construye un modelo propio en el cual se definen escalas de valoración para siete niveles. Como se aprecia en la tabla 24, el primer nivel inicia en 0 con una equivalencia cualitativa de “No Usable”, hasta llegar al último nivel que termina en 100 y que equivale a “Completamente Usable”.

- Se plantean 7 niveles para tener una referencia más exacta acerca del nivel de usabilidad de una aplicación ya que elaborar escalas más grandes hubiera significado obtener una valoración incluso menos objetiva.
- De la misma manera para inferir la lista de problemas se estableció que se deben tomar en cuenta aquellos criterios que obtengan en la calificación total un valor promedio menor a 5, para ello se elaboró la tabla 3.5 que permite valorar el nivel de cumplimiento de cada criterio en un sitio o interfaz evaluada. En esta escala 5 representa “Se cumple medianamente”, esto significa que todos los niveles por debajo de esta calificación tienen una equivalencia de insuficientes en el cumplimiento de los criterios.

Tabla 3.5.

Escalas de valoración del cumplimiento de criterios

Escala de valoración	Grado de conformidad de cumplimiento
10	Se cumple en absoluto
8 - 9	Se cumple casi siempre
6 - 7	Se cumple muchas veces
5	Se cumple medianamente
3 - 4	Se cumple pocas veces
1 - 2	Se cumple casi nunca
0	No se cumple en absoluto

- Finalmente para generar las recomendaciones se aplica el mismo concepto definido en la tabla 3.5 con la diferencia de que en este caso se recogen las calificaciones individuales de cada sub-criterio, se seleccionan aquellas inferiores a 5 y se presenta una recomendación específica.

Integración del conocimiento

Todo el conocimiento adquirido se integra para el desarrollo del sistema experto, es decir: se parte del conocimiento del proceso que desarrolla el experto para realizar una evaluación heurística, luego se determina cuáles son los criterios y sub-criterios que se van a medir y finalmente se establecen las métricas que se van a utilizar para obtener los resultados de la evaluación y las recomendaciones respectivas.

a.3. Técnicas de Ingeniería de Software

Sin ser parte de esta metodología se consideró aplicar algunas técnicas de Análisis y Diseño Orientadas a Objetos las mismas apoyaron en la definición de requisitos y en la conceptualización del flujo de procesos más complejo del problema a resolver. Dentro de las técnicas determinadas por el Lenguaje de Modelado Unificado (UML) que se utilizaron, están:

Para el análisis:

- Modelo de casos de uso

Para el diseño

- Modelo de clases
- Diagrama de actividad

A continuación se presentan los resultados generados:

- **Requisitos funcionales del sistema**

Tabla 3.6.

Requisito Funcional # 001

RQF001	Iniciar sesión
Descripción	El sistema debe permitir al usuario iniciar sesión. El inicio significa acceder al sistema a través de un proceso de autenticación, esto es ingresando su respectivo usuario y contraseña.
Usuario	Administrador, evaluador
Tarea	T010
Prioridad	Alta
Precondición	El usuario debe estar previamente registrado en el sistema

Tabla 3.7.

Requisito Funcional # 002

RQF002	Ejecutar evaluación
Descripción	El sistema debe permitir que se evalúe la usabilidad de una aplicación específica (proyecto de evaluación actual).
Usuario	Evaluador
Tarea	T003
Prioridad	Alta
Precondición	El evaluador debe haber iniciado sesión en el sistema

Tabla 3.8.**Requisito Funcional # 003**

RQF003	Editar evaluación
Descripción	El sistema debe permitir que el usuario edite la evaluación del proyecto actual, esto significa que puede tener varias sesiones de trabajo para un mismo proyecto.
Usuario	Evaluador
Tarea	T003
Prioridad	Alta
Precondición	El evaluador puede editar la evaluación del proyecto actual, siempre y cuando no se haya generado el informe de evaluación.

Tabla 3.9.**Requisito Funcional # 004**

RQF004	Genera informe individual de evaluación
Descripción	El sistema debe permitir que se genere el informe de cada evaluación ejecutada.
Usuario	Evaluador
Tarea	T005
Prioridad	Alta
Precondición	La evaluación debe estar completa para que se pueda generar el informe individual.

Tabla 3.10.**Requisito Funcional # 005**

RQF005	Añadir evaluador
Descripción	El sistema debe permitir que se registren a los evaluadores de un proyecto.
Usuario	Administrador
Tarea	T010
Prioridad	Alta
Precondición	El usuario debe haber iniciado sesión con perfil de administrador

Tabla 3.11.**Requisito Funcional # 006**

RQF006	Lee información de un evaluador
Descripción	El sistema debe permitir mostrar la información de un evaluador.
Usuario	Administrador
Tarea	T010
Prioridad	Baja
Precondición	El usuario debe haber iniciado sesión con perfil de administrador

Tabla 3.12.**Requisito Funcional # 007**

RQF007	Actualiza la información de un evaluador
Descripción	El sistema debe permitir actualizar la información de un evaluador.
Usuario	Administrador
Tarea	T010
Prioridad	Baja
Precondición	El usuario debe haber iniciado sesión con perfil de administrador

Tabla 3.13.**Requisito Funcional # 008**

RQF008	Añadir nuevo proyecto
Descripción	El sistema debe permitir añadir un nuevo proyecto de evaluación
Usuario	Administrador
Tarea	T001
Prioridad	Alta
Precondición	El usuario debe haber iniciado sesión con perfil de administrador

Tabla 3.14.**Requisito Funcional # 009**

RQF009	Consolidar evaluación general de un proyecto
Descripción	El sistema tiene que generar un informe consolidado de evaluación, el mismo debe inferir el nivel de usabilidad de la aplicación evaluada y debe incluir además, un reporte de recomendaciones para mejorar su usabilidad.
Usuario	Administrador
Tarea	T004, T005, T006, T007.
Prioridad	Alta
Precondición	Todas las evaluaciones del proyecto deben haberse completado.

Tabla 3.15.**Requisito Funcional # 010**

RQF010	Generar reportes históricos
Descripción	El sistema debe generar reportes históricos acerca de las evaluaciones realizadas en un tiempo determinado.
Usuario	Administrador
Tarea	T008
Prioridad	Baja
Precondición	Todos los proyectos deben estar debidamente registrados, no existe la posibilidad de eliminar un proyecto o una evaluación realizada.

Tabla 3.16.**Requisito Funcional # 011**

RQF011	Leer evaluación de proyecto
Descripción	El sistema debe presentar la evaluación de un proyecto anterior, esto es la memoria de trabajo de una evaluación específica.
Usuario	Administrador
Tarea	T009
Prioridad	Media
Precondición	Todas las evaluaciones del proyecto han sido completadas.

- **Requisitos no funcionales**

Los requisitos no funcionales aplicables a este sistema son los siguientes:

RNF001. La aplicación funcionará bajo la plataforma de Windows 7 como mínimo.

RNF002. El tiempo de respuesta de la aplicación debe ser rápido, dependerá de las condiciones particulares de los equipos de los usuarios del sistema.

RNF003. Las interfaces deben tener un diseño sencillo y amigable para que puedan ser fácilmente manipuladas por los usuarios.

- **Restricciones del sistema**

R001. El sistema no debe generar el informe de evaluador individual, hasta que todos los ítems (sub-criterios) hayan sido evaluados.

R002. El sistema no debe generar el informe consolidado de evaluación, sí no se han completado las cuatro evaluaciones mínimas por proyecto.

R003. Ningún usuario del sistema puede modificar o eliminar una evaluación o un proyecto de evaluación.

R004. Ningún usuario del sistema puede eliminar a un evaluador registrado.

- **Usuarios del sistema**

- **Evaluador:** Es el usuario principal del sistema, gestiona la evaluación de una aplicación web definida en el proyecto actual de evaluación.
- **Administrador:** Es el usuario que administra el sistema, gestiona la información de los evaluadores de un proyecto, la información de los proyectos de evaluación y genera los reportes históricos de evaluaciones.

- **Modelo de Casos de Uso**

El presente diagrama describe los casos de uso del sistema.

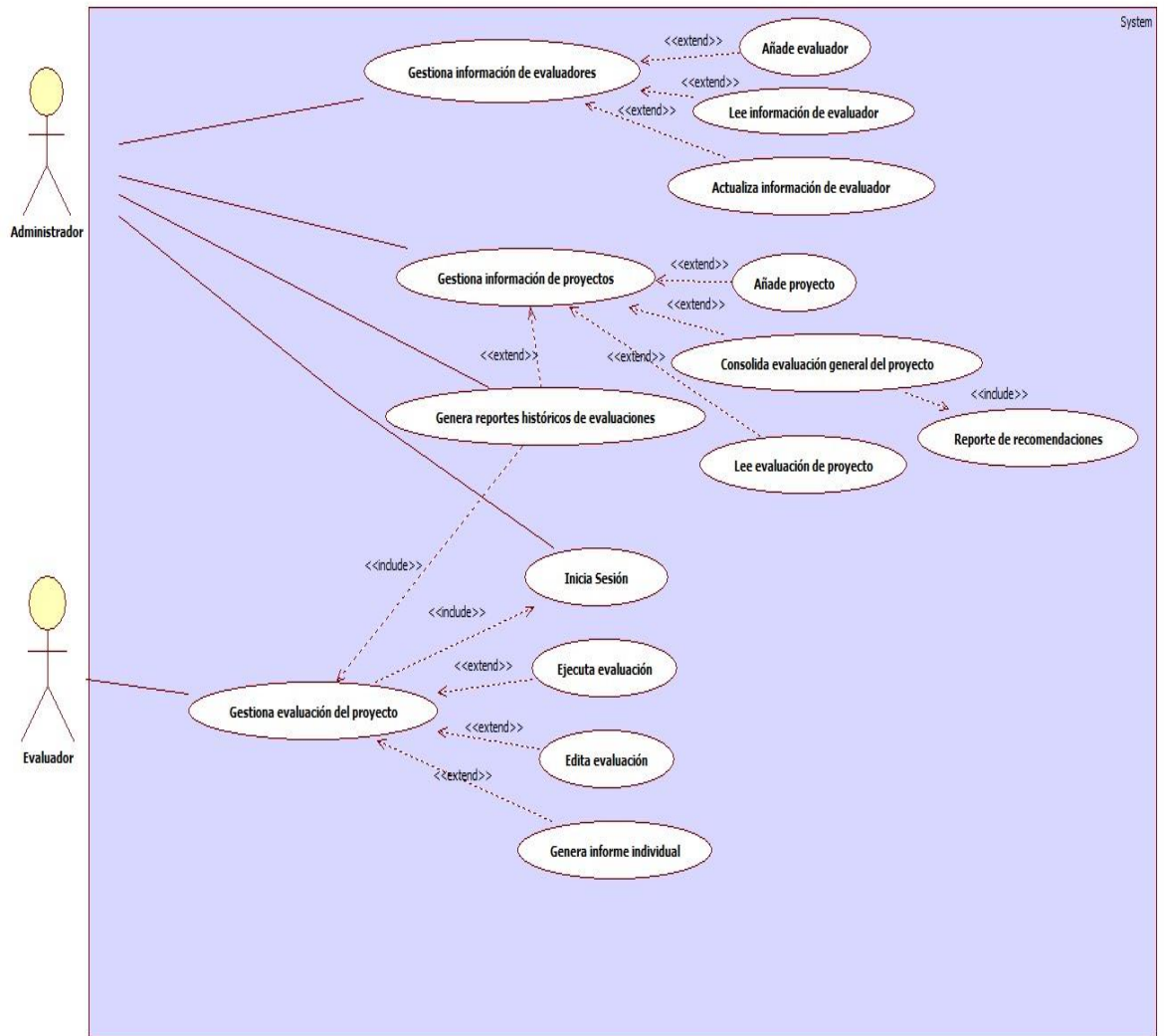


Gráfico 3.1: Modelo de casos de uso

- **Modelo de Clases**

El presente diagrama describe las clases de sistema experto.

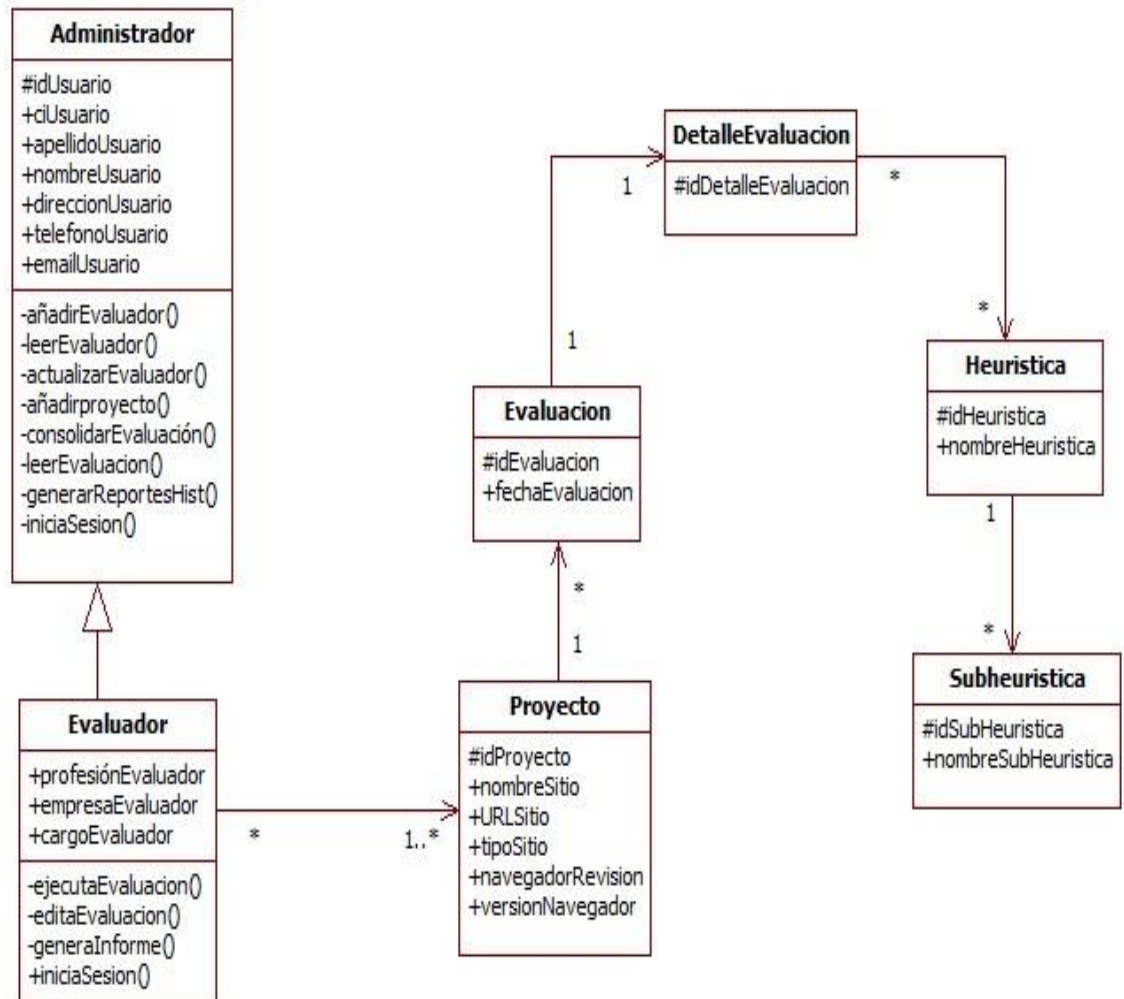


Gráfico 3.2: Modelo de clases

- **Diagrama de Actividad (Caso de uso: Ejecutar Evaluación)**

El presente diagrama describe el proceso más importante que va a ejecutar el sistema experto.

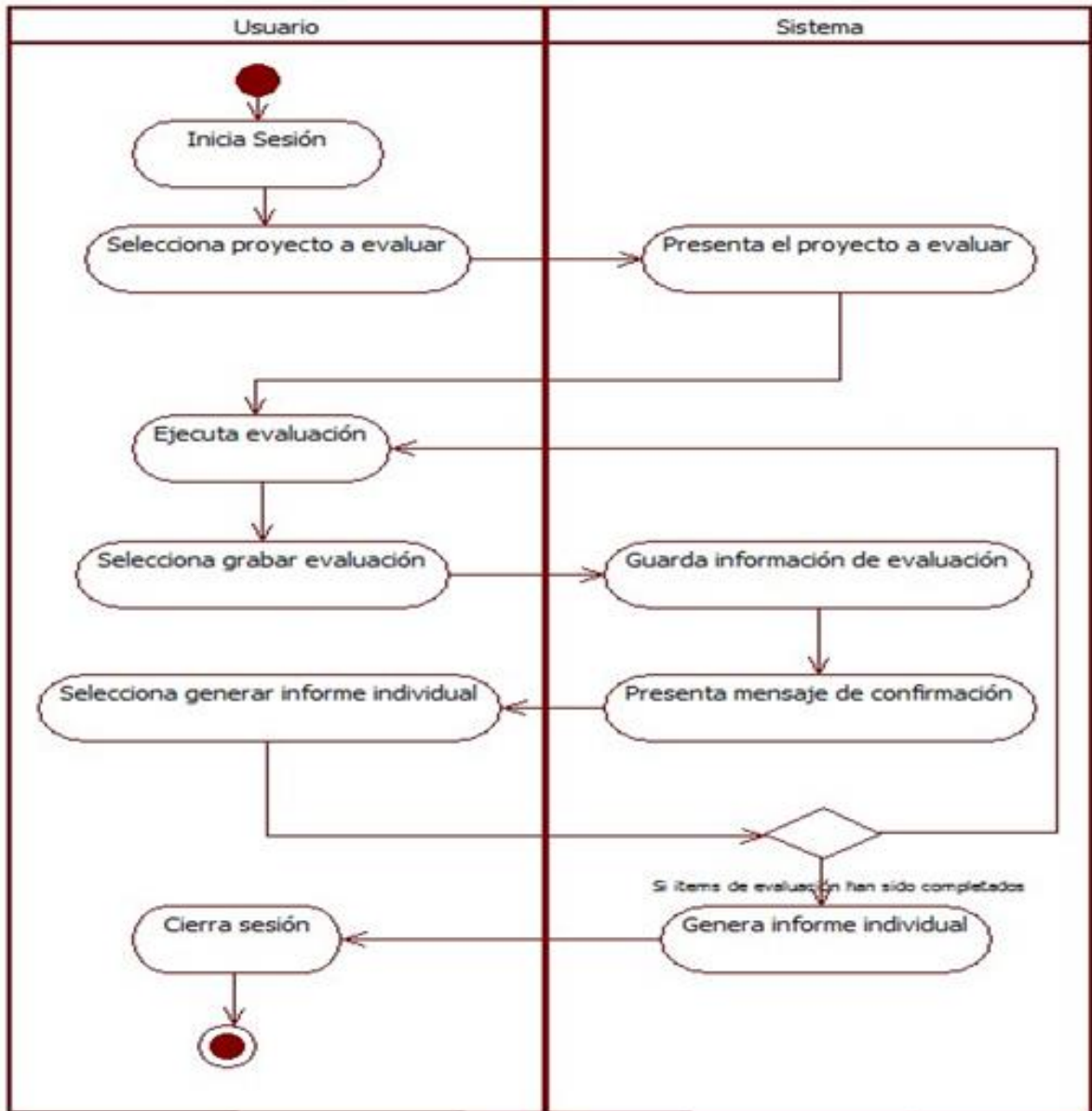


Gráfico 3.3: Diagrama de actividad - Ejecutar evaluación

b. Desarrollo del prototipo de prueba

Para el desarrollo del prototipo de prueba se han considerado los requerimientos de alta prioridad. (Ver Sección 3.3.5, literal a.3, pág. 93)

- RQF001: Iniciar sesión
- RQF002: Ejecutar evaluación
- RQF003: Editar evaluación
- RQF004: Generar informe individual
- RQF005: Añadir evaluador
- RQF008: Añadir proyecto
- RQF009: Consolidar evaluación general de proyecto

c. Evaluación de resultados

La evaluación del prototipo de prueba pretende verificar si las técnicas utilizadas resultan efectivas para la solución del problema a resolver.

El alcance y criterios de esta verificación se definen en base a los requisitos implementados descritos en cada uno de los casos de prueba.

c.1. Casos de prueba

A continuación se presentan los casos de prueba que van a permitir evaluar las funcionalidades implementadas para este prototipo.

Tabla 3.17.**Caso de prueba # 001**

CP001	Inicio de sesión
RQF	RQF001: Iniciar sesión
Condiciones de ejecución	El usuario debe estar registrado en el sistema
Entrada/Pasos de la ejecución	Ingresa usuario y contraseña Selecciona la opción ingresar
Resultado esperado 1:	Despliega mensaje “Bienvenido Usuario”
Resultado esperado 2:	Despliega mensaje “Contraseña o Username incorrecto”
Resultado esperado 3:	Despliega mensaje “Por favor complete los campos”
Evaluación de la prueba	Prueba superada

Tabla 3.18.**Caso de prueba # 002**

CP002	Ejecución de evaluación
RQF	RQF002: Ejecutar evaluación
Condiciones de ejecución	El usuario debe haber ingresado al sistema con el perfil de Evaluador Los proyectos de evaluación deben haber sido cargados previamente
Entrada/Pasos de la ejecución	Selecciona el menú de Evaluaciones Seleccionar proyecto de evaluación Ingresa el nombre y la versión del navegador web en cuál se abrió la aplicación a evaluar Selecciona la opción Ejecutar evaluación Selecciona el proyecto a evaluar Selecciona la opción Calificar
Resultado esperado 1:	Se inicia la sesión de evaluación
Evaluación de la prueba	Prueba superada

Tabla 3.19.**Caso de prueba # 003**

CP003	Calificar evaluación
RQF	RQF002: Ejecutar evaluación, RQF003: Editar evaluación
Condiciones de ejecución	La evaluación debe estar generada
Entrada/Pasos de la ejecución	<p>Selecciona la calificación al sub-heurístico</p> <p>Seleccionar la opción siguiente</p> <p>Repite los pasos anteriores hasta terminar la evaluación</p> <p>Selecciona Guardar sesión</p> <p>Selecciona Finalizar</p>
Resultado esperado 1:	Se despliega el mensaje “Evaluación finalizada exitosamente”
Resultado esperado 2:	Se despliega el mensaje “No se puede finalizar, aún tienen criterios pendientes por calificar”
Evaluación de la prueba	Prueba superada

Tabla 3.20.**Caso de prueba # 004**

CP004	Informe de evaluador
RQF	RQF004: Generar informe individual
Condiciones de ejecución	El usuario debe haber ingresado con el perfil de Evaluador La evaluación debe estar finalizada
Entrada/Pasos de la ejecución	Selecciona la opción Informes del menú Evaluaciones Selecciona la evaluación correspondiente Selecciona la opción Informe
Resultado esperado 1:	Se genera el informe de evaluación individual
Evaluación de la prueba	Prueba superada

Tabla 3.21.**Caso de prueba # 005**

CP005	Añadir evaluador
RQF	RQF005: Añadir evaluador
Condiciones de ejecución	El usuario debe haber ingresado con el perfil de Administrador
Entrada/Pasos de la ejecución	Selecciona la opción Mantenimiento del menú Evaluadores

CONTINÚA →

ejecución	Registrar datos del evaluador Selecciona la opción guardar
Resultado esperado 1:	Despliega mensaje “Evaluador guardado exitosamente”
Resultado esperado 2:	Despliega mensaje “Por favor complete los campos”
Evaluación de la prueba	Prueba superada

Tabla 3.22.

Caso de prueba # 006

CP006	Nuevo proyecto
RQF	RQF008: Añadir proyecto
Condiciones de ejecución	El usuario debe haber ingresado con el perfil de Administrador
Entrada/Pasos de la ejecución	Ingresar el nombre del sitio a evaluar Ingresar el URL del sitio a evaluar Selecciona Tipo del sitio Selecciona los evaluadores del proyecto Selecciona la opción Guardar
Resultado	Aparecen los datos generales y evaluadores del nuevo proyecto

CONTINÚA →

esperado 1:	
Resultado esperado 2:	No permite guardar el proyecto con evaluadores duplicados
Resultado esperado 3:	No permite guardar el proyecto con menos de 4 evaluadores
Evaluación de la prueba	Prueba superada

Tabla 3.23.

Caso de prueba # 007

CP007	Evaluación consolidada
RQF	RQF009: Consolidar evaluación general de proyecto
Condiciones de ejecución	El usuario debe haber ingresado con el perfil de Administrador El proyecto debe tener todas las evaluaciones finalizadas
Entrada/Pasos de la ejecución	Selecciona la opción Proyectos Selecciona el proyecto Selecciona la opción Informe
Resultado esperado 1:	Se genera el informe de evaluación consolidado
Resultado esperado 2:	No permite generar el informe porque no se han completado las evaluaciones

CONTINÚA →

Evaluación de la prueba	Prueba superada
--------------------------------	-----------------

c.2. Informe de la evaluación

Los casos de prueba han sido ejecutados varias veces, una vez que todos han sido superados con éxito se puede concluir que las funcionalidades implementadas en este primer prototipo cumplen con los requerimientos de usuario clasificados como de alta prioridad, por consiguiente se puede continuar con el resto de funcionalidades para completar el desarrollo del sistema experto.

3.3.6. Fase VI: Representación del conocimiento y formalización del razonamiento

En vista de que los resultados del prototipo de prueba han sido satisfactorios, pues han dado soluciones adecuadas para la aplicación de la evaluación heurística en mediciones de usabilidad, es viable continuar con el desarrollo de EXSHE. A continuación en esta fase se procede a formalizar el conocimiento implementado, el mismo que se ha generado a través de reglas semánticas desarrolladas en el lenguaje de programación Jess.

a. Implementación de la base del conocimiento

A continuación se presentan ejemplos de las reglas implementadas:

- Reglas para inferir el nivel de usabilidad

(deftemplate Informe

(slot nombre)

```

        (slot porcentaje)
        (slot condicion)
    )
    (defrule completamenteUsable
        ?f1<-(Informe (nombre ?nombre) (porcentaje ?porcentaje) (condicion
nil))
        =>( if (>= ?porcentaje 85.76 ) then
            (retract ?f1)
            (assert (Informe(nombre ?nombre)(porcentaje
?porcentaje)(condicion " Completamente Usable")))
        )
    )
    (defrule pocoUsable
        ?f1<-(Informe (nombre ?nombre) (porcentaje ?porcentaje) (condicion
nil))
        =>(if ( and (>= ?porcentaje 28.59) (< ?porcentaje 42.89)) then
            (retract ?f1)
            (assert (Informe(nombre ?nombre)(porcentaje
?porcentaje)(condicion "Poco Usable")))
        ))

```

- Reglas para generar las recomendaciones

En esta sección se presenta como ejemplo la implementación para generar las recomendaciones correspondientes al criterio aspectos generales.

```

(deftemplate Criterio
    (slot nombre)
    (slot calificacion)

```

```

        (slot condicion)
    )
    (defrule ag1
      ?f1<-(Criterio (nombre ?nombre) (calificacion ?calificacion)
      (condicion nil))
      =>
      (if (and (< ?calificacion 5)(= ?nombre AG1)) then
        (retract ?f1)
        (assert (Criterio(nombre ?nombre)(calificacion
?calificacion)(condicion "Los objetivos del sitio deben ser concretos y bien
definidos"))))
    )
  )
  (defrule ag2
    ?f1<-(Criterio (nombre ?nombre) (calificacion ?calificacion)
    (condicion nil))
    =>
    (if (and (< ?calificacion 5)(= ?nombre AG2)) then
      (retract ?f1)
      (assert (Criterio(nombre ?nombre)(calificacion
?calificacion)(condicion "Los contenidos y servicios ofrecidos deben ser precisos y
completos"))))
  )

```

- **Reglas para inferir la lista de problemas**

```

(deftemplate Heuristico
  (slot nombre)

```

```

        (slot promedio)
        (slot condicion)
    )
(defrule ag
    ?f1<-(Heuristico (nombre ?nombre) (promedio ?promedio)
(condicion nil))
    =>
    (if (and (< ?promedio 5)(= ?nombre AG)) then
        (retract ?f1)
        (assert (Heuristico(nombre ?nombre)(promedio
?promedio)(condicion "Aspectos generales")))
    )
)
(defrule ii
    ?f1<-(Heuristico (nombre ?nombre) (promedio ?promedio)
(condicion nil))
    =>
    (if (and (< ?promedio 5)(= ?nombre II)) then
        (retract ?f1)
        (assert (Heuristico(nombre ?nombre)(promedio
?promedio)(condicion "Identidad e Información")))
    )
)
(defrule en
    ?f1<-(Heuristico (nombre ?nombre) (promedio ?promedio)
(condicion nil))
    =>

```



```

      (if (and (< ?promedio 5)(= ?nombre EN)) then
        (retract ?f1)
        (assert (Heuristico(nombre ?nombre)(promedio
?promedio)(condicion "Estructura y navegaci3n")))
      )
    )
  )

```

- Funciones para inferir las conclusiones

```

(
deffunction final (?listaProblemas ?total)
(bind ?conclusion)
(
  if(and(= ?listaProblemas "Se cumple pocas veces")(> ?total 0))
  then ((?conclusion "Los resultados obtenidos permiten inferir que los criterios
de la lista de problemas se cumplen pocas veces, por lo tanto el grado de dificultad es
MODERADO y se requiere mejorar el dise1o de las interfaces gr1ficas en relaci3n a
estos criterios espec1ficos")
    else
    (
      if(and(= ?listaProblemas "Se cumple casi nunca")(> ?total 0))
      then ((?conclusion "Los resultados obtenidos permiten inferir que los
criterios de la lista de problemas casi nunca se cumplen, por lo tanto el grado de
dificultad es GRAVE y se requiere con urgencia tomar medidas orientadas a
reestructurar el dise1o de las interfaces gr1ficas en relaci3n a estos criterios
espec1ficos")
        else
        (
          if(and(= ?listaProblemas "No se cumple en absoluto")(> ?total 0))

```

then ((?conclusion "Los resultados obtenidos permiten inferir que los criterios de la lista de problemas no se cumplen en absoluto, por lo tanto el grado de dificultad es CRITICO y se requiere redefinir el sitio incorporando en el diseño de las interfaces gráficas, todos los criterios de usabilidad"))

)
)
)
)

3.3.7. Fase VII: Desarrollo del prototipo y pruebas

a. Evolución del prototipo

En la presente sección se completa el prototipo del sistema, el objetivo es obtener el SSEE funcional en su totalidad, es decir con todos los requerimientos del usuario implementados.

a.1.Requisitos funcionales a implementar

Los requerimientos a implementar son los que tienen una clasificación de media y baja prioridad. (Ver Sección 3.3.5, literal a.3, pág. 93). A continuación se especifican los mismos:

- RQF006: Leer información de un evaluador
- RQF007: Actualizar información de un evaluador
- RQF010: Generar reportes históricos
- RQF011: Leer evaluación de proyecto

b. Evaluación de resultados

En esta sección se detallan los casos de prueba que validan los requisitos implementados.

b.1. Casos de prueba

Tabla 3.24.

Caso de prueba # 008

CP008	Lectura de la información de un evaluador
RQF	RQF006: Leer información de un evaluador
Condiciones de ejecución	El usuario debe estar registrado en el sistema con el perfil de Administrador
Entrada/Pasos de la ejecución	Selecciona la opción Evaluadores del menú principal Selecciona la opción Mantenimiento
Resultado esperado 1:	Ventana con la información de todos los evaluadores registrados
Evaluación de la prueba	Prueba superada

Tabla 3.25.

Caso de prueba # 009

CP009	Actualización de la información de un evaluador
RQF	RQF007: Actualizar información de un evaluador
Condiciones de ejecución	El usuario debe estar registrado en el sistema con el perfil de Administrador
Entrada/Pasos de la ejecución	<p>Selecciona la opción Evaluadores del menú principal</p> <p>Selecciona la opción mantenimiento</p> <p>Selecciona la información del evaluador que va a actualizar</p> <p>Modifica los campos pertinentes</p> <p>Selecciona la opción Actualizar</p>
Resultado esperado 1:	Despliega el mensaje “Evaluador actualizado exitosamente”
Evaluación de la prueba	Prueba superada

Tabla 3.26.

Caso de prueba # 010

CP010	Generación de reportes históricos
RQF	RQF010: Generar reportes históricos
Condiciones de ejecución	El usuario debe estar registrado en el sistema con el perfil de Administrador

CONTINÚA →

Entrada/Pasos de la ejecución	<p>Selecciona la opción Evaluaciones del menú principal</p> <p>Selecciona la opción Finalizadas</p>
Resultado esperado 1:	Reporte con los datos de todas las evaluaciones finalizadas
Evaluación de la prueba	Prueba superada

Tabla 3.27.

Caso de prueba # 011

CP011	Lectura de evaluación de proyecto
RQF	RQF011: Leer evaluación de proyecto
Condiciones de ejecución	El usuario debe estar registrado en el sistema con el perfil de Administrador
Entrada/Pasos de la ejecución	<p>Selecciona la opción Evaluaciones del menú principal</p> <p>Selecciona la opción Resultados</p> <p>Selecciona el proyecto a leer</p>
Resultado esperado 1:	Reporte de los resultados de la evaluación seleccionada
Evaluación de la prueba	Prueba superada

b.2. Informe de la evaluación

Los casos de prueba han sido superados con éxito, se puede concluir que las funcionalidades implementadas en esta segunda fase del prototipo cumplen con los requerimientos de usuario clasificados como de media y baja prioridad, por consiguiente el sistema experto cumple con todos los requisitos funcionales especificados.

3.3.8. Fase VIII: Adquisición del conocimiento II

Si bien la metodología incluye esta segunda fase de adquisición del conocimiento, en el caso específico de este proyecto no es necesario implementarla por cuanto el sistema ha sido completado.

3.3.9. Fase IX: Mantenimiento y actualización

En relación al mantenimiento y actualización del sistema es posible de realizarlo ya que el mismo ha sido desarrollado en herramientas flexibles que permiten la escalabilidad de cualquier software, por lo tanto es factible incrementar su base de conocimiento y otras funcionalidades dependiendo de las necesidades de la institución en donde se va a implantar el SSEE.

Además, como soporte del sistema se presentan las respectivas asistencias a los usuarios a través de las ayudas incorporadas al sistema experto (Ver opción Ayuda del menú principal) y el manual técnico de instalación. (Ver Anexo 3: Manual Técnico)

CAPÍTULO IV

IMPLANTACIÓN DEL SISTEMA EXPERTO EN EL DEPARTAMENTO DE SERVICIOS INFORMÁTICOS DE LA UTC.

4.1. Implantación del SSEE.

El sistema experto ha sido implantado en el Departamento de Servicios Informáticos de la Universidad Técnica de Cotopaxi. A continuación se detalla las características del equipo donde fue instalado:

- Características del equipo de instalación

- Edición Windows 7 Ultimate
- Procesador Intel ® Core™ i7 – 2600 CPU @ 3.40 GHz
- Memoria RAM 4,00 GB
- Sistema operativo de 64 bits

4.2. Pruebas de aceptación

Las pruebas de aceptación del sistema las realizó el profesional encargado de Administrar los Entornos Virtuales de Aprendizaje, esta responsabilidad tal como se manifestó en secciones anteriores está a cargo del Ing. Gonzalo Borja Almeida (Ver Anexo 4), quien será también el encargado de administrar EXSHE en la institución.

- Pruebas de Inicio de Sesión y Gestión de Proyectos

1. El usuario ingresa con el perfil de administrador.



Gráfico 4.1: Ventana de inicio de EXSHE

2. Se muestra la ventana de administración de proyectos

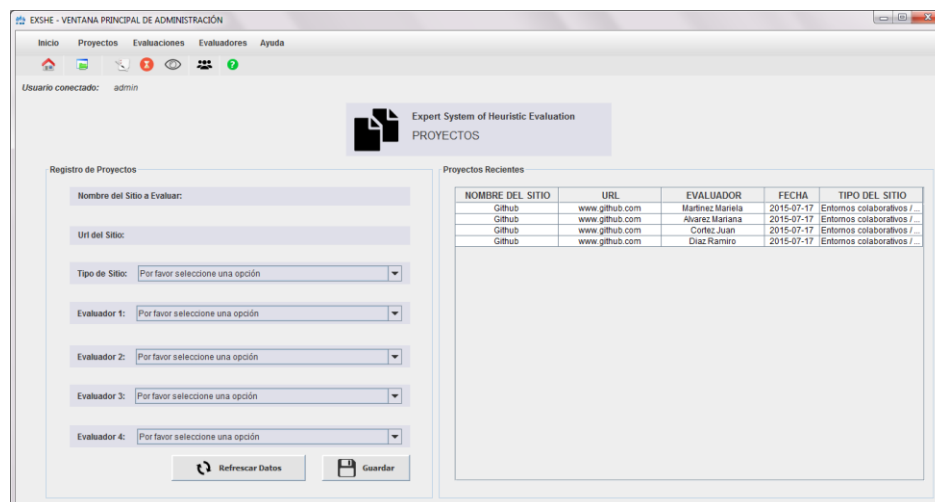


Gráfico 4.2: Ventana de Administración de proyectos

3. En esta ventana el administrador pueda añadir nuevos proyectos y leer la información de todos proyectos creados.

NOMBRE DEL SITIO	URL	EVALUADOR	FECHA	TIPO DEL SITIO
Universidad Técnica de...	www.ut.edu.ec	Pérez Jara Carlos Andrés	2015-07-27	Educativo / Formativo
Universidad Técnica de...	www.ut.edu.ec	Martinez Mariela	2015-07-27	Educativo / Formativo
Universidad Técnica de...	www.ut.edu.ec	Alvarez Mariana	2015-07-27	Educativo / Formativo
Universidad Técnica de...	www.ut.edu.ec	Cortez Juan	2015-07-27	Educativo / Formativo
Universidad Técnica de...	www.ut.edu.ec	Tapia Cerda Verónica de	2015-07-27	Educativo / Formativo
Universidad Técnica de...	www.ut.edu.ec	Pérez Jara Carlos Andrés	2015-07-27	Educativo / Formativo
Universidad Técnica de...	www.ut.edu.ec	Alvarez Mariana	2015-07-27	Educativo / Formativo
Universidad Técnica de...	www.ut.edu.ec	Diaz Ramiro	2015-07-27	Educativo / Formativo
GitHub	www.github.com	Martinez Mariela	2015-07-17	Entornos colaborativos /
GitHub	www.github.com	Alvarez Mariana	2015-07-17	Entornos colaborativos /
GitHub	www.github.com	Cortez Juan	2015-07-17	Entornos colaborativos /
GitHub	www.github.com	Diaz Ramiro	2015-07-17	Entornos colaborativos /

Gráfico 4.3: Ventana de gestión de proyectos

4. Para consolidar la evaluación general de un proyecto, selecciona el menú Evaluaciones, luego la opción resultados y finalmente escoge el proyecto. A continuación se generará el informe consolidado de evaluación.

CÓDIGO	CRITERIO DE EVALUACIÓN	PROMEDIO
BUF	Muestra los resultados de la búsqueda de forma comprensible por el usuario	6,00
BUB	Asiste al usuario en caso de no poder obtener resultados para sus consultas	7,00
AY1	El acceso a la sección de Ayuda está ubicado en una zona visible y accesible	7,00
AY2	Fácil acceso y retorno al área de ayuda	6,00
AY3	Se ofrece ayuda contextual en tareas complejas	6,25
AY4	FACD (o las hay) conecta la elección con la redacción de los proyectos	7,00
AY5	FACD (o las hay) conecta la redacción de los proyectos	7,25

PORCENTAJE DE USABILIDAD: 75,13% SICD: Muy Usable

Abreviatura	Significado
ACD	Accesos Contextuales
I	Identidad e Información
EN	Estructura y Navegación
RO	Recursos
LA	Layout de la Página
EF	Entendibilidad y Facilidad en la Interacción
OM	Control de Interactividad
EM	Elementos Multimediales
BU	Búsqueda
AY	Ayuda

Gráfico 4.4: Reporte consolidado de evaluación

5. En la misma ventana el administrador puede generar el informe de recomendaciones.

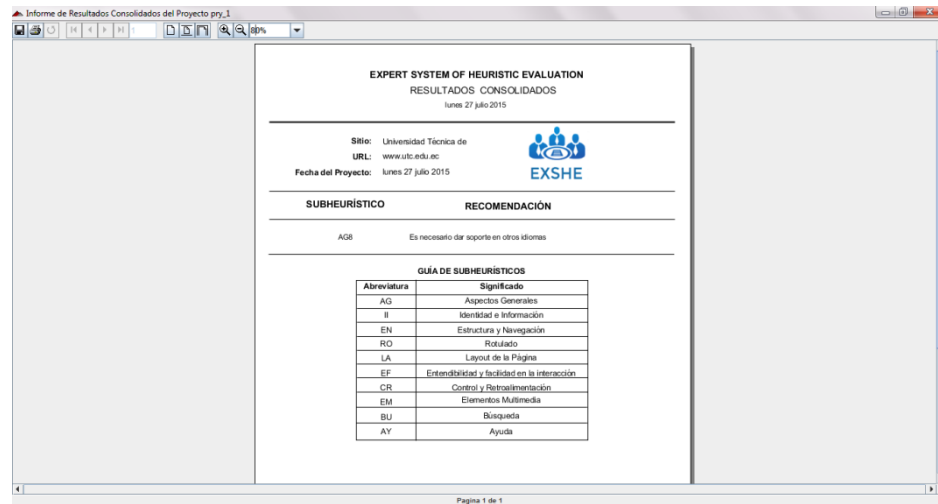


Gráfico 4.5: Reporte de recomendaciones

6. Para generar el reporte histórico de evaluaciones, selecciona el menú Evaluaciones, luego la opción Finalizadas.

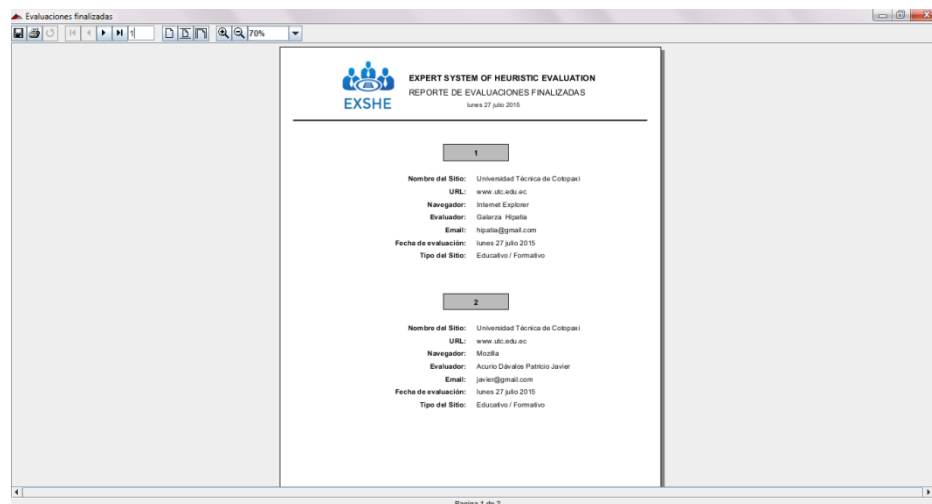


Gráfico 4.6: Reporte histórico de evaluaciones

- **Prueba de Gestión de Información de Evaluadores**

1. El administrador del sistema ingresa al menú Evaluadores y selecciona la opción Mantenimiento.

EXSHE - MANTENIMIENTO DE EVALUADORES

← Atrás

Expert System of Heuristic Evaluation
Mantenimiento de Evaluadores

Registro de Evaluadores

Apellido: Tapia Cerda Nombre: Verónica del consuelo Email: verot500@yahoo.es

Dirección: Latacunga, Campo Alegre Teléfono: 0984518357 Indique la profesión: Ingeniera en Sistemas

Empresa: UTC Cargo que ocupa: Docente titular

Escriba una Contraseña: ***** Confirme la Contraseña: *****

Limpiar Actualizar Guardar

Evaluadores Registrados

APELLIDO	NOMBRE	EMAIL	DIRECCIÓN	TELÉFONO	PROFESIÓN	EMPRESA	CARGO	PERFIL	PASSWORD
Alvarez	Mariana	alvarez@gmail.com						EVALUADOR	123456
Cortez	Juan	cortez@gmail.com						EVALUADOR	123456
Diaz	Ramiro	diaz@gmail.com			Ingeniero en Sistemas			EVALUADOR	123456
Martinez	Mariana	martinez@gmail.com			Ingeniera en sistemas			EVALUADOR	123456
Rivero Jara	Carlos Andrés	carlos.rivero@gmail.com	Quito, Cuero Caicedo y Ma.	02282371	Ingeniero en Sistemas	InterExpress	Jefe de Desarrollo	EVALUADOR	123456

Gráfico 4.7: Ventana de mantenimiento de evaluadores

2. En esta ventana el usuario puede añadir nuevos evaluadores o actualizar la información de un evaluador.

- **Prueba de Gestión de Evaluación del Proyecto**

1. El usuario ingresa al sistema con el perfil de Evaluador, selecciona el proyecto a evaluar, completa la información de la evaluación a realizar, selecciona la opción de Ejecutar y finalmente la opción Calificar. El resultado será la ventana del test de evaluación.

Calificación de criterios

Nombre del Sitio: CNT URI del Sitio: www.cnt.gob.ec

Navegador para la evaluación: Chrome Versión: 44.0.2403.107 m

CRITERIOS A EVALUAR 1 de 10

AG1: Objetivos del sitio web concretos y bien definidos
Seleccione una opción

AG2: Contenidos y servicios ofrecidos precisos y completos
Seleccione una opción

AG3: Estructura general del sitio web orientada al usuario
Seleccione una opción

AG4: Look & Feel general se corresponde con los objetivos, características, contenidos y servicios del sitio web
Seleccione una opción

AG5: Diseño general del sitio web reconocible
Seleccione una opción

AG6: Diseño general del sitio web coherente
Seleccione una opción

AG7: Se utiliza el idioma del usuario
Seleccione una opción

AG8: Se da soporte a otros idiomas
Seleccione una opción

AG9: Traducción del sitio completa y correcta
Seleccione una opción

Anterior Siguiente Guardar Sesión Finalizar

Gráfico 4.8: Ventana de calificación de proyecto

- El evaluador puede guardar cada una de las sesiones que realiza durante la evaluación, solo en el caso de haber completado todos los sub-criterios puede seleccionar la opción Finalizar.
- Para generar el informe de evaluación final, el evaluador debe retornar a la ventana principal de evaluación y seleccionar la opción Informe de la evaluación correspondiente.

Reporte de la Evaluación en_1

EXPERT SYSTEM OF HEURISTIC EVALUATION
REPORTE DE EVALUACIÓN INDIVIDUAL

Nombre: Universidad Tecnológica
Sitio: www.cnt.gob.ec
Navegador: Google Chrome
Versión: 44.0.2403.107 m
Fecha de Evaluación: lunes 27 julio 2015

RESULTADOS OBTENIDOS

CODIGO	CRITERIO DE EVALUACIÓN	PUNTO
AG1	Objetivos del sitio web concretos y bien definidos	7.0
AG2	Contenidos y servicios ofrecidos precisos y completos	9.0
AG3	Estructura general del sitio web orientada al usuario	7.0
AG4	Look & Feel general se corresponde con los objetivos, características, contenidos y servicios del sitio web	9.0
AG5	Diseño general del sitio web reconocible	7.0
AG6	Diseño general del sitio web coherente	9.0
AG7	Se utiliza el idioma del usuario	10.0
AG8	Se da soporte a otros idiomas	9.0
AG9	Traducción del sitio completa y correcta	9.0
AG10	Sitio web actualizado periódicamente	9.0

EVALUADOR: Tania Cristina Gonzalez del lunes 27 julio 2015 Pagina 1 de 4

Página 1 de 4

Gráfico 4.9: Informe de evaluación individual

- Lista de verificación de las pruebas de aceptación

La tabla 4.1 contiene la lista de verificación de las pruebas realizadas por el usuario, las mismas que se ejecutaron en base a los casos de prueba descritos en las secciones: 3.3.5, literal c1, pág. 103 y 3.3.7, literal b1, pág. 116.

Tabla 4.1.

Lista de verificación de las pruebas de aceptación

N°	Requisito Funcional	Código del Caso de Prueba	Resultado
Usuario Evaluador			
1	RQF001: Iniciar sesión	CP001	Aprobada
2	RQF002: Ejecutar evaluación	CP002, CP003	Aprobada
3	RQF003: Editar evaluación	CP003	Aprobada
4	RQF004: Generar informe individual	CP004	Aprobada
Usuario Administrador			
5	RQF001: Iniciar sesión	CP001	Aprobada
6	RQF005: Añadir evaluador	CP005	Aprobada
7	RQF006: Leer evaluador	CP008	Aprobada
8	RQF007: Actualizar evaluador	CP009	Aprobada
9	RQF008: Añadir proyecto	CP006	Aprobada
10	RQF009: Consolidar evaluación	CP007	Aprobada
11	RQF010: Generar reportes históricos	CP010	Aprobada
11	RQF011: Leer evaluación	CP011	Aprobada

4.3. Formalización de la aceptación

Se formaliza la aceptación del sistema por medio de una carta con la firma y cédula del Administrador de los Entornos Virtuales de Aprendizaje de la UTC, Ing. Gonzalo Borja Almeida. (Ver Anexo 5)

CAPÍTULO V

VALIDACIÓN DEL SISTEMA EXPERTO EN LAS AULAS VIRTUALES DE LA UNIVERSIDAD TÉCNICA DE COTOPAXI

5.1. Organización de la recogida y procesamiento de datos

En esta fase se valida el sistema experto EXSHE a través de la evaluación de los entornos virtuales de aprendizaje de la Universidad Técnica de Cotopaxi, tomando en cuenta que esta institución tiene una oferta académica de 18 carreras, se ha seleccionado como muestra las aulas virtuales de la Carrera de Ingeniería en Informática y Sistemas Computacionales del período académico Abril - Agosto 2015. (Ver gráfico 5.1)

The screenshot displays the 'Aulas Virtuales' management interface. The main content area is titled 'Gestión de cursos y categorías' and shows a table of categories and courses for the 'PERIODO ABRIL - AGOSTO 2015'.

Categorías		PERIODO ABRIL - AGOSTO 2015
■ UA-CIYA	86	Crear nuevo curso Por página: 20
■ INGENIERÍA EN INFORMÁTICA	57	Prueba Formato Gift jc01
■ PERIODO ABRIL - AGOSTO 2015	7	Restauración del curso iniciada copia 21
* PRIMER CICLO	1 CICLO 0	Restauración del curso iniciada copia 20
* SEGUNDO CICLO	2 CICLO 0	Restauración del curso iniciada copia 19
* TERCER CICLO	3 CICLO 0	EMPRENDIMIENTO II SISTEMAS
* CUARTO CICLO	4 CICLO 0	Lenguajes de Programación 01
* QUINTO CICLO	5 CICLO 0	EDATOS 15-15
* SEXTO CICLO	6 CICLO 0	Mostrando todos 7 cursos
* SEPTIMO CICLO	7 CICLO 0	
* OCTAVO CICLO	8 CICLO 0	
* NOVENO CICLO	9 CICLO 0	
■ PERIODO OCTUBRE 2014 - FEBRERO 2015	INFO1415	
■ DISEÑO GRÁFICO	38	
■ INGENIERÍA INDUSTRIAL	15	

Gráfico 5.1: Aulas virtuales de la carrera de Ingeniería en Informática y Sistemas Computacionales del período académico Abril – Agosto 2015

- Aulas virtuales para la evaluación

La tabla 5.1 describe la información detallada de las aulas virtuales que se van a evaluar con el sistema EXSHE.

Tabla 5.1.

Entornos virtuales de aprendizaje que se van a evaluar

Carrera de Ingeniería en Informática y Sistemas Computacionales		
Asignatura	URL	Ciclo
Administración de Sistemas Operativos	http://avirtuales.utc.edu.ec/course/view.php?id=1681	Octavo
Ingeniería de software I	http://avirtuales.utc.edu.ec/course/view.php?id=1470	Séptimo
Bases de Datos Aplicadas	http://avirtuales.utc.edu.ec/course/view.php?id=185	Séptimo
Aplicaciones Web	http://avirtuales.utc.edu.ec/course/view.php?id=1683	Séptimo
Proyecto Integrador I	http://avirtuales.utc.edu.ec/course/view.php?id=1109	Quinto
Redes	http://avirtuales.utc.edu.ec/course/view.php?id=963	Sexto
Mantenimiento de computadores	http://avirtuales.utc.edu.ec/course/view.php?id=187	Quinto

- Evaluadores

Los evaluadores de los entornos virtuales son profesionales del campo de la ingeniería de software y del diseño gráfico computarizado, se ha tratado de consolidar la colaboración de personas con este perfil ya que se busca tanto la validación del sistema EXSHE como la medición especializada y objetiva de estas aplicaciones, es necesario indicar que la medición se realizará específicamente sobre las interfaces de las aulas verificando su diseño y la facilidad de interacción del usuario sobre ellas.

A continuación la tabla 5.2 presenta la información de los profesionales que van a evaluar los entornos virtuales de aprendizaje de la UTC.

Tabla 5.2.

Evaluadores de los entornos virtuales de aprendizaje

Nombre	Profesión	Cargo	Empresa/Institución
Edwin Oswaldo Chasiquiza Molina	Ingeniero en Sistemas e Informática Magister en Ingeniería en Software	Analista-Desarrollador	GRUPO BABEL –EC
Diego Ismael Velasco Sánchez	Ingeniero en Sistemas e Informática Magister en Ingeniería en Software	Analista-Desarrollador	GRUPO BABEL –EC
Marcela Ximena Parra Pérez	Diseñadora Magister en Diseño Multimedia	Docente de la Carrera de Diseño Gráfico	Universidad Técnica de Cotopaxi
Carlos Efraín Chasiluisa Taco	Ingeniero en Diseño Gráfico Computarizado	Docente de la Carrera de Diseño Gráfico	Universidad Técnica de Cotopaxi

- Proceso de medición

El proceso de medición se desarrolló a través de las siguientes actividades:

1. Inducción acerca del sistema EXSHE. Esta actividad consistió en explicar a los evaluadores acerca de los objetivos y el manejo del sistema, el tiempo estimado de duración de esta etapa fue de 60 minutos.
2. Evaluación de entrenamiento. Los evaluadores realizaron una evaluación de entrenamiento que les permitió familiarizarse con el uso del sistema EXSHE antes de proceder con la evaluación de los entornos virtuales de aprendizaje.
3. Evaluación de los entornos virtuales de aprendizaje a través del siguiente proceso:
 - Cada uno de los entornos virtuales que se van a evaluar están organizados en un proyecto de evaluación en el sistema EXSHE
 - Los evaluadores proceden a ejecutar la evaluación de cada proyecto asignado.
 - Las evaluaciones se realizan en diferentes tiempos ya que se tarda alrededor de 60 minutos en cada una de ellas.
 - Al finalizar cada evaluación el sistema EXSHE genera los reportes con los resultados individuales correspondientes.
 - Finalizadas todas las evaluaciones individuales EXSHE genera los resultados consolidados de cada proyecto de evaluación.
 - En total los 7 proyectos de evaluación se componen de 35 evaluaciones, de las cuales 28 son individuales (1 por cada evaluador) y 7 son generales de cada proyecto.
 - Con los resultados totales de cada evaluación se procesan los mismos agrupándolos a través de tablas que presentan el resumen de la información generada y que permiten inferir estadísticamente el nivel de usabilidad general de los entornos virtuales de aprendizaje de la UTC.

5.2. Corroboración de los resultados

Los resultados obtenidos de la evaluación de los entornos virtuales son los siguientes:

- Resultados de la calificación del nivel de usabilidad

La tabla 5.3 presenta la calificación de la usabilidad de cada proyecto evaluado y el promedio general de todos los proyectos.

Tabla 5.3.

Calificación de la usabilidad de los proyectos evaluados

Aula virtual de asignatura	Calificación	Usabilidad
Análisis y Diseño Orientado a Objetos	40,83%	Poco Usable
Proyecto Integrador I	49,53%	Medianamente Usable
Mantenimiento de Computadores	52,76%	Medianamente Usable
Redes	54,91%	Medianamente Usable
Aplicación de Bases de Datos	57,87%	Usable
Administración de Sistemas Operativos	58,66%	Usable
Ingeniería de Software	68,04%	Usable
Promedio	54,66%	Medianamente Usable

Como se puede ver, de las 7 evaluaciones realizadas 3 obtienen una calificación que corresponde al nivel de medianamente usable, las otras 3 llegan al nivel de usable y 1 tiene un nivel de poco usable (ver tabla 14 acerca de las escalas del nivel de usabilidad). Así mismo, determinado el promedio de todas las evaluaciones se obtiene que son medianamente usables, por lo tanto se puede indicar que la usabilidad de todos los EVA de la UTC está en este mismo nivel.

- **Resultados de los problemas de usabilidad encontrados**

La tabla 5.4 presenta los problemas de usabilidad inferidos en cada proyecto y la frecuencia con la que aparecen en relación al total de proyectos evaluados.

Tabla 5.4.

Problemas de usabilidad y su frecuencia en cada proyecto

Criterio	Frecuencia del criterio en la lista de problemas	Promedio de calificación en la lista de problemas
Identidad e información	6	3,76
Ayuda	5	0,60
Búsqueda	4	4,65
Elementos multimedia	3	3,67
Aspectos generales	2	4,83
Rotulado	2	4,28
Control y retoralimentación	1	3,69
Entendibilidad y facilidad en la interacción	1	3,84
Estructura y navegación	1	3,63
Layout de la página	1	3,44

Con relación a esta tabla se puede determinar que de los 10 criterios de evaluación todos son parte de la lista de problemas, se muestran de acuerdo a su frecuencia en el total de los proyectos evaluados, siendo el más recurrente el criterio de identidad e información que está presente en 6 de las 7 evaluaciones, le siguen ayuda en 5 evaluaciones, búsqueda en 4, elementos multimedia en 3 y el resto de criterios en 2 y en 1 evaluación. En conclusión, esta información permite deducir que los problemas más importantes de usabilidad en los entornos virtuales de aprendizaje de la UTC tienen que ver con el aspecto de identidad e información y los sistemas de búsqueda y ayuda que se aplican en estos sitios.

- **Resultados de las recomendaciones generadas**

La tabla 5.5 presenta las recomendaciones de usabilidad inferidas para cada proyecto y la frecuencia con la que aparecen en relación al total de proyectos evaluados.

Tabla 5.5.

Recomendaciones generadas y su frecuencia en cada proyecto

Subcriterio	Frecuencia en las evaluaciones	Recomendación
AG8	7	Se debe dar soporte al usuario en otros idiomas
EN14	7	Se debe incluir un mapa del sitio para acceder directamente a los contenidos sin navegar
EN7	7	Es necesario mejorar la caracterización de los enlaces, deben indicar su estado (visitados, activos)
AG10	6	El contenido del sitio debe ser actualizado periódicamente
AG9	6	Es necesario mejorar la traducción del sitio, ésta debe ser completa y correcta
BU3	6	La opción de búsqueda debe estar fácilmente accesible desde cualquier parte del sitio
BU8	6	El sistema de búsqueda debe incluir asistencia al usuario en el caso de no ofrecer resultados para una consulta dada
II3	6	El eslogan o tagline debe ser adecuado al objetivo del sitio
II6	6	Se debe ofrecer información sobre la protección de datos de carácter personal o los derechos de autor de los contenidos del sitio web
II7	6	Se debe ofrecer información sobre el autor, fuentes, fechas de creación y de revisión en artículos, noticias e informes
RO5	6	Las URLs de páginas internas deben ser claras
AG2	5	Los contenidos y servicios ofrecidos deben ser precisos y completos
AY1	5	El enlace a la sección de ayuda debe estar colocado en una zona visible y estándar
AY2	5	Debe haber fácil acceso y retorno del sistema de ayuda
AY3	5	Es necesario ofrecer ayuda contextual para las tareas complejas
BU1	5	Si es necesaria la búsqueda, debe estar accesible desde todas las páginas del sitio
EF5	5	Se deben colocar metáforas visuales reconocibles y comprensibles por cualquier usuario (íconos)
EN11	5	Las imágenes de enlace deben indicar el contenido al que se va a acceder
II5	5	Debe existir mecanismos de contacto
RO4	5	La URL de la página principal debe ser correcta, clara y fácil de recordar
CR2	4	Se debe informar al usuario acerca de lo que está pasando en el sitio o proceso solicitado

CONTINÚA →

CR4	4	Se debe mejorar los sistemas de validación antes de que el usuario envíe información para tratar de evitar errores
CR5	4	Cuando se produce un error, se debe informar al usuario de forma clara y no alarmista acerca de lo ocurrido y de cómo solucionar el problema
CR6	4	Se debe mejorar el control del tiempo de respuesta
EM3	4	Las fotografías incluidas deben tener una resolución adecuada
EM4	4	El uso de imágenes o animaciones deben proporcionar algún tipo de valor añadido
EN8	4	No deben haber enlaces redundantes
EM2	3	Las fotografías incluida en el contenido deben ser comprensibles para el usuario
EM6	3	El uso de sonido debe proporcionar algún tipo de valor añadido
EN10	3	No deben haber enlaces que lleven a la misma página que se está visualizando
EN9	3	No deben haber enlaces rotos
LA2	3	Se debe evitar la sobrecarga informativa
LA3	3	Las interfaces deben ser limpias, se debe evitar el ruido visual
BU5	2	El sistema de búsqueda debe ser simple y claro
BU6	2	La búsqueda debe incluir un sistema avanzado
BU7	2	Los resultados de la búsqueda se deben mostrar de forma comprensible para el usuario
CR10	2	Si existen tareas de varios pasos, se debe indicar al usuario en cuál está y cuántos faltan para completar la tarea
CR3	2	Se debe informar al usuario de lo que ha pasado en el sistema
EF7	2	Si el usuario tiene que rellenar un campo, las opciones disponibles se deben seleccionar en lugar de escribirlas
EM1	2	Las fotografías incluidas en el contenido deben estar bien recortadas
EM5	2	Se debe evitar el uso de animaciones cíclicas
EN13	2	Deben existir elementos de navegación que orienten al usuario acerca de dónde está y cómo deshacer su navegación (migas de pan)
II4	2	Se debe incluir información sobre el sitio web, empresa o institución
RO1	2	Se debe mejorar el sistema de rotulación, los rótulos deben ser significativos
AG3	1	La estructura general del sitio web debe estar orientada al usuario
AG4	1	La apariencia (Look & Feel) debe ser coherente con los objetivos, características, contenidos y servicios del sitio web
AG6	1	El diseño general del sitio web deber ser consistente y coherente
AG7	1	El contenido debe estar publicado en el idioma del usuario
AY4	1	Se deben incluir preguntas frecuentes coherentes y precisas
AY5	1	La redacción de las preguntas frecuentes debe ser correcta
BU4	1	La caja de texto de la opción de búsqueda debe ser lo suficientemente ancha
CR1	1	El sistema debe permitir al usuario tener el control sobre la interfaz
CR7	1	Se debe evitar que las ventanas del sitio anulen o se superpongan a la ventana del navegador

CONTINÚA →

CR9	1	Se debe evitar que el usuario tenga que descargar plugins o complementos adicionales
EF1	1	El lenguaje empleado deber ser claro y conciso para que el usuario lo pueda entender con facilidad
EF3	1	Cada párrafo presentado debe expresar una sola idea
EF4	1	El uso de los controles sobre la interfaz debe ser consistente
EF6	1	Si se usan menús desplegables, el orden en el que se presenten debe ser coherente
EN1	1	Se debe evitar la pantalla de bienvenida, esto puede provocar demora y cansancio en el usuario
EN12	1	Debe existir un enlace para volver al inicio en cada página
EN2	1	La estructura de organización y navegación del sitio debe ser adecuada
EN5	1	Debe existir equilibrio entre profundidad y anchura en el caso de utilizar una estructura jerárquica para presentar los contenidos
EN6	1	Todos los enlaces deben ser fácilmente reconocibles como tales
II1	1	La etiqueta de identidad o logotipo debe ser significativo, identificable y suficientemente visible
II2	1	La etiqueta de identidad del sitio, debe estar incluida en todas las páginas
LA1	1	Se deben aprovechar las zonas de alta jerarquía informativa de la página para los contenidos de mayor relevancia
LA10	1	Se debe evitar el texto parpadeante o deslizante
LA4	1	Deben existir zonas en blanco entre los objetos informativos de la página para que el usuario descanse la vista
LA5	1	El espacio visual de la página debe ser usado correctamente
LA6	1	Debe existir jerarquía visual para expresar las relaciones del tipo "parte de" entre los elementos de la página
LA7	1	Se debe controlar la longitud de página
LA8	1	La versión impresa de la página debe ser correcta
LA9	1	El texto de la página se debe leer sin dificultad
RO3	1	El título de las páginas debe ser correcto y planificado
RO6	1	Las URLs de las páginas internas deben ser permanentes

De igual manera que la lista de problemas, las recomendaciones se presentan en orden descendente de acuerdo a su frecuencia en los proyectos evaluados, tienen que ver directamente con los subcriterios evaluados, por lo tanto son mucho más específicas. Aparecen en la lista 75 recomendaciones relacionadas con 75 subcriterios, si se toma en cuenta que son un total de 83 subcriterios, significa que la mayoría de ellos tiene una recomendación inferida como resultado de las evaluaciones realizadas.

En general se puede concluir que las recomendaciones que se repiten con mayor frecuencia tienen que ver con los criterios de aspectos generales, estructura y navegación, búsqueda, identidad e información, rotulado, ayuda, entendibilidad y facilidad, control y retroalimentación y elementos multimedia, las mismas que deben ser aplicadas también a la mayoría de aulas virtuales de la UTC.

- **Plan de mejoras**

De los resultados obtenidos en las evaluaciones se diseña el plan de mejoras que indica una serie de tareas que se deberían ejecutar para corregir los errores de usabilidad encontrados.

A continuación en la tabla 54 se describe el plan de mejoras para la usabilidad de los EVA de la UTC.

Tabla 5.6.

Plan de mejoras para la usabilidad de los EVA de la UTC

PLAN DE MEJORAS										
NOMBRE DEL PROYECTO:	Mejoramiento de la usabilidad de los Entornos Virtuales de Aprendizaje de la UTC									
FECHA DE ELABORACIÓN:	12 de Agosto del 2015									
ACTIVIDAD QUE REQUIERE MEJORA	MEJORA PLANIFICADA	RESPONSABLE	FECHA DE INICIO	FECHA DE FINALIZACION	REPROGRAMACION	INDICADOR	PRESUPUESTO	% DE AVANCE	RESPONSABLE DEL SEGUIMIENTO	OBSERVACIONES
Conocimiento de los docentes para el diseño de aulas virtuales	Capacitación a los docentes sobre el diseño de aulas virtuales y el manejo de la plataforma Moodle	Unidad de Desarrollo Académico	Ciclo Académico Octubre 2015 – Marzo 2016	Por verificar con la unidad responsable		Programación analítica de la capacitación Registro de asistencias			Vicerrectorado	
Conocimiento de los docentes para el diseño de aulas virtuales	Elaboración de una guía de diseño de aulas virtuales	Unidad de Desarrollo Académico	Ciclo Académico Octubre 2015 – Marzo 2016	Por verificar con la unidad responsable		Guía de diseño de aulas virtuales Certificado de aprobación de la guía			Vicerrectorado	

CONTINÚA →

Diseño de EVA	Rediseño de la estructura general de los EVA	Administrador de la plataforma Moodle	Ciclo Académico Octubre 2015 – Marzo 2016	Por verificar con la unidad responsable	Recomendaciones inferidas en la evaluación de la usabilidad incorporadas en los EVA.			Vicerrectorado	
Conocimiento acerca de la usabilidad de los EVA	Evaluación de la usabilidad de los EVA	Administrador de la plataforma Moodle	Abril 2016	Por verificar con la unidad responsable	Resultados de la evaluación realizada			Vicerrectorado	

5.3. Conclusiones

- La validación del sistema EXSHE se realiza en los entornos virtuales de aprendizaje de la UTC, se aplica la medición a una muestra que corresponde a las aulas virtuales de la carrera de Ingeniería en Informática y Sistemas Computacionales; participan como evaluadores, cuatro profesionales del campo de la ingeniería de software y del diseño gráfico computarizado.
- Se ejecutan 7 proyectos de los cuales se obtienen 28 evaluaciones individuales y 7 evaluaciones consolidadas, de igual manera se obtiene un promedio general de todos los resultados generados.
- Los resultados reflejan que los entornos virtuales de aprendizaje de la UTC tienen una calificación de 54.66% que corresponde a un nivel de medianamente usable. (Ver tabla 5.3: Calificación de la usabilidad de los proyectos evaluados)
- La lista de problemas está compuesta por los 10 criterios medidos, esto significa que todos presentan conflictos de usabilidad, sin embargo los de mayor incidencia son: identidad e información, ayuda, búsqueda y elementos multimedia. (Ver tabla 5.4: Problemas de usabilidad y su frecuencia en cada proyecto)
- De los 83 subcriterios de evaluación, 75 tienen recomendaciones de mejora, los criterios de mayor incidencia son: aspectos generales, estructura y navegación, búsqueda, identidad e información, rotulación y ayuda. (Ver tabla 5.5: Recomendaciones generadas y su frecuencia en cada proyecto)

CAPÍTULO VI

CONCLUSIONES Y RECOMENDACIONES

6.1. Conclusiones

- El análisis de la literatura recogida permitió determinar que la usabilidad es uno de los factores de la calidad del software más importante, esto significa que la medición de este parámetro es necesaria para conocer cuáles son los aspectos en los que las aplicaciones deben mejorar con el fin de brindar mayor facilidad de uso y satisfacción a sus usuarios. De igual manera se determinó que uno de los métodos de evaluación más económico y eficaz para medir la usabilidad, es el método de evaluación heurística.
- La metodología aplicada en la creación del sistema experto estableció un ciclo de vida de nueve etapas de desarrollo, una de las fases más importantes fue la fase de Adquisición del Conocimiento I, porque es en donde se desarrolló el prototipo de prueba del sistema, en este caso el prototipo pasó todas las pruebas funcionales y permitió continuar con el desarrollo aplicando las mismas técnicas utilizadas, ya que demostraron ser eficaces para el cumplimiento de los objetivos propuestos.
- De todas las fuentes del conocimiento que se consultaron para recoger el mismo, se consolidaron en este proyecto: Jakob Nielsen, Yusef Hassan, Francisco Martín y María del Carmen Suárez. Lamentablemente los 18 entrevistados pertenecientes a empresas de desarrollo de software localizadas en la Región 3 y en la ciudad de Quito, demostraron no tener un conocimiento sólido acerca de la usabilidad y que no aplican formalmente ninguna técnica de usabilidad en sus proyectos.

- La validación del sistema EXSHE se realizó en los entornos virtuales de aprendizaje de la carrera de Ingeniería en Informática y Sistemas Computacionales de la UTC, los resultados muestran que estos entornos se encuentra en el nivel de “medianamente usables” obteniendo una calificación del 54,66%; esto permite indicar por lo tanto, que todos los EVA de la UTC son medianamente usables y que los problemas más importantes de usabilidad tienen que ver con los criterios de identidad e información, ayuda, búsqueda y elementos multimedia.
- La evaluación que realiza EXSHE está compuesta por 10 criterios y 83 subcriterios, la lista de problemas se genera con los criterios que han obtenido una calificación promedio menor a 5 y las recomendaciones con los subcriterios que han obtenido una calificación individual menor a 5. En el caso de los EVA de la UTC, aparecen en la lista de problemas los 10 criterios y se generan 75 recomendaciones relacionadas con el mismo número de subcriterios, se concluye por lo tanto, que estos sitios presentan problemas de usabilidad en todos los criterios de evaluación.
- Se puede concluir además, en relación a los EVA de la UTC, que la mayor parte de problemas están relacionados con el diseño de la estructura general de las aulas virtuales, es decir tienen una orientación técnica que la puede resolver el administrador de la Plataforma Moodle. Sin embargo, a esto se debe sumar el hecho de que los docentes que son quienes crean los cursos, ocasionan problemas de usabilidad a la hora de subir y diseñar el contenido de cada aula.
- Finalmente, es necesario indicar que los resultados de cada evaluación están sujetos a la particularidad del evaluador, por lo tanto la evaluación es subjetiva, pues depende de la experiencia del evaluador y del campo profesional en el que se desenvuelve, de ahí la importancia de tener al menos cuatro evaluadores por proyecto.

6.2. Recomendaciones

- Todos los proyectos de desarrollo de software deben considerar factores de usabilidad, es importante que el diseño se haga centrado en los potenciales usuarios de las aplicaciones ya que se trata de entregarles productos que satisfagan sus expectativas y necesidades, para ello se recomienda a los profesionales de este campo, capacitarse en temas de usabilidad y en general en temas que les ayuden a desarrollar productos que garanticen su calidad a los usuarios.
- Realizar mediciones de usabilidad en forma continua a los productos desarrollados y a los productos en desarrollo, una de las ventajas de la evaluación heurística es que es aplicada sobre las interfaces de las aplicaciones, es decir la medición se puede realizar desde el primer prototipo funcional, permitiendo encontrar errores en forma temprana y reducir costos de depuración tanto en tiempo como en dinero.
- Aplicar el plan de mejoras para la usabilidad de los EVA de la UTC sobre todo en cuanto a su diseño estructural en general, otorgando especial atención en los criterios de identidad e información, ayuda, búsqueda y elementos multimedia que son los de mayor problema de acuerdo a las evaluaciones realizadas.

REFERENCIAS BIBLIOGRÁFICAS

1. Acuña Herrera, M. E., & Vasco Amores, K. P. (2012). Análisis de usabilidad y accesibilidad en sistemas de información utilizados por personas con capacidades especiales. Latacunga-Ecuador.
2. Acuña Silvia Teresita. (2003). MPAS: Metodología participativa para un sistema basado en conocimiento aplicado al cultivo de algodón. Copy Red, S. A., Argentina.
3. Aguaded, Cumberras C. (2006). Usabilidad en las páginas web. Recuperado de: <http://revistas.um.es/index.php/eglobal/article/viewFile/392/353!>, el 25 de Julio del 2014.
4. Alcalá, L. (2010). Medida de la usabilidad en aplicaciones de escritorio. Recuperado de: http://sedom.es/wp-content/themes/sedom/pdf/4cbd571ba0446art_3.pdf, el 22 de Junio de 2014.
5. Alva Obeso, M. E. (2009). Metodología de medición y evaluación de la usabilidad en sitios Web educativos. España: Ediuno - Universidad de Oviedo. Recuperado de: <http://site.ebrary.com/lib/espesp/docDetail.action?docID=10311112>, el 12 de Junio del 2014.
6. Ayala Organista M., & Muñoz Benítez S. (n.d.). Sistema experto: Elección de jugadores de fútbol a partir de unas determinadas características seleccionadas. Recuperado de: <http://www.it.uc3m.es/jvillena/irc/practicas/09-10/18mem.pdf>, el 20 de Agosto del 2014.
7. Calvo, A., Ortega, S., Valls, A., & Zapata, M. (Septiembre de 2011). Evaluación de la Usabilidad. Recuperado de: http://openaccess.uoc.edu/webapps/o2/bitstream/10609/9862/1/PID_00159105.pdf, el 12 de Septiembre del 2014.

8. Campos, A. (2013). Scrib. Libertad y control por parte del usuario, Recuperado de: <http://es.scribd.com/doc/173269969/10-reglas-heuristicas-de-usabilidad-de-Jakob-Nielsen-pdf>, el 20 de Julio del 2014.
9. Candamil M., (2012), Análisis de sitios web universitarios mediante evaluaciones heurísticas, publicado en ene 29, 2012 en Arquitectura de información, Investigación, Recuperado de: <http://www.mauriciocandamil.com/analisis-de-sitios-web-universitarios-colombianos-mediante-evaluaciones-heuristicas/>, el 12 de Mayo del 2014.
10. Carbonell, D. (20 de Febrero de 2012). Las Leyes de la Simplicidad. Recuperado de Las Leyes de la Simplicidad: <http://www.slideshare.net/dfcarbonell/las-leyes-de-la-simplicidad>, el 12 de Mayo del 2012.
11. Carrillo Verdún, J. (1987). Metodología para el desarrollo de sistemas expertos. Informática. Recuperado de: <http://oa.upm.es/1057/>, el 20 de Mayo del 2014.
12. Curiel A. (2013). Sistema experto basado en reglas para una aplicación de monitorización de producción industrial. Universidad Politécnica de Valencia, Valencia - España. Recuperado de: <https://riunet.upv.es/bitstream/handle/10251/39143/TFM%20-%20IARFID%20-%20Alexander%20Curiel%20Robles.pdf?sequence=1>, el 25 de Mayo del 2012.
13. Del Valle, E. (19 de Junio de 2013). ¿Cumple tu web los 10 principios heurísticos de la usabilidad? Obtenido de ¿Cumple tu web los 10 principios heurísticos de la usabilidad?: Recuperado de: <http://www.socialancer.com/los-10-principios-heuristicos-de-la-usabilidad-en-tu-web/>, el 20 de Agosto del 2014.
14. Doctorpc. (28 de 07 de 2010), 10 principios generales de la usabilidad en sitios web, Recuperado de: <http://notebookypc.com/10-principios-generales-de-la-usabilidad-en-sitios-web>, el 24 de Junio del 2014.
15. Dtoub, S. (2009). Evaluating Information Architecture: A Practical Guide to Assessing Web Site Organization. ARGUS Associates. Recuperado de: http://argus-acia.com/white_papers/evaluating_ia.html, el 24 de Junio del 2014.

16. Elliott, J. (1990). La investigación-acción en educación (ilustrada, reimpresa). Ediciones Morata. Madrid – España.
17. Feigenbaum, E.A. (1980). Knowledge Engineering: the Applied Side of Artificial Intelligence. Report STAN-CS-80-812. Department of Computer Science, Stanford University. Recuperado de: <http://cs.stanford.edu/people/eaf/wordpress/papers/>, el 26 de Julio del 2014.
18. Fernández E., García J., Tornero I.y Sierra A. (2011). Evaluación de la usabilidad de un sitio web educativo y de promoción de la salud en el contexto universitario, publicado en: Edutec-e. Revista Electrónica de Tecnología Educativa. Núm. 37 / Sep.2011, issn: 1135-9250, Recuperado de: http://edutec.rediris.es/Revelec2/Revelec37/pdf/Edutec-e_n37_Fernandez_Garcia_Tornero_Sierra.pdf, el 20 de Septiembre del 2014.
19. Finelli, F. (28 de Junio de 2011). 10 REGLAS HEURÍSTICAS DE USABILIDAD DE JAKOB NIELSEN, braintive. Recuperado de: <http://web.braintive.com/10-reglas-heuristicas-de-usabilidad-de-jakob-nielsen/>, el 21 de Junio de 2014.
20. Floría, A. (2011). Principios - Nielsen. SIDAR. Recuperado de: <http://www.sidar.org/recur/desdi/traduc/es/visitable/Herramientas.htm>, el 23 de Junio del 2014.
21. Friedman-Hill Ernest. (2008, November 5). Jess® The Rule Engine for the Java™ Platform., Recuperado de: <http://herzberg.ca.sandia.gov/>, el 20 de Marzo del 2015.
22. García, E. (12 de Abril de 2011). Web-usable. Recuperado de <http://web-usable.com/?p=97>, el 21 de Junio de 2014.
23. Gilb, T., (1988) Principles of Software Project Management, Addison-Wesley. New York – EE UU.
24. Hassan Montero, Y., & Martín Fernández, F. J. (2004). Propuesta de adaptación de la metodología de diseño centrado en el usuario para el desarrollo de sitios web accesibles. Revista española de documentación científica, 27(3). Recuperado de

- <http://redc.revistas.csic.es/index.php/redc/article/viewArticle/156>, el 20 de Mayo del 2014.
25. Hassan Montero, Y. & Martín Fernández, F. (2003). Guía de Evaluación Heurística de Sitios Web. En: No Solo Usabilidad, n° 2, <nosolousabilidad.com>. ISSN 1886-8592, Recuperado de: <http://www.nosolousabilidad.com/>, el 20 de Junio del 2014.
 26. Hernández, A. R., & Cruz, Y. R. (2009). La usabilidad. Un acercamiento a su utilización en la UCI. Serie Científica, 2(2). Recuperado de: <http://publicaciones.uci.cu/index.php/SC/article/view/93>, el 26 de Junio del 2014.
 27. Hidalgo L. (1998). Inteligencia artificial y Sistemas Expertos. Universidad de Córdoba. ISBN: 9788478013463, Córdoba – España.
 28. IEEE Std 610.12-1990, IEEE Standard Glossary of Software Engineering Terminology, identifies terms currently in use in the field of Software Engineering. Standard definitions for those terms are established. Recuperado de: http://www.mit.jyu.fi/ope/kurssit/TIES462/Materiaalit/IEEE_SoftwareEngGlossary.pdf, el 26 de Junio del 2014.
 29. Isabel Casanova, L. (02 de 05 de 2011). Principios heurísticos. Artículo publicado (Blog Escritura). Manning Publications. Recuperado de: <http://evaluación-y-experiencia-de-usuario/>, el 24 de Junio del 2014.
 30. ISO/IEC FDIS 9126-1 Information technology — Software product quality — Part 1: Quality model. Reference number ISO/IEC FDIS 9126-1:2000(E), Recuperado de: http://www.iso.org/iso/catalogue_detail.htm?csnumber=22749, el 26 de Junio del 2014.
 31. Markessinis, A. (15 de Junio de 2014). Diseño Web, Usabilidad. Recuperado de: <http://andreamarkessinis.com/blog/principios-heuristicos-usabilidad/>, el 26 de Julio del 2014.
 32. Molich, R., y Nielsen, J. (1990). Mejorar el diálogo hombre-máquina, Communications of the ACM 33, 3 (marzo), 338-348. Recuperado de: <http://dl.acm.org/citation.cfm?id=77486>,

33. Nielsen, J. (1994a). Aumento de la capacidad explicativa de heurísticas de usabilidad. Proc. ACM CHI'94 Conf. (Boston, MA, April 24-28), 152-158., Recuperado de: <http://www.nngroup.com/people-jakob-nielsen-publications/>, el 26 de Junio del 2014.
34. Nielsen, J. (1994b). La evaluación heurística. En Nielsen, J., y Mack, RL (Eds.), Usabilidad métodos de inspección, John Wiley&Sons, Nueva York, NY. – EE.UU.
35. Nielsen, J. (1995). How to Conduct a Heuristic Evaluation, Nielsen Norman Group, publicado el 01 de Enero de 1995 en: <http://www.nngroup.com/articles/how-to-conduct-a-heuristic-evaluation/>, Recuperado el 25 de Octubre del 2014.
36. Nielsen, J. (2011). Braintive. Publicado el 28 de junio del 2011. Recuperado de: <http://web.braintive.com/10-reglas-heuristicas-de-usabilidad-de-jakob-nielsen/>, el 24 de Junio del 2014.
37. Nielsen, J. (2012). Usability 101: Introduction to Usability. Nielsen Norman Group, publicado el 04 de enero del 2012. Disponible en: <http://www.nngroup.com/articles/usability-101-introduction-to-usability/>. Recuperado el 25 de Julio del 2014.
38. Nielsen, J., y Molich, R. (1990). La evaluación heurística de interfaces de usuario, Proc. ACM CHI'90 Conf. (Seattle, WA, 1-5 de abril), 249-256., EE.UU.
39. Obeso, A., M. E. (2009b). Metodología de medición y evaluación de la usabilidad en sitios Web educativos. España: Ediuno - Universidad de Oviedo. Recuperado de: <http://site.ebrary.com/lib/espesp/docDetail.action?docID=10311112>., el 18 de Septiembre del 2014.
40. Piattini M. (2012). Medusas: Mejora y Evaluación del Diseño, Usabilidad, Seguridad y Mantenibilidad del Software. IDI-20090557-60, España: UCLM.

41. Pressman, R., (2010). Ingeniería del software. Un enfoque Práctico, Séptima Edición, McGRAW-HILL INTERAMERICANA EDITORES, S.A. DE C.V., México D.F. – México.
42. Proceedings of 2011 Fifth Asia Modelling, AMS 2011Conference. pp. 68-72. Recuperado de: http://www.ijens.org/Vol_12_I_05/1216805-7373-IJECS-IJENS.pdf, el 12 de Septiembre del 2014.
43. Roa M. y Caldera Jorge, (2015). Evaluación Heurística de las sedes web de los medios digitales: El País y El Mundo, publicado en Cuadernos de Documentación Multimedia, ISSN: 1575-9733, Universidad Complutense de Madrid, Recuperado de: <http://revistas.ucm.es/index.php/CDMU/article/viewFile/38340/37096>, el 22 de Septiembre del 2014.
44. Rodríguez, A. C.-F., Santamaría, S. O., Sáez, A. V., & Lluch, M. Z. (2011). Evaluación de la usabilidad. UOC. Recuperado de: [http://www.exabyteinformatica.com/uoc/Informatica/Evaluacion_de_la_usabilidad/Evaluacion_de_la_usabilidad_\(Intro\).pdf](http://www.exabyteinformatica.com/uoc/Informatica/Evaluacion_de_la_usabilidad/Evaluacion_de_la_usabilidad_(Intro).pdf), el 26 de Julio del 2014.
45. Romero M., Bartó C., Cuevas J., Muñoz R. & Damiano L. (2014). Seguimiento y Control de Diseño de Proyecto de Ingeniería del Conocimiento. Recuperado de: <http://conaiisi.frc.utn.edu.ar/PDFsParaPublicar/1/schedConfs/1/237-648-1-DR.pdf>, el 18 de Septiembre del 2014.
46. Salazar Jácome, M. E. (2013). Marco de integración de la usabilidad en el proceso de desarrollo de software en las principales empresas desarrolladoras del Ecuador (Quito, Guayaquil y Cuenca). Latacunga – Ecuador.
47. Sánchez, W. (Agosto de 2011). La usabilidad en Ingeniería de Software: definición y características. Recuperado de: <http://rd.udb.edu.sv:8080/jspui/bitstream/123456789/1184/1/2.%20La%20usabilidad%20en%20Ingenieria%20de%20Software-%20definicion%20y%20caracteristicas.pdf>, el 22 de Junio de 2014,
48. Sierra Enrique A., Hossian A. & García Martínez R. (2007). Sistemas expertos que recomiendan estrategias de instrucción. un modelo para su desarrollo. Revista

- Latinoamericana de Tecnología Educativa-RELATEC 1(1): 33–47., Recuperado de: <http://relatec.unex.es/article/view/7>, el 22 de Junio del 2014.
49. Sivaji, A., A. Abdullah and A. G. Downe (2011). Usability Testing Methodology: Effectiveness of Heuristic Evaluation in EGovernment Website Development, Recuperado de: http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=5961243&url=http%3A%2F%2Fieeexplore.ieee.org%2Fexpl%2Fabs_all.jsp%3Farnumber%3D5961243, el 25 de Septiembre del 2014.
50. Suárez, E. (27 de Octubre de 2013). Dispersium. Recuperado de: http://www.useit.cat/es/servicios/usabilidad/evaluaciones_heuristicas, el 22 de Agosto del 2014.
51. Suárez, M. (2011, February). Sirius: Sistema de Evaluación de la Usabilidad Web Orientado al Usuario y basado en la Determinación de Tareas Críticas. UNIVERSIDAD DE OVIEDO-Departamento de Informática, Oviedo. Recuperado de: <http://www.di.uniovi.es/~cueva/investigacion/tesis/Sirius.pdf>, el 20 de Julio del 2014.
52. Vall A. y Marcos M., (2012). Evaluación de sitios web multilingües: metodología y herramienta heurística Mar, publicado en: El profesional de la información, 2012, mayo-junio, v. 21, n. 3. ISSN: 1386-6710, Recuperado de: <http://eprints.rclis.org/17065/1/254-260.pdf>, el 14 de Julio del 2014.
53. Villa, L, (2003). La usabilidad y el ROI. Obtenido de Alzado.org: Recuperado de: http://www.alzado.org/articulo.php?id_art=30, el 26 de Junio del 2014.