



UNIVERSIDAD CENTRAL DE VENEZUELA
FACULTAD DE CIENCIAS
ESCUELA DE COMPUTACIÓN

Trabajo Especial de Grado:
Técnica para la Migración de Software
basada en Modelos Cognitivos del Usuario

Trabajo Especial de Grado presentado ante la Ilustre
Universidad Central de Venezuela para optar al título de
Licenciado en Computación.

Presentado por:

Br. Florecita Palomino. C.I.: 16.225.037

Br. Laura Malavé. C.I.: 16.615.031

Tutores:

Profa. Nora Montaña

Prof. Andrés Sanoja

Caracas, septiembre de 2007

Acta del Veredicto

Quienes suscriben, miembros del jurado designado por la Escuela de Computación de la Facultad de Ciencias de la Universidad Central de Venezuela para examinar el Trabajo Especial de Grado de los bachilleres FLORECITA KAREM PALOMINO LI y LAURA ALEJANDRA MALAVÉ BRICEÑO, titulares de las cédula de identidad No. 16.225.037 y 16.615.031 respectivamente, bajo el título: “Técnica para la Migración de Software basada en Modelos Cognitivos del Usuario”, a los fines de cumplir con el requisito legal para optar al grado de Licenciado en Computación, dejan constancia de lo siguiente:

Una vez suficientemente leído este trabajo por cada uno de los miembros del jurado, se fijó el día veintiocho de septiembre a las 10:00 am., fecha para la defensa de manera pública. Mediante una exposición oral de este Trabajo Especial de Grado, realizada en la Escuela de Computación, aula PB-III, en la fecha acordada, luego de lo cual respondieron satisfactoriamente a las preguntas que le fueron formuladas por el jurado, todo ello conforme dispuesto a la Ley de Universidades y demás normativas vigentes de la Universidad Central de Venezuela.

El jurado decidió APROBARLO con la nota _____ puntos.

En fe de lo cual se levanta la presente acta a los veintiocho días del mes de septiembre de dos mil siete.

Prof. Andrés Sanoja (Tutor)

Prof. Jossie Zambrano (Jurado)

Prof. Eleonora Acosta (Jurado)

Agradecimientos

A nuestro Dios, en quien creemos, simplemente por su amor que lo hace posible todo.

A nuestros padres, hermanos, abuelos, tías y tíos, por su inmenso amor, cariño, dedicación y por estar siempre dispuestos a ayudarnos en todo lo necesario, a toda hora. Por estar muy cerca y muy pendientes de la consecución de este trabajo. Sin su inigualable cooperación no hubiera sido posible cumplir nuestro objetivo.

A nuestros tutores, la profesora Nora Montaña y el profesor Andrés Sanoja, por sus consejos oportunos y acertados, así como el apoyo profesional brindado para el desarrollo de este Trabajo Especial de Grado.

A los siguientes bachilleres por su valiosa participación en la aplicación de la técnica de migración propuesta en varias herramientas de software: Br. Diana Barrios, Br. Gabriela Millán, Br. Jesús Urbáez, Br. Elisa Conesa, Br. Pedro Contreras, Br. William Domínguez, Br. Tomás Angulo, Br. Andrés Blanco, Br. Jonathan Vásquez, Br. Rendall Navarro, Br. Lily Tsang, Br. Jorge Valdéz, Br. Elicenda Martínez, Br. Virginia Rodríguez, Br. Yolibert Ghaleb, Br. Jéssica González, Br. Luis León, Br. Alejandro Chacón, Br. Mayerling Mendoza, Br. Corina Ruiz y Br. Sergio Vera.

A nuestros amigos y amigas, y en especial a M.A.J.C., por todas sus críticas constructivas, para el enriquecimiento de este documento y de nuestras vidas.

A nosotras mismas, mutuamente, por no desmayar ante las adversidades y complementarnos para alcanzar nuestra meta. Gracias compañera y amiga.

Y finalmente, a todas aquellas personas que de alguna manera y en forma desinteresada, quizá sin saberlo, también nos apoyaron.

A todos ellos, gracias totales.

Resumen

Las migraciones de software son procesos de cambio que pueden completarse satisfactoriamente o no, es por eso que deben ser conducidos de forma planificada para evitar posibles inconvenientes en el proceso y asegurar que el nuevo software sea tan productivo como el anterior. En este documento se muestra la estrecha relación que tiene una migración con los aspectos humanos y el modelo mental del usuario, así como la importancia de la usabilidad en este tema, y cómo estos factores pueden ser aprovechados para garantizar el éxito del proceso.

En atención a lo planteado, el siguiente Trabajo Especial de Grado tiene como propósito desarrollar una propuesta que haga frente a un proceso de migración de software cuando el usuario está arraigado al uso de una herramienta.

Para lograr predecir los resultados de una migración, se plantea una técnica basada en un análisis de compatibilidad de software, nutrida de técnicas propias de usabilidad, cuya finalidad es, proyectar el modelo mental que posee un usuario, de un software a otro, a partir de la proyección de los modelos conceptuales de los productos de software implicados en la migración.

La técnica es aplicada en un caso de estudio entre dos navegadores web, donde se realiza un análisis exhaustivo para demostrar su compatibilidad y luego se lleva a cabo un trabajo de campo en el que intervienen usuarios finales, cuyos resultados permiten comprobar la efectividad de esta propuesta. Finalmente, la técnica fue ampliamente aceptada por evaluadores que la aplicaron en diversos sistemas.

Palabras Clave: migración, software, usuario, modelo mental, modelo conceptual, proyección, compatibilidad, técnica.

Índice

Introducción	1
Capítulo 1: Marco Teórico	3
1.1 Contexto para la migración de Software	3
1.1.1 Definición de Migración Tecnológica	3
1.1.2 Esquemas de Procesos de Migración	4
1.1.3 Problemas durante el proceso	6
1.2 Aspectos Humanos en el proceso de migración	7
1.2.1 Modelo Mental del usuario	10
1.2.2 Modelo Conceptual del Sistema	11
1.2.3 Tareas y Metáforas	12
1.3 Usabilidad y el proceso de migración	13
1.3.1 Definición de Usabilidad	14
1.3.2 Usabilidad del Software	17
1.3.3 Hacia la Madurez del software	18
1.3.4 Técnicas de Usabilidad	22
1.3.5 Métricas de Usabilidad	28
Capítulo 2: Marco Metodológico	30
2.1 Problema	30
2.2 Objetivo	30
2.3 Técnica de Migración de Software	30
2.3.1 Extraer el modelo mental del usuario a través del modelo conceptual del software origen	32
2.3.2 Proyectar el modelo conceptual del software origen hacia el modelo conceptual del software destino	34
2.3.3 Analizar la compatibilidad entre los productos de software	35
2.3.4 Validar con los usuarios.....	39
2.4 Guía práctica para la aplicación de la técnica	42
Capítulo 3: Marco Experimental	44
3.1 Aplicación de la técnica para una Migración de Navegadores Web	44
3.2 Resultados de la aplicación de la técnica en otras herramientas de software 74	
3.2.1 Acrobat Reader vs. KPDF	75
3.2.2 Winzip vs. 7-Zip	75
3.2.3 MSN Messenger vs. Alvaro's MSN	76
3.2.4 Microsoft Office Word vs. Open Office Writer.....	77
3.3 Efectividad de la técnica	78
Conclusiones	86
Recomendaciones	88
Referencias	89

Índice de Tablas

Tabla 1-1 Tasa de éxitos del usuario propuesta por Nielsen	28
Tabla 2-1 Ejemplo de Análisis de compatibilidad de t _i	38
Tabla 2-2 Ejemplo de Resultados de Compatibilidad de software por tareas	38
Tabla 2-3 Ejemplo de Resultados de las pruebas por usuario	41
Tabla 3-1 Barra de dirección de IE7 vs. Barra de Navegación de MF2.0.....	48
Tabla 3-2 Barra de herramientas de IE7 vs. Barra de pestañas de MF2.0.....	50
Tabla 3-3 Análisis de compatibilidad para la tarea: Ir a una dirección web	52
Tabla 3-4 Análisis de compatibilidad para la tarea: Abrir nueva ventana.....	54
Tabla 3-5 Análisis de compatibilidad para la tarea: Abrir nueva pestaña	56
Tabla 3-6 Análisis de compatibilidad para la tarea: Ir atrás o Regresar.....	57
Tabla 3-7 Análisis de compatibilidad para la tarea: Ir adelante	58
Tabla 3-8 Análisis de compatibilidad para la tarea: Actualizar.....	60
Tabla 3-9 Análisis de compatibilidad para la tarea: Agregar a favoritos	62
Tabla 3-10 Análisis de compatibilidad para la tarea: Ver historial	64
Tabla 3-11 Resultados del análisis de compatibilidad por tareas entre IE7 y MF2.0	65
Tabla 3-12 Resultados de las actividades para el usuario #1	67
Tabla 3-13 Resultados de las actividades para el usuario #2	68
Tabla 3-14 Resultados de las actividades para el usuario #3	69
Tabla 3-15 Resultados de las actividades para el usuario #4	70
Tabla 3-16 Resultados de las actividades para el usuario #5	71
Tabla 3-17 Resultados de la validación con usuarios.....	72
Tabla 3-18 Resultados del análisis de compatibilidad por tareas entre Acrobat Reader y KPDF.....	75
Tabla 3-19 Resultados del análisis de compatibilidad por tareas entre Winzip y 7-Zip	76

Tabla 3-20 Resultados del análisis de compatibilidad por tareas entre MSN Messenger y Alvaro's MSN..... 77

Tabla 3-21 Resultados del análisis de compatibilidad por tareas entre Microsoft Office Word y Open Office Writer..... 78

Tabla 3-22 Encuesta de efectividad de la técnica..... 79

Índice de Figuras

Figura 1-1 Actividades cognitivas de un usuario (Norman, 1986)	8
Figura 1-2 Curva necesidad-satisfacción de una tecnología (1998: The Invisible Computer) 19	
Figura 1-3 El cambio en los clientes según madura la tecnología (1998: La computadora invisible).....	21
Figura 2-1 Proyección de modelos mentales a partir de la proyección de modelos conceptuales	31
Figura 2-2 Función homologación	35
Figura 3-1 Metáfora de Interacción de Internet Explorer 7 y Mozilla Firefox 2.0.....	45
Figura 3-2 Barras de Internet Explorer 7.....	46
Figura 3-3 Barras de Mozilla Firefox 2.0.....	46
Figura 3-4 Barra de direcciones de Internet Explorer 7	47
Figura 3-5 Barra de herramientas de navegación de Mozilla Firefox 2.0	47
Figura 3-6 Barra de menús de Internet Explorer 7	48
Figura 3-7 Barra de menús de Mozilla Firefox 2.0	48
Figura 3-8 Barra de herramientas de Internet Explorer 7.....	48
Figura 3-9 Barra de pestañas de Mozilla Firefox F2.0.....	48
Figura 3-10 Barra de estado de Internet Explorer 7	51
Figura 3-11 Barra de estado de Mozilla Firefox 2.0	51
Figura 3-12 Resultados de la validación teórica vs. la práctica.....	73
Figura 3-13 Resultados de la Pregunta #1	80
Figura 3-14 Resultados de la Pregunta #2.....	80
Figura 3-15 Resultados de la Pregunta #3.....	81
Figura 3-16 Resultados de la Pregunta #4.....	81
Figura 3-17 Resultados de la Pregunta #5.....	82

Figura 3-18 Resultados de la Pregunta #6..... 82

Figura 3-19 Resultados de la Pregunta #7..... 83

Figura 3-20 Resultados de la Pregunta #8..... 84

Figura 3-21 Resultados de la Pregunta #9..... 84

Figura 3-22 Resultado de la Encuesta de Efectividad de la Técnica..... 85

Introducción

Desde los inicios de las ciencias de la computación, se ha tenido la necesidad de migrar de un sistema a otro, debido a la constante evolución que caracteriza la producción de software.

Inicialmente la comunidad de expertos informáticos desarrollaba el software que era usado por ellos mismos, de modo que la migración no representaba una dificultad para ellos. El software no era un producto de uso masivo.

Hoy por hoy, es creciente la comunidad de usuarios informáticos y las alternativas en materia de software son extensas, de modo que se puede encontrar una gama de productos de software que satisfagan los mismos requerimientos. Asimismo, las exigencias del mercado son cada vez mayores, los usuarios devengan calidad y funcionalidad ajustada a sus necesidades, y en esta búsqueda pueden migrar de un software a otro hasta encontrar el adecuado.

Así, las migraciones de software pueden darse por diversas razones, ya sea porque los sistemas han quedado obsoletos, o porque aparezcan necesidades que no cubran los sistemas actuales, o que el software coloque restricciones para su uso. También por simple deseo de cambiar y probar cosas nuevas o mantenerse actualizado con las últimas versiones y tendencias o a causa de decisiones organizacionales.

Sin embargo, el éxito de un proceso de migración depende de diversos factores, algunos relacionados con el diseño de las interfaces de usuario. Por otro lado, los aspectos humanos son primordiales, pues la forma en que las personas adquieren el conocimiento para utilizar un software es determinante en su decisión de adoptarlo o rechazarlo. Es por eso que la usabilidad juega un papel importante en este campo, pues si el software se diseña con el fin de ser usable e intuitivo para el usuario, las migraciones de sistemas serán más fáciles de desarrollar.

Una vez conocida la motivación y disposición por parte del usuario ante una migración, resultaría interesante saber si él asimilaría fácilmente la nueva plataforma, y qué tanto afectaría en su productividad un cambio en la forma de utilizar el nuevo software.

También, cabría preguntarse qué aspectos habría que modificar o alterar en la visión o modelo mental que ha adquirido el usuario con el uso arraigado de un software, para que no sea difícil y traumática la transición a otro sistema.

Para dar respuestas a estas inquietudes, se tiene como objetivo medir el impacto y la viabilidad de un proceso de migración de software, proponiendo una técnica basada en el estudio de los modelos cognitivos del usuario, que permita medir la asimilación tecnológica de éste, de forma sistemática y efectiva, con lo cual sea posible planificar de manera responsable cualquier proceso de migración sobre un escenario similar.

En atención a lo planteado, este Trabajo Especial de Grado está estructurado en tres capítulos. En el Capítulo 1 se expone de manera general todo lo relacionado a migración de software, desde la definición, pasando por esquemas de procesos de migración y los problemas que pueden presentarse en la transición. También se resalta la importancia de los aspectos humanos en este tema, lo cual incluye modelos mentales del usuario y su relación con los modelos conceptuales de sistemas. Asimismo se explican los tópicos de usabilidad que deben involucrarse en los procesos de migración de software. Se muestra cómo un producto de software alcanza su madurez y cómo actúan los usuarios en este proceso. Por último se presentan las técnicas y métricas de usabilidad que pueden ser de ayuda fundamental para lograr la aceptación del software en una migración.

En el Capítulo 2 se explica el problema abordado y el objetivo de este Trabajo Especial de Grado, así como se plantea una propuesta para hacer frente a un proceso de migración, de forma que se pueda predecir *a priori* su factibilidad siguiendo las mejores prácticas determinadas, con el fin de ahorrar costos innecesarios y llevar a buen término cualquier proceso similar.

Finalmente, el Capítulo 3 presenta la parte experimental, donde se aplica la técnica propuesta en el Capítulo 2 en varios casos de migración de software, así como su correspondiente análisis de resultados y conclusiones.

A lo largo de este documento, se usarán los términos *producto de software*, *herramienta de software*, *sistema*, *aplicación* y *software* con el mismo significado para referirse al producto que puede ser migrado. Así también se utilizará el término *interfaz o interfaces* para referirse a interfaces de usuario.

Capítulo 1: Marco Teórico

Este capítulo presenta los aspectos generales de una migración de software y la relación que tienen los aspectos humanos y la usabilidad con este proceso. Así pues, este capítulo se divide en tres secciones.

La sección 1.1 muestra el contexto para la migración de software, en este sentido, contiene la definición de migración tecnológica, presenta esquemas de migración y los problemas que puedan aparecer durante este proceso.

La sección 1.2 describe los aspectos humanos que deben tomarse en cuenta para la usabilidad del software en términos de metáforas y tareas, y destaca los conceptos *affordance*, *constraints* y *mapping* introducidos por Norman.

La sección 1.3 reúne los tópicos de usabilidad relacionados con un proceso de migración, de manera que, se define la usabilidad del software mostrando como éste puede alcanzar madurez, y se hace referencia a las técnicas y métricas aplicables para determinar si el software es usable.

1.1 Contexto para la migración de Software

El significado de una migración de software y la descripción de diferentes formas de implementarla, así como las recomendaciones e inconvenientes que surgen en ese proceso, son temas abordados en esta sección.

Este escenario ayuda a comprender que en un proceso de migración no puede darse sólo con la sustitución del software, pues están involucrados factores de preparación y previsión, que deben ser tomados en cuenta en el momento justo, para disminuir la posibilidad de fracaso.

1.1.1 Definición de Migración Tecnológica

Una migración tecnológica es un proceso de cambio en los elementos del software y/o hardware. Toda migración incluye una serie de pasos a seguir (Ministerio de Ciencia y Tecnología, 2005):

- Determinación de la causa de la migración (por qué)
- Fijar el momento y lapso de migración (cuándo)
- Fijar el procedimiento de migración (cómo)
- Evaluar la migración.

Un ejemplo de migración de software puede ser la transición de una plataforma a otra completamente distinta. Otro ejemplo, es el cambio de versión de un sistema operativo a una versión más actualizada (Netproject, 2003), incluso en este cambio dentro de un mismo vendedor no se debe suponer que los formatos de archivos, por ejemplo, serán transportables, por lo que habrá que hacer pruebas de funcionamiento antes de realizar cualquier modificación. Todas las migraciones deben basarse en una cuidadosa planificación.

Netproject (2003) recomienda entender el entorno final, junto con las diferentes opciones y posibilidades disponibles. Esto significa que hay que formar al personal, contratar personal o recurrir a consultores. Esto implica algunos costos iniciales y por ello es necesario disponer de respaldo de los responsables de la gestión.

Antes de tomar cualquier decisión, Netproject (2003) sugiere algunos aspectos que hay que considerar, como tener claro cuáles son las implicaciones del nuevo software. Cuando se tienen varias opciones, los responsables del proceso de migración deben conocer las ventajas y desventajas de cada producto. Consultar a los usuarios y explicarles las razones que hay detrás de la migración y cómo les afectará, estudiar sus preocupaciones con seriedad y permitirles que practiquen con la nueva tecnología sin pérdida de tiempo, cuanto antes se impliquen los usuarios será mejor. Supervisar la respuesta de los usuarios y tomar nota de los problemas que surjan.

1.1.2 Esquemas de Procesos de Migración

En cualquier proceso de migración de software se debe tomar en cuenta ciertas consideraciones que determinan el desarrollo de tal transición.

A continuación se presentan las siguientes opciones en cuanto a esquemas de migración según la velocidad del proceso de migración (Netproject, 2003):

a) Big bang

Todos los usuarios cambian del viejo sistema al nuevo de un día para otro. En la práctica, esto significa programar el cambio en un fin de semana o fiesta nacional.

La ventaja es que no se necesitan disposiciones de doble acceso y el personal no se va a encontrar yendo y viniendo de un sistema a otro. Entre las desventajas está el alto riesgo y la gran exigencia de recursos durante el cambio. Este esquema de migración sólo cabe en el caso de las pequeñas administraciones.

En todo caso, si es posible hay que evitar la migración de tipo big bang, porque puede ocasionar muchos inconvenientes que sean difíciles de controlar. Este esquema casi siempre falla debido a problemas de gestión.

b) Transición por fases en grupos

Se pasa a los usuarios del antiguo sistema al nuevo en grupos. Puede que los grupos funcionales completos se trasladen juntos para minimizar tener que compartir datos y los problemas de trabajo en el grupo. Se pueden contener los riesgos y gestionar los recursos eligiendo grupos del tamaño adecuado.

c) Transición de usuario a usuario

Básicamente se trata de la misma transición en grupos, pero con un grupo compuesto por una sola persona. Ese método de “goteo” tiene escasos requisitos en cuanto a los recursos, pero no resulta eficaz ni apropiado para grandes administraciones. Pero sí puede ser una buena manera de ejecutar proyectos piloto.

d) Introducir nuevas aplicaciones en un entorno familiar

Es probable que tanto los viejos como los nuevos sistemas tengan que funcionar “codo con codo” durante cierto tiempo. Así mismo, Netproject (2003) señala que es importante contar con una estrategia de transición que permita que ambos sistemas funcionen juntos, de manera que las actividades de producción se puedan continuar correctamente durante el período de transición.

Netproject (2003) propone los siguientes enfoques para iniciar los procesos de migración a partir de las actividades más fáciles e intuitivas, de forma que plantea:

- Período de transición

Muchas aplicaciones de software no están atadas a un sistema operativo y esto brinda la oportunidad de introducir estas aplicaciones sin tener que cambiar totalmente el entorno. Por ejemplo OpenOffice.org, Mozilla y Apache funcionan con Windows y pueden usarse en sustitución de Office, Internet Explorer e IIS (Internet Information Service) respectivamente.

Este enfoque permite que la reacción del usuario pueda ser calibrada a pequeña escala y que los planes para la formación de los usuarios puedan hacerse sobre la base de la experiencia real.

- Lo fácil primero

Los primeros cambios deben ser los que no afecten a la comunidad de usuarios, es decir, aquellos que se puedan hacer en el servidor y que proporcionen la plataforma para la posterior introducción de los demás cambios en el lado del cliente.

- Mirar hacia delante

Evitar hacer cosas que pueda dificultar la migración en el futuro. Por ejemplo:

- Insistir en que los desarrollos web produzcan un contenido que se pueda visualizar en todos los navegadores actuales de la web
- Insistir en el uso de formatos de archivo estándares
- Al escribir documentos en colaboración con otros, usar el formato que sea mínimo común denominador, esto aumentará la posibilidad de que las nuevas aplicaciones puedan participar
- Utilizar protocolos abiertos estándares. Desarrollar sistemas basados, por lo menos, en un modelo de tres niveles, donde el código de aplicación es independiente de la interfaz humana y de los métodos de acceso a los datos
- Evitar lenguajes y APIs de arquitecturas específicas.

1.1.3 Problemas durante el proceso

Netproject (2003) resalta la importancia de conocer la opinión del personal de la organización y mantenerlo informado acerca del proceso de migración, a la vez señala, que es vital ofrecer cursos de formación para capacitar a sus empleados. Aunque se tome en cuenta estas recomendaciones, habrá que hacer frente a reacciones típicas a los cambios en las prácticas laborales, estos son:

- a) El miedo a lo desconocido

Este miedo provoca que las personas se resistan al software, por representar algo nuevo para ellas. Sin embargo, al aplicar la transición por fases en grupos, algunos mostrarán más interés en aprender cosas nuevas y son ellos los que deberían probar primero el software, ya que podrían proporcionar los comentarios y sugerencias más útiles. Por eso se sugiere utilizar este grupo de usuarios en pruebas piloto y una vez que tengan cierta experiencia, puedan convencer y enseñar a sus colegas. En una segunda fase, los usuarios que pudieran ser más reservados necesitarán disponer de mayores facilidades de apoyo en forma de ventanillas de atención, *intranets* y usuarios locales con experiencia. También se recomienda que el personal de sistemas confíe y crea en el proceso de cambio para animar a los demás usuarios de manera positiva.

b) Saber es poder

La gente que conoce los sistemas y configuraciones existentes tiene un cierto poder, esto los haría sentirse bastante reacios a perderlo si el entorno nuevo es muy diferente del existente. Entonces aparece la necesidad de una gestión cuidadosa, ya que esas personas tienen un papel fundamental en el funcionamiento de los sistemas existentes. Quizá sea necesario que estén entre los primeros en recibir formación sobre los nuevos sistemas para que su posición en la entidad se mantenga.

1.2 Aspectos Humanos en el proceso de migración

El ser humano posee un sistema cognitivo compuesto por unos sistemas sensoriales encargados de extraer la información del ambiente. Esta información es analizada por los procesos perceptuales y almacenada en la memoria, para poder ser recuperada y utilizada posteriormente.

La memoria humana participa prácticamente en todos los actos de la interacción del humano con el computador y está compuesta por varios tipos de almacén: la memoria sensorial, la memoria a corto plazo y la memoria a largo plazo (Cañas, Salmerón & Gámez, 2001).

La memoria sensorial considera una serie de almacenes de información proveniente de los distintos sentidos que alargan la duración de un estímulo, aunque posee un tiempo de duración muy leve. Los almacenes más estudiados han sido los de los sentidos de la vista y el oído (Cañas, Salmerón & Gámez, 2001).

La memoria a corto plazo es un tipo de memoria que almacena cantidades limitadas de información por períodos breves. Si las unidades de información que se almacenan en este tipo de memoria no reciben un procesamiento, aproximadamente, entre quince y veinticinco segundos, desaparecerán. Un ejemplo simple puede ser buscar un número de teléfono en la libreta y mantenerlo en la memoria para utilizarlo por tan sólo unos pocos segundos mientras se hace la llamada, luego desaparece. (Poggioli Lisette, 1989)

La memoria a largo plazo guarda la información recibida a través de los sentidos la cual, al ser procesada en la memoria a corto plazo, puede ser transferida a este tipo de memoria. La memoria a largo plazo es ilimitada, no sólo en cuanto al período de duración de la información en ella, sino también en cuanto a su capacidad, es decir, la cantidad de unidades o grupos de información que se pueden almacenar.

En la memoria a largo plazo se almacenan los conceptos y las asociaciones o relaciones que existen entre ellos. Los conceptos pueden ser objetos percibidos (por ejemplo, tijera), rasgos de conceptos (por ejemplo, amarillo, rojo, cuadrado, redondo),

relaciones entre conceptos (por ejemplo, arriba, debajo, delante) o conceptos de alto nivel (por ejemplo, familia, sabiduría, paz) (Poggioli, 1989).

Norman (1986) ha señalado que un modelo de interacción puede servir para especificar cómo las variables psicológicas de las personas se relacionan con las variables de un sistema. Según este autor un usuario realiza siete actividades cuando interactúa con un sistema (ver Figura 1-1):

- 1) Establecer un objetivo
- 2) Formar una intención
- 3) Especificar las secuencias de acciones
- 4) Ejecutar la acción
- 5) Percibir el estado del sistema
- 6) Interpretar el estado
- 7) Evaluar el estado del sistema con respecto a los Objetivos y a las Intenciones.

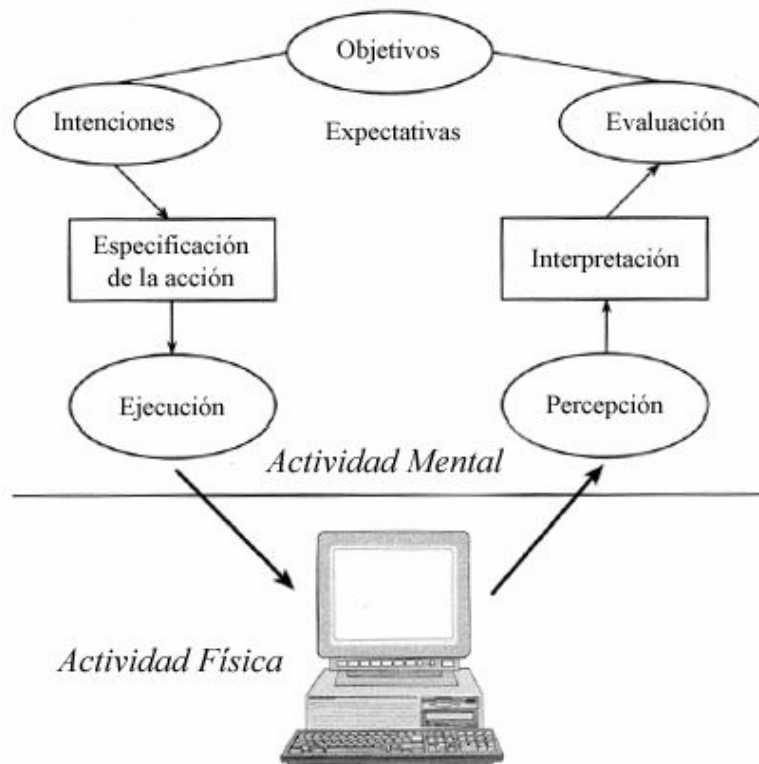


Figura 1-1 Actividades cognitivas de un usuario (Norman, 1986)

La interacción entre una persona y un computador ocurre precisamente cuando hay un intercambio de información entre ambos. El computador presenta cierta información en un formato físico determinado y la persona debe captarla a través de sus sentidos para después procesarla e interpretarla. De la misma manera, la persona transmite cierta información al computador que la capta a través de sus sistemas de entrada.

En el pasado los diseñadores de sistemas informáticos no habían dado importancia al elemento humano porque se suponía que sin mucho esfuerzo, los usuarios podían aprender y hacer uso de los sistemas y las aplicaciones desarrolladas. No obstante, la experiencia ha demostrado que el uso de los sistemas es muchas veces difícil, complicado y frustrante (Cañas, Salmerón & Gámez, 2001).

Los seres humanos están sujetos a pérdidas de concentración, cambios en el carácter, motivación y emociones, tienen prejuicios y miedos, cometen errores y faltas de juicio (Cañas, Salmerón & Gámez, 2001).

La dificultad de enfrentarse a nuevas situaciones está directamente relacionada al número de posibilidades. Los usuarios ven la situación e intentan descubrir cuáles partes pueden ser operadas y qué operaciones se pueden hacer (Norman, 1990).

Los problemas ocurren cuando hay más de una posibilidad. Si hay sólo una parte que puede ser operada y sólo una posible acción que realizar, no habrá dificultad (Norman, 1990).

Cuando encontramos un objeto nuevo, ¿cómo podemos saber qué hacer con él? Si nosotros tratamos con algo similar en el pasado, podemos transferir el conocimiento previo al nuevo objeto, a esto se la llama *mapping* (proyección). En estos casos, la información que necesitamos está en nuestras cabezas. Otra manera es el uso de la información del entorno, particularmente si el diseño del objeto nuevo se nos ha presentado con información que podamos interpretar (Norman, 1990).

“¿Cómo diseñar señales para las acciones apropiadas?, un conjunto importante de señales se manifiestan a través de *constraints* naturales de los objetos, éstas son las restricciones físicas que limitan lo que se puede hacer. Otro conjunto de señales proviene del *affordance* (apariencia de las cosas o comprensión intuitiva) de los objetos, lo cual permite descubrir sus posibles usos, acciones y funciones. *Affordance* puede indicar cómo mover un objeto, cuánto soportará, cuáles partes se mueven, y cuáles partes son fijas, etc. *Affordance* sugiere el rango de posibilidades, los *constraints* limitan el número de alternativas. El uso serio y consciente de *affordances* y *constraints* juntos en el diseño permite a los usuarios determinar rápidamente el curso de una acción, en cada situación nueva” (Norman, 1990).

Norman explica cómo reacciona el humano ante una situación nueva, tal como enfrentarse a un nuevo sistema o una interfaz diferente, utilizando todo el

conocimiento previo que ha adquirido a lo largo de su vida. Aunado a esto, la presencia de *affordances* y *constraints* influye en tales comportamientos; todos estos factores conllevan al *mapping* de conocimiento que hace que el humano esté predispuesto a actuar de una u otra forma.

Es importante dedicar un poco de tiempo, para intentar comprender los aspectos humanos de la informática y concretamente dentro de este aspecto el modelo cognitivo, porque puede ser muy útil para definir modelos de interfaces de usuario que se adapten más fácilmente a los modelos cognitivos del ser humano (Cañas, Salmerón & Gámez, 2001).

1.2.1 Modelo Mental del usuario

Durante el aprendizaje una persona adquiere conocimientos de las relaciones estructurales y el funcionamiento de los sistemas con los que interactúa a lo largo de su vida.

La estructuración de un modelo mental es parte de la naturaleza humana, pues el hombre necesita un modelo que le ayude a explicarse el mundo y a hacerlo en cierta forma predecible (Alonso, 1999).

El concepto de modelo mental es fácil de comprender si se piensa en ejemplos de la vida cotidiana. Cuando se interactúa con objetos del hogar se aplican modelos mentales previamente contruidos; por ejemplo, al encender la televisión se tiene un conocimiento de cómo funciona ésta, este conocimiento no tiene que ser perfecto ni amplio, pues si sólo se desea encenderla y ver un programa se necesita saber que funciona con corriente eléctrica y que hay un interruptor, que las imágenes llegan a través de la antena; sin embargo, un técnico que repara televisores necesita un conocimiento más profundo sobre su estructura y funcionamiento, es decir, que necesita tener un modelo mental más completo (Cañas, Salmerón & Gámez, 2001).

Un modelo mental es una representación de un estado de cosas del mundo exterior almacenada en la memoria de largo plazo. Se trata de una forma de representación de los conocimientos reconocida por numerosos investigadores en ciencias cognitivas por ser la manera natural por la cual la mente humana construye la realidad, concibe sus alternativas y verifica hipótesis cuando entra en un proceso de simulación mental. (Johnson Laird, 1983).

Un modelo mental según Norman (1983), tiene las siguientes características:

- Es incompleto
- Es ejecutable mentalmente, es decir, el usuario puede mentalmente simular su funcionamiento

- Es inestable en el sentido de que el usuario olvida sus detalles fácilmente
- No tiene unos límites claros y se confunde con los modelos mentales de sistemas físicos similares
- Es acientífico e incluye supersticiones y creencias erróneas sobre la conducta del sistema
- Es parsimonioso porque los usuarios prefieren reducir su complejidad.

Por consiguiente, el modelo mental no es ni completo, ni consistente, ni exacto, representa mas bien un conocimiento general, que hasta puede ser vago, acerca del sistema.

Tener un modelo mental del sistema no implica saber cómo el sistema funciona internamente. Un modelo mental contiene un conocimiento mínimo del funcionamiento interno.

Cuando un usuario se enfrenta por primera vez a un sistema, se forma un modelo mental para tratar de entender su funcionamiento. El modelo mental puede no ser acertado en principio, sin embargo, a través de la continua interacción con el sistema y mediante un proceso de aprendizaje, se va refinando para convertirse en un conocimiento específico. Este proceso hace que el modelo mental del usuario se vaya acercando a la definición del modelo conceptual del sistema.

1.2.2 Modelo Conceptual del Sistema

El modelo conceptual debe suministrar información al usuario acerca de qué hace el sistema y los mecanismos para llevarlo a cabo. Su importancia radica en que debe favorecer el aprendizaje del sistema, es una guía para predecir el comportamiento del sistema y de sistemas similares y, además, el usuario utilizará este modelo para establecer estrategias encaminadas a resolver sus problemas.

El modelo conceptual es la imagen del sistema tal como fue concebida por sus desarrolladores. En este sentido, un modelo conceptual puede verse representado por el Modelo de Casos de Uso (Pressman Roger, 2002) ya que éste agrupa las funcionalidades del sistema. Así también, la misma interfaz de usuario refleja un modelo conceptual.

Un modelo conceptual correcto permite al usuario predecir los efectos de sus actos (GRIHO a, 2005). Un usuario experto posee un modelo conceptual acertado, con lo que adquiere seguridad para efectuar tareas.

1.2.3 Tareas y Metáforas

Al utilizar un software, el usuario busca realizar una actividad específica para cumplir su meta planteada. A estas acciones individuales e identificables que tienen un objetivo específico sobre un software se les denominan tarea.

Las tareas en un sistema pueden ejecutarse siguiendo las opciones de un menú desplegable, a través del reconocimiento de íconos o utilizando los atajos de combinaciones de teclas conocidos como *shortcuts* o atajos.

Algunos ejemplos de tareas en un procesador de textos son acciones como: crear documento nuevo, abrir documento, copiar texto, pegar texto, alinear párrafo, insertar imagen, cerrar documento, entre otras.

Los productos de software que han sido diseñados para el mismo propósito, no sólo deberían compartir funcionalidades, sino también la forma de ejecutar las tareas.

Cuando se quiere comunicar un concepto abstracto de una manera más familiar y accesible se utiliza el recurso de las metáforas. En general, las metáforas pueden ser herramientas útiles para ayudar a los usuarios a iniciarse en el aprendizaje de las tareas de un nuevo sistema.

La metáfora visual es una imagen que permite representar alguna cosa y que el usuario puede reconocer lo que representa y, por extensión puede comprender el significado de la funcionalidad que recubre (Lorés, 2001).

Las metáforas pueden variar de pequeñas imágenes puestas en botones de barras hasta pantallas completas, es por eso que se suele referir a éstas como metáforas de interacción. Lo realmente importante de una metáfora es su silueta, pues al ser observada perdura por más tiempo que los detalles o colores de la misma, por tanto la silueta o forma externa de una metáfora transmite la mayor información.

Las personas entienden las metáforas por intuición. La intuición se define como: “Conocimiento claro, instantáneo, sin intervención de inferencia o razonamiento” (López, Olga L, 1996).

Una persona comprende el significado de las metáforas como los controles de una interfaz, al conectarlos mentalmente con otros procesos que previamente había aprendido.

Un ejemplo se evidencia con la metáfora del escritorio (Larés, 2001), que ha sido utilizada con éxito en muchas interfaces. En el momento de su creación, el uso del computador estaba dirigido a los oficinistas, por tanto esta metáfora correspondía exactamente al entorno en que la mayor parte de los usuarios trabajaban.

La base de la metáfora del escritorio consiste en crear objetos electrónicos que simulan los objetos físicos en una oficina, lo que incluye papel, carpetas, bandejas, archivadores.

La metáfora del escritorio al combinarse con otras metáforas permite que los usuarios puedan ejecutar un amplio abanico de tareas en el computador. Un ejemplo es la barra de desplazamiento, es una metáfora basada en la idea de un rollo, es decir como un papiro que se desplaza para leer.

La papelera es una herramienta habitual en la mayoría de las culturas avanzadas. Sirve para poner todos los papeles u otros elementos que no sirven, para después tirarlo como basura.

De este contexto, emerge la importancia de las metáforas como instrumentos que permiten recuperar rápidamente un modelo conceptual, que posibilita la comprensión de los objetos del sistema, en función de una experiencia pasada similar, lo cual conlleva a poder usarlos.

Al ser utilizadas en diversos productos de software, las metáforas deberían ser consistentes a nivel de aspecto y funcionalidad, para provocar en la mente del usuario los mismos modelos mentales, lo cual cooperaría de forma significativa en un proceso de migración. En la medida que el modelo mental del usuario se asemeja al modelo conceptual del sistema, el aprendizaje se facilita.

Si el usuario puede identificar las metáforas en un software y encontrar sus imágenes o equivalentes en el software al que desea migrar, para luego descubrir que el significado de ambas metáforas es similar, entonces no tiene que desechar su modelo mental previo, sino simplemente reutilizarlo.

Finalmente, se puede concluir, que el conocimiento adquirido por el usuario acerca de metáforas y tareas con un cierto sistema, constituyen las estructuras cognitivas necesarias para construir el modelo conceptual de otro software.

1.3 Usabilidad y el proceso de migración

Esta sección resalta la importancia que se debe dar a la usabilidad dentro de un proceso de migración de software, pues al fin y al cabo gran parte del éxito de esa transición se ve reflejado en la aceptación del software por parte de los usuarios finales.

Para muchos usuarios de sistemas de información, la frustración y la ansiedad forman parte de su vida diaria. Luchan por aprender el lenguaje de comandos y los sistemas de selección de menús que supuestamente les ayudan a realizar su trabajo. Algunas personas se encuentran con casos tan serios de *shocks* informáticos, terror en el

terminal o neurosis en la red, que evitan utilizar sistemas computarizados (Shneiderman, 1987).

Se puede argumentar que la interfaz de usuario es el elemento más importante de un sistema o producto interactivo basado en computadora. Si la interfaz tiene un diseño pobre, la capacidad que tiene el usuario de aprovecharse de la potencia de proceso de una aplicación se puede dificultar gravemente. En efecto, una interfaz débil puede llevar al fracaso de una aplicación con una implementación sólida y un buen diseño (Pressman, 2002)

Es importante que el software al que se desea migrar cumpla con las características idóneas de diseño para permitir la realización de una migración, pues de esta manera se obtendrá¹:

- Reducción de los costos de aprendizaje
- Disminución de los costos de asistencia y ayuda al usuario
- Optimización de los costos de diseño, rediseño y mantenimiento de los sitios
- Mejora la calidad de vida de los usuarios de la aplicación, ya que reduce su estrés, incrementa la satisfacción y la productividad.

Todos estos beneficios implican una reducción y optimización general de los costos de producción, así como un aumento en la productividad de los sistemas. La usabilidad permite mayor rapidez en la realización de tareas y reduce las pérdidas de tiempo.

Para lograr esto, durante la construcción del software se debe incorporar la usabilidad como una cualidad en el proceso de diseño y desarrollo del mismo, convirtiéndola en una medida de calidad que conlleve a apreciar la usabilidad como un atributo del producto final.

1.3.1 Definición de Usabilidad

La usabilidad de un software se refiere a la capacidad que posee éste de ser comprendido, aprendido, usado y atractivo para el usuario, en condiciones específicas de uso (ISO, 2000).

¹ Wikipedia (2006). *Beneficios de la Usabilidad*. Consultado el 06 de mayo de 2006 en <http://es.wikipedia.org/wiki/Usabilidad>.

Otro concepto define la usabilidad como la medida en la cual un producto puede ser usado por usuarios específicos para conseguir objetivos específicos con efectividad, eficiencia y satisfacción en un contexto de uso especificado (Sidar, 2000).

La efectividad a la que se hace referencia se aplica a la precisión y la plenitud con las que los usuarios alcanzan los objetivos especificados. Aunado a esto, se asocian la facilidad de aprendizaje, en la medida en que este sea lo más amplio y profundo posible y la tasa de errores del sistema, así como la facilidad de éste para ser recordado, es decir, que no se olviden las funcionalidades ni sus procedimientos.

La eficiencia involucra los recursos que han de emplearse para alcanzar la precisión y plenitud con que los usuarios logran los objetivos especificados; en este sentido, la facilidad de aprendizaje está relacionada con el costo de tiempo que implica y si se requiere acceso continuo a los mecanismos de ayuda del sistema.

La satisfacción se refiere a la ausencia de incomodidad y la actitud positiva en el uso del producto. Sin embargo, esto se trata de un factor subjetivo.

Finalmente la usabilidad depende básicamente de la rapidez y facilidad con que los usuarios llevan cabo las tareas de algún software, lo cual implica según Sidar (2000) lo siguiente:

- Una aproximación al usuario: Para desarrollar un producto usable, se debe contemplar la usabilidad enfocada en los usuarios, para esto se tiene que conocer, entender y trabajar con las personas que representan a los usuarios actuales o potenciales del producto
- Un amplio conocimiento del contexto de uso: Las personas utilizan los productos para incrementar su propia productividad. Un producto se considera fácil de aprender y usar en términos del tiempo que toma el usuario para llevar a cabo su objetivo, el número de pasos que tiene que realizar para ello, y el éxito que tiene en predecir la acción apropiada para llevar a cabo. Para desarrollar productos usables hay que entender los objetivos del usuario, hay que conocer los trabajos y tareas del usuario que el producto automatiza, modifica o embellece
- El producto ha de satisfacer las necesidades del usuario: El producto debe ofrecer la mejor relación de usabilidad, productividad y calidad
- Son los usuarios, y no los diseñadores y desarrolladores, quienes determinan cuándo un producto es fácil de usar.

El gran avance en la tecnología de los computadores ha incrementado la potencia de éstos, a la vez que ha ampliado la banda de comunicación entre el humano y el computador. Aun así, los principios aplicables al proceso de interacción son

independientes de la tecnología, puesto que dependen mucho más de un mejor conocimiento de los elementos humanos de la interacción que de la tecnología. (Dix, Finlay, Abowd & Beale, 1993).

Donald Norman (1999) asegura que el verdadero problema no radica en el énfasis de la propia tecnología, sino en la persona para la cual está hecho el dispositivo mientras el despliegue actual normalmente olvida al usuario final.

Nielsen (1994), por su parte, afirma que la importancia de la usabilidad en el desarrollo de software radica en que se trata de un factor crítico para que el sistema alcance su objetivo. Los usuarios deben tener la sensación real de que el sistema les ayudará a realizar sus tareas y éste debe hacerlo; de otra forma serán reacios a su utilización.

Nielsen (2005) propone diez principios de usabilidad para el diseño de interfaces de usuario:

- 1) Visibilidad del estatus del sistema: El sistema debe siempre mantener a los usuarios informados sobre lo que está pasando, a través de un apropiado *feedback* y en un tiempo razonable
- 2) Conexión entre el sistema y el mundo real: El sistema tiene que hablar el lenguaje de los usuarios, con palabras, frases y conceptos familiares a estos. Siguiendo las convenciones del mundo real, haciendo que la información aparezca en un lenguaje lógico y natural
- 3) Control y libertad por parte del usuario: Con frecuencia los usuarios eligen funciones del sistema por error y necesitan una salida de emergencia claramente marcada para dejar el estado indeseado sin tener que pasar por un diálogo extendido
- 4) Consistencia y estándar: Los usuarios no deben tener que preguntarse si las diversas palabras, situaciones o acciones significan la misma cosa
- 5) Prevención de errores: mejor que un buen mensaje de error es un diseño adecuado que prevenga que un problema aparezca por primera vez
- 6) Reconocimiento: hacer objetos, acciones y opciones visibles. El usuario no tiene que recordar información de una parte del diálogo en otra
- 7) Flexibilidad y eficiencia de uso: la posibilidad de la introducción de aceleradores en la interacción con el usuario, que serán normalmente utilizados por usuarios avanzados
- 8) Diseño estético y minimalista: de forma que un diálogo tiene que contener sólo la información que sea relevante y necesaria

- 9) Ayuda para reconocer, diagnosticar y recuperar al usuario de errores: los mensajes de error tienen que ser expresados en lenguaje natural, indicando el problema y sugiriendo una solución de forma constructiva. En definitiva tienen que ser mensajes que ayuden a entender al usuario qué es lo que está pasando
- 10) Ayuda y documentación: aunque lo mejor es que un sistema pueda ser utilizado sin documentación, puede ser necesario proveerlo de ayuda y documentación.

1.3.2 Usabilidad del Software

La Ingeniería de Usabilidad surge porque resulta imposible diseñar una interfaz de usuario óptima sólo con el mejor deseo de hacerlo. Los usuarios tienen un infinito potencial para hacer malas interpretaciones inesperadas de los elementos de una interfaz y para el diseño de sus propios trabajos, de forma muy distinta a lo que se pueda imaginar (Nielsen, 1994).

En la primera fase de todo proyecto de desarrollo de software es esencial tener una idea acerca de las características de los usuarios y de los aspectos del producto de mayor interés y necesidad (GRIHO b, 2005). El diseño puede mejorar si se trabaja pensando en la comprensión de los usuarios y sus tareas.

Teniendo en cuenta estas consideraciones de forma temprana se ahorra tiempo y dinero, dado que la posterior implementación de nuevos aspectos o nuevas interfaces de usuario implica un enorme esfuerzo adicional. Durante todo el desarrollo se deben realizar pruebas para comprobar que se está considerando la usabilidad del producto. Incluso una vez que el producto está en el mercado se debería preguntar a los usuarios acerca de sus necesidades y actitud respecto del mismo (GRIHO b, 2005).

Los expertos en usabilidad no deben avergonzarse de revisar un diseño de interfaz a partir del resultado de las pruebas de usuario, ya que esto puede representar una verdadera medida de la madurez de la usabilidad, el hecho de estar dispuesto a reconocer la necesidad de modificar las opciones iniciales del diseño para adaptarse a los usuarios. Nielsen afirma que se pueden ganar mercados si se admiten los errores de usabilidad del software como resultado de la aplicación de pruebas de usuario (Nielsen, 1994).

La usabilidad debería ser considerada en todo momento, desde el mismo comienzo del proceso de desarrollo hasta las últimas acciones antes de librar el sistema, producto o servicio a sus destinatarios (GRIHO b, 2005).

“Si el software es difícil de utilizar, si obliga a cometer errores, o causa frustración para conseguir los objetivos, no será de agrado, independientemente de la potencia informática que demuestre o de la funcionalidad que ofrezca. Dado que la interfaz es

la que da forma a la percepción del software por parte del usuario, tiene que estar bien diseñada” (Pressman, 2002).

“Una aplicación usable es la que permite que el usuario se concentre en su tarea y no en la aplicación” (Lorés, Granollers & Lana, 2002).

1.3.3 Hacia la Madurez del software

Michlmayr (2005) plantea la hipótesis de que existe una relación entre la madurez y el éxito del software. Este estudio trata de determinar si la madurez de un software particular influye en el éxito o fracaso del proyecto. Las experiencias obtenidas de este estudio se pueden aplicar para mejorar el proceso de desarrollo de software.

Después que el software se ha creado, el impacto del producto final puede ser medido o juzgado como cualquier otro producto del mercado. Esto es porque un software al ser insertado socialmente puede ser considerado como un producto que el comprador acepta, en principio, como algo que va a satisfacer sus necesidades.

En este sentido un software alcanza su madurez cuando es capaz de llegar a todos los estratos de la sociedad, y es aceptado y bien recibido por satisfacer las necesidades de sus usuarios, al punto que se hace cada vez más popular ofreciendo la mejor fiabilidad, rendimiento y seguridad.

Todas las tecnologías tienen un ciclo de vida, éstas progresan desde su nacimiento, a través de su adolescencia y hasta su madurez, en tales etapas sus características cambian. Durante este ciclo de vida, los tipos de clientes varían. Cada producto nuevo que surge presume de sus mejoras en la tecnología: rapidez, potencia, gran alcance. La tecnología toma el control, guiada por las características del mercado (Norman, 1998).

Cuando la tecnología madura, ya puede ser reconocida. Los clientes cambian, y desean diversas cosas del producto. La conveniencia y las experiencias de usuario dominan sobre la superioridad tecnológica.

Según Norman (1998), la fase de madurez de un producto tiene que estar dirigida por las necesidades del cliente, ahora se requiere un ciclo de desarrollo centrado en el humano. Como resultado las compañías deben cambiar: deben aprender a hacer productos para sus nuevos clientes, dejar que la tecnología sea servicial. Esto es una transición difícil para entender a una industria conducida por la tecnología, un cambio difícil de hacer. Es aquí donde la industria del computador personal está situada hoy. Los clientes desean el cambio, pero aún la industria vacila, se muestra indispuesta e incapaz de alterar sus maneras.

Si la tecnología de la información está para servir al consumidor medio, las empresas de tecnología deben estar dispuestas a cambiar y comenzar a examinar lo que hacen

los consumidores realmente, tienen que conocer las actividades de los que utilicen sus dispositivos. Pero esto requiere un cambio dramático en la actitud de los tecnólogos (Norman, 1998).

En sus primeros días, una tecnología no puede saber todo lo que necesita de sus clientes. Los primeros consumidores que necesitan la tecnología, están dispuestos a sufrir inconvenientes y altos costos para conseguirla. Mientras tanto, ellos siguen exigiendo la mejor tecnología, y el más alto funcionamiento. Con el tiempo, la tecnología madura, ofreciendo una mejor presentación, precios más bajos, y una alta confiabilidad. Estas fases son mostradas en la Figura 1-2; ahora hay un mayor cambio en el comportamiento de los clientes.

Cuando la tecnología alcanza el punto donde se satisfacen las necesidades básicas, las mejoras en la tecnología sobrepasan las expectativas del cliente, de manera que tales mejoras podrían cobrar menos interés. Por otro lado, la nueva clase de clientes sigue entrando en el mercado a medida que el producto madura.

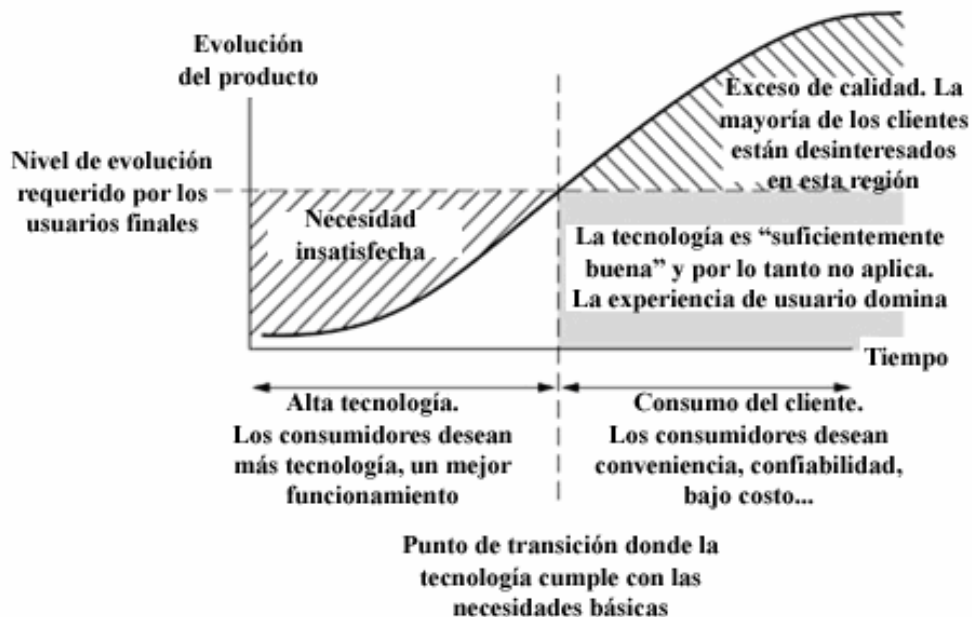


Figura 1-2 Curva necesidad-satisfacción de una tecnología (1998: The Invisible Computer)

Como puede verse en la Figura 1-2, las nuevas tecnologías comienzan a surgir desde el lado izquierdo del punto de transición al inicio de la curva, ofreciendo menos de lo que los clientes requieren. Pero los clientes demandan cada vez mejor tecnología y más funcionalidades, a pesar del costo. Sin embargo, ocurre una transición cuando la

tecnología alcanza el punto donde puede satisfacer las necesidades básicas del cliente. Este gráfico fue propuesto por Christensen (1997) (tomado de Norman, 1998).

Por muchos años, quienes estudian las formas en que las ideas innovadoras y productos entran a la sociedad han clasificado a las personas blancos de la innovación en cinco categorías: innovadores, adeptos tempranos, mayoría temprana, mayoría tardía, y rezagados. Cada uno juega un rol diferente en el desarrollo de una tecnología, con los adeptos innovadores y tempranos conduciendo la tecnología, y la mayoría temprana y tardía colocándose sobre las líneas laterales, esperando a que sea seguro saltar. Note, sin embargo, que en términos del tamaño creciente del mercado, son estos últimos clientes quienes dominan el término mayoritario. Estos clientes demandan conveniencia, facilidad de uso, confiabilidad, ellos quieren soluciones que simplifiquen sus vidas, no tecnologías que se las compliquen. Para Moore (1995), había un abismo entre estas dos clases de adeptos. A cada lado del abismo, una empresa debe tomar una actitud diferente hacia su mercado y hacia cómo concebir su producto.

Esta segmentación de la base de los clientes es altamente simplificada, pues el mercado mundial se compone de billones de personas, con una amplia variedad de intereses, habilidades, niveles socioeconómicos y educativos, y preocupaciones. Pero el determinar cuáles productos son necesidades y cuáles son lujos, depende de la cultura, el estilo de vida y un sin fin de otras variables, puesto que, las necesidades varían, el umbral que es bajo para unos, es alto para otros. Algunos individuos juegan múltiples roles en la sociedad: padre o hijo, estudiante o trabajador, empleado o encargado. Cada rol puede tener diferentes niveles de necesidades tecnológicas o mentir en las diferentes partes del ciclo de adopción.

En la Figura 1-3 se puede observar cómo en las fases tempranas se ubican los adeptos tempranos, quienes están dispuestos a apostarle a la nueva tecnología porque ellos sienten que las ventajas superan los costos. Los clientes más conservadores se contienen, esperando que la tecnología se demuestre por sí misma, o sea, madure y se convierta en confiable. Este es el ciclo de la adopción de mercado descrito por Geoffrey Moore (1995) en su libro *Crossing the Chasm* (Cruzando el Abismo) (Norman, 1998).



Figura 1-3 El cambio en los clientes según madura la tecnología (1998: La computadora invisible)

El punto de transición mostrado en la Figura 1-2 se ajusta a la brecha situada en la Figura 1-3. Los adeptos tardíos quieren conveniencia, bajos costos, una buena experiencia de usuario, esto es posible cuando la tecnología excede el punto de transición de la Figura 1-2 y esto corresponde a la ubicación de la brecha en la Figura 1-3 (tomado de Norman, 1998).

Una vez que la tecnología alcanza madurez, la naturaleza entera del producto cambia; tiene que ser diseñado, desarrollado y comercializado diferentemente (Norman, 1998).

También los usuarios tempranos son relativamente pocos en número, ellos conducen la tecnología. Estos son los clientes que adoran un desafío, quienes quieren estar en el borde inicial. Los entusiastas y visionarios de la tecnología ven la promesa ofrecida por las nuevas tecnologías y están dispuestos a tomar algunos riesgos, pues, en sus ojos, las ventajas compensan cualquier dificultad que se les presente (Norman, 1998).

La gran mayoría de las personas son pragmáticas y conservadoras. Ellos son los adeptos tardíos quienes tienen una vista más realista del mundo. Ellos no tienden a comprar nuevas tecnologías. Por el contrario, ellos miran y aprenden de la experiencia de los adeptos tempranos, esperan que las cosas se calmen, los precios bajen y la tecnología se estabilice, hasta que el producto sea verdaderamente capaz de satisfacer sus necesidades y ofrecerle confiabilidad y, beneficios reales sin interrumpir su manera de trabajar (Norman, 1998).

En el ciclo de aceptación del software al que se desea migrar, se puede apreciar los diversos tipos de usuario de forma análoga a los observados en la Figura 1-3. Los entusiastas y los adeptos tempranos aparecen como aquellos que han aceptado el reto y están dispuestos a probar y disfrutar las bondades que ofrece el nuevo software, hasta el punto de adoptar por completo su uso y comprobar que le es útil a sus propósitos. Estos se encuentran al lado izquierdo de la brecha que la Figura 1-3 resalta. Sin embargo, una gran mayoría de usuarios aún se mantiene del lado derecho de la brecha, esperando ver cómo resulta la experiencia de los adeptos tempranos con el nuevo software. Mientras tanto, estos adeptos tardíos que se ubican al otro lado de la brecha, aún utilizan el mismo software y no han podido saltar la brecha para definir la migración.

Las razones por las cuales los adeptos tardíos no se atreven a migrar constituyen la brecha, éstas son diversas, y ya se han indicado a lo largo de este documento. Incluyen, entre otras, el miedo a lo desconocido, la necesidad de aprender cosas nuevas, la inseguridad de no conocer el destino de la inversión, la falta de un modelo a seguir, el temor a que los conocimientos anteriores pierdan vigencia. En resumen, todo esto no es más que el temor al cambio. Ante esto, cabe preguntarse, cómo lograr que estos adeptos tardíos salten la brecha definitivamente.

Si al usuario se le presenta una barrera para enfrentarse al nuevo software, existe entonces un problema para el usuario final, de manera que resulta importante revisar el diseño de la interfaz del software y la fase del levantamiento de requerimientos funcionales, para conocer con toda propiedad lo que realmente necesitan los usuarios.

Lo ideal es que la brecha se desplace hacia la derecha, ofreciendo un panorama en el que el software nuevo abarque un gran mercado, dejando atrás a unos pocos rezagados. Este estado definiría el alcance de la madurez de la nueva opción de software.

1.3.4 Técnicas de Usabilidad

Para aminorar la brecha, las características del software que se desea cambiar deben ser proyectadas hacia el software elegido para la transición y así permitir una migración menos traumática, y provocar la aceptación del nuevo software por la gran mayoría de los adeptos.

Tal vez sea posible aplicar técnicas en el proceso de diseño y desarrollo de software que ayuden a reducir esa brecha. Valdría la pena preguntarse si existe alguna técnica de usabilidad que permita ayudar al proceso de migración de un software a otro.

Las técnicas de usabilidad pueden dividirse según James Hom (2003) en cuatro grupos: Indagación, Prototipaje y Categorización, Inspección y técnicas de *Test* (prueba).

1) Técnicas de Indagación

Los métodos de indagación se definen como aquellos en los que las fuentes de información básicas para dar comienzo al proceso de diseño se encuentran en el entorno del usuario. Sin embargo, existe una serie de distinciones y se admite la consideración de otras opciones, que han de incluirse en esta categoría. Estas son (Sidar a, 2000):

a) Aproximación Contextual

Se distinguen dos variantes claramente diferenciadas:

Una aproximación contextual, se trata de un método estructurado de entrevista de campo caracterizado por la necesidad de comprender el contexto, de asimilar al usuario en el proceso de diseño y de plantear un objetivo en su aplicación.

Una aproximación etnográfica, consiste en la observación del usuario y su interacción con el producto en su entorno habitual, prescindiendo de las ventajas del laboratorio a la hora de captar y registrar datos.

b) Aproximación por Grupos

En esta técnica los integrantes van a ser usuarios representativos del producto sometido a estudio y parte del contexto, durante la sesión no se encuentran en este contexto, aunque se permite preguntarles a los usuarios acerca de sus experiencias e impresiones respecto de un producto. Estos son eventos formales y estructurados que van a proporcionar datos y generar ideas.

Uno de los métodos más conocidos es el de grupos orientados, aquí el papel del moderador es fundamental y su proceder es determinante para el éxito de la sesión. Otro método es el de grupos de debate, sin embargo, el moderador ya no tiene la misión de estimular y guiar la discusión sino que conduce, establece y propone los temas a tratar en las sesiones.

c) Aproximación Individual

En esta técnica, el factor más importante, es la formulación de preguntas efectivas. Las metodologías más habituales son:

- Las encuestas: son interactivas, pero no poseen un carácter estructurado ni se establecen ni organizan formalmente.
- Los cuestionarios: es característico el formato de lista de las preguntas, así como el requerimiento de un esfuerzo adicional por parte del usuario, quien contesta y envía de vuelta el cuestionario al evaluador.

- Las entrevistas: de forma similar a los grupos orientados, rige la filosofía estímulo-respuesta (Stimulus & Response). De hecho, es frecuente ver unidas a ambas técnicas en algunas recopilaciones de herramientas por este motivo.

d) Participación Remota

El Cuestionario o Encuesta Remota es la versión a distancia de las aproximaciones individuales, con la misma dificultad que presenta la formulación de preguntas efectivas, pero con la ventaja que puede suponer el retorno de la información a través de la red.

También se habla de sistemas automatizados de captación y reunión de datos. Las técnicas más habituales son las Sesiones Guiadas (Journaled Sessions) y las Fotografías de Pantalla (Screen Snapshots) y, por lo general, están destinadas a la evaluación o *test* de software. En ambos casos se proporciona un prototipo del producto que se desea someter a prueba y un mecanismo para el registro de datos.

e) Generación de Ideas

La generación de ideas es una forma de descubrir, aprender y refinar determinados conceptos de diseño. La participación de los usuarios en estos procedimientos contribuye a ampliar perspectivas y a profundizar en una variedad de consideraciones que, en ocasiones, pasan inadvertidas para el diseñador.

f) Métodos de Observación Experta

Se trata del análisis, recopilación e investigación de una serie de tendencias combinado con técnicas de generación de ideas por parte de expertos. Puede verse como un enfoque demasiado comercial (son recursos propios del *Marketing*) en donde la información acerca del consumidor actual del producto sometido a estudio puede constituir una valiosísima fuente de datos que puede plantear objetivos de *test*.

2) Prototipaje y Categorización

Los métodos de Prototipaje y Categorización están orientados hacia la construcción del software, de manera que, luego de tener una idea clara de lo que se requiere, se proceda a realizar un boceto o versión previa del producto que permita o ayude a construir la interfaz final (Sidar a, 2000).

a) Prototipaje

El prototipaje modela el producto final y permite efectuar inspección y pruebas sobre determinados atributos del mismo sin necesidad de que esté disponible, para simplemente controlar haciendo uso del modelo. En muchas ocasiones se dirá que cuanto más próximo se encuentre el prototipo al producto real, mejor será la evaluación (Sidar a, 2000).

En cuanto a la fidelidad de la reproducción de la interfaz se tiene: el prototipaje de Alta Fidelidad donde el prototipo es prácticamente idéntico al producto final, y el prototipaje de Baja Fidelidad donde el aspecto del prototipo reproduce la disposición de sus características, dimensiones y otros aspectos, pero no se corresponde con el prototipo del producto final.

b) Categorización

Los métodos de categorización, conectados con el amplio y complejo mundo de la calidad, recuerdan la necesidad de una planificación meticulosa que obliga a referirse muchas veces a etapas de pre-diseño. La categorización sugiere cómo establecer el contenido de los menús de un software (Sidar a, 2000).

Un método famoso de categorización es la Categorización por Tarjetas (Sorting Card), en el cual los usuarios clasifican tarjetas, en las que se representan varios conceptos, en diversas categorías.

3) Técnicas de Inspección

a) Inspecciones

En el desarrollo de software el objetivo de una inspección es encontrar defectos. Las inspecciones de usabilidad tienen por objetivo descubrir defectos de usabilidad. Se trata de un proceso sistemático, frente a la evaluación heurística que no tiene un proceso establecido (Nielsen, 1994).

Hay dos tipos de inspecciones de usabilidad:

- Inspecciones de conformidad (con una guía de estilo o un estándar): los participantes inspeccionan el diseño de la interacción del sistema para comprobar si cumple lo estipulado en determinados estándares o guías de estilo. Todos los participantes deben estar familiarizados con los estándares o guías sobre los que se evalúa.
- Inspecciones de consistencia: el objetivo es identificar inconsistencias entre contextos de interacción y sus contenidos. El evaluador intenta encontrar inconsistencias en terminología, color, disposición, formatos

de entrada y salida de datos, etc. Cuando el producto pertenece a una familia de productos, se juntan representantes de los distintos equipos de diseñadores para inspeccionar la consistencia entre los productos de la familia.

b) Evaluación Heurística

La evaluación heurística se presenta como una variante de la Inspección de Usabilidad y se trata de un método de ingeniería para encontrar los problemas de usabilidad en los diseños de interfaces, utilizado para poder atender tales problemas como parte de un proceso iterativo del diseño (Nielsen y Molich, 1990; Nielsen 1994). La evaluación heurística implica tener de un conjunto de evaluadores que examina la interfaz y juzga su conformidad con los principios reconocidos de la usabilidad (Nielsen, 1994).

La evaluación heurística de una aplicación está basada, entonces, en la observación por parte de un experto en usabilidad o en interfaces de usuario, de ciertos parámetros o guías generales. Entre otros, podemos citar la coherencia en la presentación de la información, la visualización y coherencia de las acciones e interacción con el sistema, los métodos de entrada/salida de información, el respeto por la ergonomía y los factores humanos (Mercovich, 2000).

La evaluación heurística (Nielsen, 1994) es generalmente difícil de realizar por un solo individuo, porque una persona nunca podrá encontrar todos los problemas de usabilidad de una interfaz. Afortunadamente, la experiencia de muchos proyectos ha demostrado que diferentes personas encuentran distintos problemas de usabilidad. Por lo tanto, es posible mejorar la eficacia del método perceptiblemente implicando a múltiples evaluadores.

c) Recorridos Cognitivos

Los recorridos cognitivos (Sidar a, 2000) constituyen una técnica de revisión donde los evaluadores expertos construyen escenarios para las tareas a partir de una especificación o de un prototipo temprano para desempeñar después el papel del usuario trabajando con la interfaz en cuestión (haciendo un recorrido a través de la misma). Actúan como si ellos estuvieran en el papel del usuario tipo, trabajando a través de las tareas que realizan.

El recorrido cognitivo (Lorés, Granollers & Perdrix , 2004) requiere enumerar las tareas concretas a desarrollar durante la evaluación. Implementar para cada tarea la lista completa de acciones necesarias para ser completada en el software estudiado. Esta lista consta de una serie repetitiva de pares de acciones (del usuario) y respuestas (del sistema) que llevan a la consecución de la misma.

Recorrer las acciones: los evaluadores deben realizar cada una de las tareas determinadas anteriormente siguiendo los pasos especificados y utilizando el prototipo descrito. En este proceso el evaluador utilizará la información del factor cognitivo (experiencia y conocimiento adquirido) de los usuarios para comprobar si la interfaz es adecuada para los mismos. Esta revisión ha de ser minuciosa para todas las acciones especificadas como necesarias para la consecución de la tarea.

d) Listas de Comprobación

Las técnicas más características sobre Listas de Comprobación son las siguientes (Sidar a, 2000):

- Guías de Comprobación: las guías y las listas de comprobación ayudan a asegurar que los principios de usabilidad sean considerados en un diseño. Normalmente, las listas de comprobación se utilizan en combinación con algún método de inspección de usabilidad y sirven de referencia
- Listas de Comprobación Basadas en Escenarios: se trata de una particularización de la anterior en la que la inspección se lleva a cabo a través de tres escenarios: usuario novel, usuario experto y manejo de errores. Para cada uno se deberá proporcionar una lista de aspectos a comprobar.

4) Técnicas de *Test* (Prueba)

Un *test* de usabilidad es una medida empírica de la usabilidad de una herramienta, sitio o aplicación, tomada a partir de la observación sistemática de usuarios llevando a cabo tareas reales (Mercovich a, 2000).

El *test* de usabilidad, permite:

- Verificar la existencia de posibles problemas de usabilidad en el sitio
- Encontrar posibles soluciones para los problemas detectados
- Establecer una medida concreta inicial contra la cual comparar a los competidores, futuros desarrollos de este mismo sitio o modificaciones al actual.

En la mayoría de las técnicas de *test* se contempla la captación de las impresiones y sensaciones que el usuario manifiesta oralmente, con la aplicación de protocolos de expresión de usuario, donde se distinguen (Sidar a, 2000):

- Protocolo del Pensamiento Manifestado (Thinking Aloud Protocol): durante el transcurso del *test*, donde el participante está realizando una tarea como parte de un escenario de usuario, se solicita de éste que exprese en voz alta sus pensamientos, sensaciones y opiniones mientras interactúa con el producto.
- Protocolo de Preguntas (Question-Asking Protocol): este método lleva un paso más allá al protocolo del pensamiento manifestado al provocar las manifestaciones del usuario respecto del producto mediante la formulación de preguntas directas acerca del mismo.

1.3.5 Métricas de Usabilidad

Una métrica (Cueva J.M) es un valor numérico o nominal asignado a características o atributos de un ente, obtenido por medio de la correspondencia de un dominio empírico (mundo real) a un mundo formal (matemático).

Generalmente, la usabilidad es medida en relación al rendimiento de los usuarios sobre un conjunto dado de tareas de prueba. Las medidas se basan en la definición de la usabilidad como calidad métrica (Nielsen, 2001):

- Tasa de éxito (si los usuarios pueden realizar todas las tareas): para recolectar métricas, Nielsen recomienda usar la medida más simple de usabilidad, la tasa de éxito del usuario, definiendo esta tasa como el porcentaje de las tareas que los usuarios completan correctamente.

En la Tabla 1-1, Nielsen (2001) muestra datos del éxito de la tarea de un estudio determinado por él, donde se solicitó a cuatro usuarios que realizaran seis (6) tareas.

	Tarea 1	Tarea 2	Tarea 3	Tarea 4	Tarea 5	Tarea 6
Usuario 1	F	F	E	F	F	E
Usuario 2	F	F	P	F	P	F
Usuario 3	E	F	E	E	P	E
Usuario 4	E	F	E	F	P	E

Nota: E = éxito, F = falla, P = éxito parcial

Tabla 1-1 Tasa de éxitos del usuario propuesta por Nielsen

En total, se observaron 24 intentos de realizar las tareas. De esos intentos, 9 fueron acertados y 4 fueron parcialmente acertados. A las tareas logradas con éxito parcial se les otorga medio punto (0.5).

En este ejemplo, la tasa de éxito es la siguiente: $(9 + (4 * 0.5)) / 24 = 46\%$.

Así, la tasa de éxito puede obtenerse según la siguiente fórmula:

$$\text{Éxito} = \frac{N^{\circ} \text{Tareas Éxitosas} + (N^{\circ} \text{Tareas Parcialmente Completadas} * 0.5)}{N^{\circ} \text{Total De Tareas}}$$

- El tiempo que una tarea requiere: consiste en medir el tiempo que le toma a un usuario realizar una tarea dada.
- Tasa de error: resultado del complemento de la tasa de éxito.

$$\text{Error} = \frac{N^{\circ} \text{Tareas Fallidas} + (N^{\circ} \text{Tareas Parcialmente Completadas} * 0.5)}{N^{\circ} \text{Total De Tareas}}$$

- Satisfacción subjetiva de los usuarios: medido generalmente por las respuestas de los usuarios a encuestas de satisfacción.

Estas métricas se pueden aplicar tanto a usuarios novatos y usuarios experimentados.

Todo lo referenciado en este capítulo sirve de base para entender el contexto de un proceso de migración, así como las razones de los problemas que se puedan presentar. Para solventarlos, es fundamental involucrar los aspectos humanos cuando se desea migrar de una herramienta a otra.

Entre las técnicas aquí citadas se presentan métodos para verificar la usabilidad del software, así como formas para explorarlo y medirlo. Aplicar ciertas técnicas a un software en particular, mientras se tiene referencia de otro, puede ayudar a descubrir las similitudes y desbalances entre ambos productos. De forma que sea posible identificar los focos donde se le presenten problemas al usuario, así como aquellos en los que éste se sienta a gusto. Posteriormente, los resultados de las técnicas deben ser tomados en cuenta en un replanteamiento de la selección o del desarrollo del software al que se desea migrar, con el objeto de reutilizar los conocimientos previos que tiene el usuario final para la comprensión del nuevo producto.

Capítulo 2: Marco Metodológico

En este capítulo se presenta el problema que será afrontado, así como el objetivo que se desea alcanzar, y se desarrollan la propuesta y las bases para hacer frente a un proceso de migración.

2.1 Problema

Basándose en los aspectos tratados a lo largo del documento, este Trabajo Especial de Grado aborda el problema de predecir qué tan fácil o difícil puede ser un proceso de migración de software cuando los usuarios están arraigados o acostumbrados al uso de una herramienta y sólo conocen el funcionamiento de la misma.

2.2 Objetivo

El objetivo de esta propuesta es medir el impacto y la viabilidad de un proceso de migración de software, siguiendo una técnica basada en el estudio de los modelos cognitivos del usuario (modelos mentales), que permita medir la asimilación tecnológica de éste, de forma sistemática y efectiva, con lo cual sea posible planificar de manera responsable cualquier proceso de migración sobre un escenario similar.

2.3 Técnica de Migración de Software

La técnica propuesta está basada en aprovechar la afinidad entre las herramientas de software involucradas en el proceso de migración. Esto es, que los modelos conceptuales de tales herramientas puedan ser proyectados entre sí, es decir, que cada funcionalidad, tarea y metáfora de interacción del software origen tenga su correspondiente imagen en el software destino. Así pues, si es posible establecer una función inyectiva² o biyectiva³ entre ambos sistemas, se puede decir que estas herramientas son homólogas y no debería representar un problema el uso de una u otra. Finalmente, si existe una correspondencia entre los modelos conceptuales de sistemas homólogos, ambos sistemas pueden producir un mismo modelo mental en el usuario, de forma que el modelo mental que se tiene acerca de un software pueda ser

² Inyectiva: Sea f una aplicación de A en B . Entonces f se dice inyectiva si elementos distintos de B corresponden a elementos distintos de A , es decir, si dos elementos de A tienen imágenes distintas. (Lipschutz S, 1991).

³ Una función biyectiva es aquella donde cada elemento del conjunto de llegada es imagen de un solo elemento del conjunto de partida. (Figuera Y, 1993).

proyectado (*mapping*⁴) a otro (Ver Figura 2-1), haciendo posible predecir los resultados del proceso de migración.

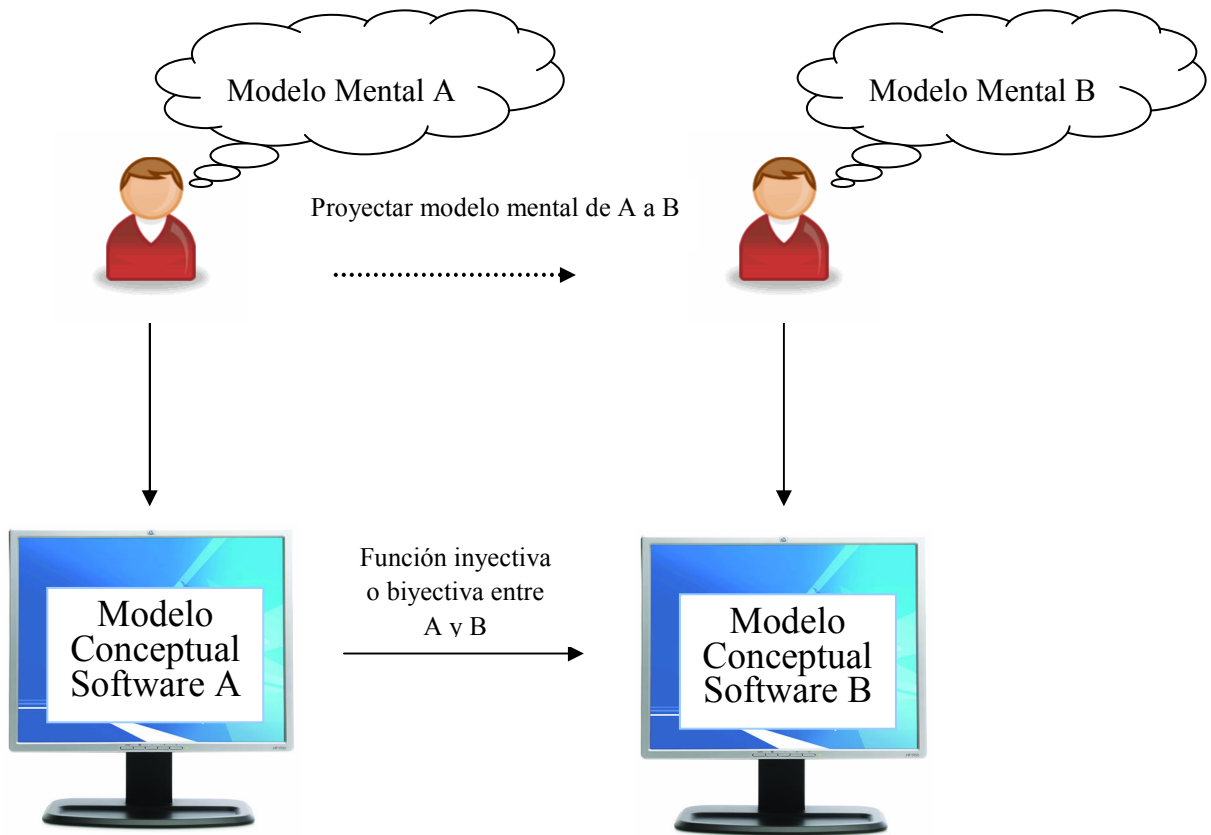


Figura 2-1 Proyección de modelos mentales a partir de la proyección de modelos conceptuales

Se plantea una estrategia de comparación para descubrir analogías entre modelos mentales y conceptuales, para lo cual es necesario identificarlos y descubrir sus características a través de técnicas de usabilidad.

La estrategia propuesta puede aplicarse a un escenario de usuarios reales, para comprobar que, si el modelo conceptual de un software puede ser proyectado hacia el modelo conceptual de otro, también es posible proyectar el modelo mental que tiene un usuario de un software a otro.

Hay que considerar que la comparación de modelos conceptuales debe llevarse a cabo entre herramientas de software diseñadas para un mismo propósito y los usuarios no

⁴ Entiéndase *Mapping* como una regla de correspondencia establecida entre conjuntos, donde se asocia cada elemento de un conjunto con otro elemento del mismo u otro conjunto (Answer, 2007).

deben ser novatos, pues ellos no tienen ningún conocimiento previo que aprovechar en una migración, en ese caso sería aplicable un entrenamiento que les enseñe a utilizar el software.

Es importante destacar entonces, que para comparar modelos conceptuales, se requiere identificarlos a través de la metáfora de interacción y un conjunto de tareas propias del software, que deben estar presentes tanto en el software A (software origen) como en el software B (software destino).

Si se tiene un conjunto de tareas en A que no tienen su correspondiente representación en B la migración simplemente no tiene sentido, pues en ese caso no estarían satisfechas las necesidades de los usuarios. Por eso es necesario que cada tarea de A pueda ejecutarse en B, sin embargo no es problema que B posea mayor número de funcionalidades que A.

Si los elementos del conjunto de tareas en A están presentes en B, pueden presentarse dos escenarios. Que la forma de realizar las tareas en B no sea análoga a la manera de realizarla en A, es decir, que se tengan que desaprender o deshacer los conocimientos adquiridos en A para ejecutar la misma tarea en B, o lo que es lo mismo, que se requiera un costo de re-aprendizaje para utilizar el software B. Y que la manera de realizar una tarea en A sea análoga a la forma de realizarla en B, sin representar una mayor carga cognitiva para llevarla a cabo; esto representaría el mejor caso.

A continuación se explica lo que debe ser tomado en cuenta para llevar a cabo un proceso de migración de forma sistemática. A modo de resumen, al final de la técnica se ofrece una guía práctica que focaliza los pasos a seguir.

2.3.1 Extraer el modelo mental del usuario a través del modelo conceptual del software origen

Es necesario hacer una exploración detallada del software origen para obtener su modelo conceptual, y a través de éste descubrir el modelo mental que pueda tener un usuario acerca del software. Este estudio debe ser llevado a cabo por evaluadores expertos que apliquen técnicas de usabilidad para tal fin.

Se supone que el usuario domina un conjunto de funcionalidades del software para llevar a cabo su trabajo, de manera que a él le resulta familiar la estructuración de la imagen del sistema, los íconos, la forma en que se combinan tales funcionalidades para conformar las tareas representativas que le permiten ser productivo. Debe quedar claro que el usuario no necesariamente tiene que conocer todo el software, se supone que él ejecuta un subconjunto de las funcionalidades del software, que le permiten llevar a cabo los objetivos que se plantea durante su interacción con el sistema. Todo esto representa su modelo mental, el cual está almacenado en su memoria de largo plazo y es recuperado cuando se le presenta algún estímulo (imagen o ícono) para traerlo a su memoria de corto plazo y comenzar a interactuar siguiendo el modelo de

la tarea planteado por Norman (Norman, 1986). Basándose en lo anterior y para el contexto de este trabajo se plantea la siguiente fórmula:

$$\text{Modelo Mental del Usuario} = \text{MetáforaDe Interacción}(\text{SoftwareOr igen}) + \text{TareasRepresentativas}(\text{SoftwareOr igen})$$

Para obtener este modelo mental, es necesario aplicar algunas técnicas de usabilidad como lo son: recorridos cognitivos, entrevista con usuarios expertos y observaciones de campo (Sidar, 2000).

El perfil de los usuarios del software A considera que sean expertos en la interacción con el mismo, es decir, que puedan ejecutar tareas con éxito y eficacia en A, no necesariamente deben conocer todas las funcionalidades de A. Se solicitan usuarios expertos porque ellos ya tienen suficientes conocimientos sobre el uso de A y un modelo mental respectivo, a diferencia de los usuarios novatos que no poseen conocimiento previo sobre el software.

Un conjunto T de tareas representativas viene dado por $T = \{t_1, t_2, t_3, \dots\}$ donde t_i es una tarea que el usuario domina. El conjunto T representa todo el conocimiento que tiene el usuario experto para ser productivo en su ámbito de trabajo. A los usuarios seleccionados se les puede aplicar una encuesta que revele cuáles son las tareas que usa cada uno de ellos en el software A, éstas pueden ser también determinadas por el evaluador, para luego contabilizar aquellas tareas usadas con mayor frecuencia, a las cuales se les denominará tareas representativas.

La metáfora de interacción está representada por la distribución de los elementos en el entorno de trabajo y las técnicas de interacción para usar el software.

Las tareas representativas vienen dadas por el análisis de aquellas tareas que ya conocen los usuarios y realizan de forma experta, con esto se conocerán los pasos o maneras de ejecutarlas, así como también se contempla la observación de las siluetas de íconos, y su coherencia con relación a las funciones asociadas. El estudio de las tareas representativas y las metáforas de interacción es lo que determina el modelo conceptual de A.

Si el usuario es experto en el uso del software A, entonces, el modelo mental que tiene el usuario acerca de A es subconjunto del modelo conceptual de A, tal que,

$$U \text{ es experto en } A \Rightarrow MMU(A) \subseteq MC(A),$$

donde,

$U = \text{usuario}$

$A = \text{software origen}$

$MMU(A) = \text{Modelo mental que posee el usuario sobre el software origen}$

$MC(A) = \text{Modelo conceptual del software origen}$

2.3.2 Proyectar el modelo conceptual del software origen hacia el modelo conceptual del software destino

A través del recorrido cognitivo se construye el modelo conceptual del software B a partir del software A. Esto consiste en ubicar en B las tareas representativas de A.

Se debe buscar la relación existente entre los modelos conceptuales del software A y el software B para proyectar el modelo de A hacia el de B, estimando que cada tarea de A tenga su correspondiente imagen en B. Esto es que cada tarea que se ejecuta en A pueda llevarse a cabo también en B.

Tomando entonces el conjunto de tareas representativas que se tienen del software A, se establece una función de correspondencia entre A y B, tal que:

$$h : A \longrightarrow B$$

La función homologación h define que una tarea de A pueda ser homologada en B.

$$\forall t_{Ai} \in A \quad \exists t_{Bi} \in B \quad | \quad (t_{Ai}, t_{Bi}) \in h$$

Esto significa que a cada tarea t_{Ai} de A, le corresponde por h una tarea t_{Bi} , y al menos una, de B, que se denomina imagen de t_{Ai} por h y que se denota $h(t_{Ai}) = t_{Bi}$.

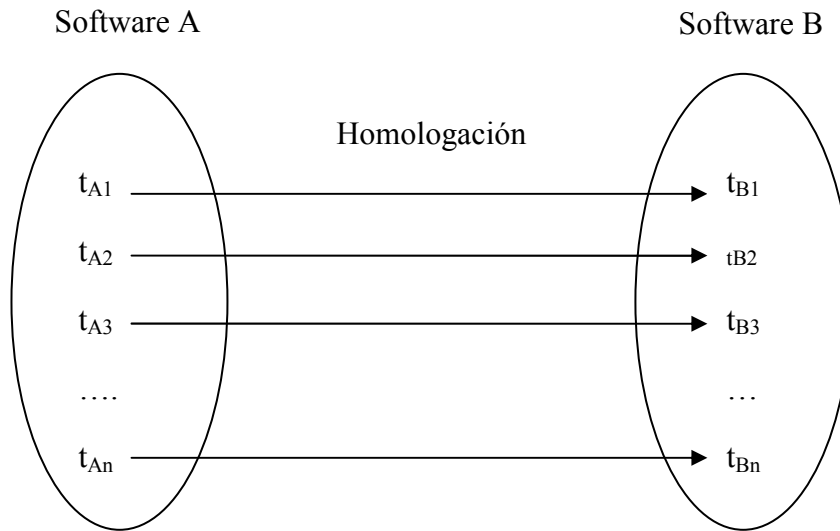


Figura 2-2 Función homologación

Si es posible proyectar el modelo conceptual de A ($MC(A)$) al modelo conceptual de B ($MC(B)$), y el usuario posee un modelo mental de A ($MMU(A)$) que es igual al $MC(A)$, entonces este $MMU(A)$ podría ser proyectado hacia el $MMU(B)$.

Es importante aclarar que si existen tareas en A que no poseen homólogos en B no se cumpliría la función propuesta, de manera que no es posible proyectar el $MC(A)$ al $MC(B)$, por lo tanto, el $MMU(A)$ no podrá ser proyectado en su totalidad hacia el $MMU(B)$, y por ende la migración sería un proceso engorroso.

2.3.3 Analizar la compatibilidad entre los productos de software

La base del análisis de compatibilidad es determinar desde el punto de vista ontológico si el software A puede ser utilizado como "ancla conceptual" para aprender efectivamente el software B, si se demuestra que existe la compatibilidad, el aprendizaje del software B se hará de forma significativa. Cuando se hace la referencia ontológica se está introduciendo los conceptos de familiaridad y transporte, el primero permite traer de la memoria de largo plazo a la de corto plazo, elementos que pueden ser útiles para el aprendizaje, y el segundo concepto permite transportar elementos que deben ser re-aprendidos en el nuevo software.

Un ejemplo de familiaridad se evidencia en la metáfora del escritorio, donde se encuentran elementos propios de la vida real como carpetas, reloj, papeleras que resultan familiares y reconocibles al usuario. El concepto de transporte, como se ha

indicado, implica destruir un conocimiento para aprender uno nuevo, esto se puede observar en la funcionalidad que le asignan los procesadores de texto (como *Microsoft Office*® y *OpenOffice*®) al ícono representado por una brocha, la cual es completamente distinta a la función de una brocha en la vida cotidiana, pues la brocha es usada para pintar en el mundo real, pero en el procesador de texto se utiliza para la función específica de “*Copiar formato*”.

Una vez identificadas las tareas de A en B, se realiza un estudio para descubrir qué tan compatible es la forma de ejecutarlas. Esto indicará la compatibilidad del software A respecto al B.

La compatibilidad va a depender del cumplimiento de los siguientes criterios:

- Que la metáfora de interacción de A sea similar a la de B. Se debe considerar la misma distribución del espacio de trabajo y el uso de las mismas técnicas de interacción para soportar la familiaridad y el transporte entre las dos metáforas de interacción.
- Que t_{Ai} se identifique con el mismo nombre que t_{Bi} : Si existe una compatibilidad nominal entonces aumenta la compatibilidad, pero si no la tiene entonces los cambios en el modelo mental del usuario pueden afectar el aprendizaje del software B, puesto que implica un doble proceso desaprender-aprender. Esto lleva a definir una estrategia de aprendizaje dentro del proceso de migración.
- Que el ícono de t_{Ai} posea la misma silueta que el ícono de t_{Bi} : Según la teoría de la Gestalt⁵, lo primero que ve el usuario de un ícono es su silueta, todas aquellas diferencias y detalles que puedan estar plasmadas en la imagen resultan irrelevantes para traer su significado desde la memoria de largo plazo hacia la de corto plazo. El análisis de compatibilidad es el mismo que se aplica para el caso de las tareas con el mismo nombre.
- Que la funcionalidad de t_{Ai} sea la misma de t_{Bi} : Se presentan dos casos posibles. En el primero la funcionalidad de t_{Ai} es la misma en t_{Bi} pero se llevan a cabo con diferentes pasos, esto implica un re-aprendizaje por parte del usuario que genera cambios en el modelo mental, que sirve de transporte para iniciar el aprendizaje en B. En el segundo la funcionalidad de t_{Ai} es la misma en t_{Bi} y se ejecutan siguiendo los

⁵ Wikipedia (2007). Psicología de la Gestalt. Consultado el 18 de marzo de 2007 en <http://es.wikipedia.org/wiki/Gestalt>

mismos pasos, lo cual implica la reutilización del modelo mental y por ende la familiaridad tiene una alta incidencia.

Si se logran validar estos criterios para cada tarea, se podrá predecir los resultados del proceso de migración. En caso contrario, la migración no tendrá éxito inmediato, pues estará retrasada por la necesidad de algún nivel de re-aprendizaje por parte de los usuarios.

Al demostrar la compatibilidad de las herramientas involucradas en la migración, se espera que los usuarios puedan utilizar el modelo mental que tienen del software origen para operar el software destino.

De este análisis se pueden obtener tres resultados:

- a) **Compatibilidad por familiaridad** que implica la reutilización del modelo mental de A en B, caracterizando el proceso de migración de una alta factibilidad. Este resultado cumple con todos los criterios de compatibilidad descritos anteriormente.
- b) **Compatibilidad por transporte** que implica el re-aprendizaje de nuevas formas para ejecutar las tareas en B, donde no necesariamente haya que reaprender todo el espacio de trabajo, sino al menos su distribución y las técnicas de interacción, nombres e íconos que formarán parte del nuevo modelo mental. Esto caracteriza al proceso de migración como factible, pero necesariamente harán falta métodos de aprendizaje para la formación del modelo mental de B. Este resultado se obtiene cuando falla al menos un criterio de compatibilidad.
- c) **No compatibilidad** entre los modelos, esto supone un proceso de migración lento y difícil para el usuario, puesto que implica el duro proceso de desaprender lo que conoce del software A para aprender una nueva forma de realizar las tareas en B. En este caso no se cumple ninguno de los criterios de compatibilidad.

Estos resultados se obtienen como producto del análisis de la técnica de interacción correspondiente a cada tarea, a través del recorrido cognitivo. La Tabla 2-1 está diseñada para registrar las acciones o pasos requeridos para ejecutar cada tarea, tanto en el software A como en el B, y de esa forma identificar el tipo de compatibilidad presente, cuya justificación se explica en las *Observaciones*.

Análisis de compatibilidad de t_i	
Software A	Software B
Nombre de t_{Ai} :	Nombre de t_{Bi}
Forma de realizarla en A:	Forma de realizarla en B:
Compatibilidad:	
Familiar <u> </u> Transportable <u> </u> No compatible <u> </u>	
Observaciones:	

Tabla 2-1 Ejemplo de Análisis de compatibilidad de t_i

Después de analizar el conjunto de tareas representativas, los resultados pueden ser contabilizados calculando el porcentaje de éstas que son compatibles (ver Tabla 2-2), bien sea por familiaridad o por transporte, para definir el nivel de factibilidad del proceso de migración e incluso hacer proyecciones sobre los costos de entrenamiento necesario.

Resultados de Compatibilidad de software por tareas			
Nombre de t_{Ai}	t_{Ai} es familiar a t_{Bi}	t_{Ai} es transportable a t_{Bi}	t_{Ai} no es compatible con t_{Bi}
t_{A1}	X		
t_{A2}	X		
t_{A3}			X
t_{A4}		X	
t_{A5}	X		
t_{A6}	X		
t_{A7}	X		
t_{A8}		X	
t_{A9}	X		
t_{A10}	X		
Total	70%	20%	10%

Tabla 2-2 Ejemplo de Resultados de Compatibilidad de software por tareas

Se supone que si se tiene un gran porcentaje de tareas compatibles por familiaridad entonces el nuevo **aprendizaje** tendrá lugar a través de la **exploración** que realice el usuario por sí solo sobre el nuevo software. Si se tiene un alto porcentaje de compatibilidad por transporte, se requerirá de algún entrenamiento previo antes de iniciar el proceso de migración, pero si la no compatibilidad entre tareas es de alto porcentaje, la migración requerirá de cursos de entrenamiento fuera del contexto de trabajo.

Para que los productos de software sean compatibles y por ende la migración factible, el 100% debería estar compartido entre las columnas de familiaridad y transporte. Partiendo de esto se puede obtener que el porcentaje de tareas compatibles sea de un 50% para familiaridad y un 50% para transporte, en cuyo caso los usuarios podrían asimilar la migración de forma fácil en un 50%, mientras que el 50% restante requeriría de un reaprendizaje no tan complejo y asimilable en poco tiempo.

El caso que se presenta en el ejemplo de tabla de compatibilidad se puede predecir un 70% de aceptación inmediata por parte de los usuarios, un 20% necesitará un reaprendizaje simple y para enfrentar el 10% no compatible se requerirá reentrenamiento externo que implica un gasto no previsto, sin embargo esto no garantiza completamente que el usuario asimile ese 10%. Por ello es recomendable que el porcentaje de la no compatibilidad sea nulo o el mínimo posible, por ejemplo, si se tienen varias opciones de productos de software para migrar, se sugiere escoger aquella que aporte el menor porcentaje de no compatibilidad, siempre y cuando éste no supere al porcentaje de familiaridad o transporte.

Si el porcentaje de no compatibilidad es considerablemente alto, se puede predecir que la migración requiere un proceso de desaprendizaje, y puede acarrear costos en cursos de entrenamiento.

2.3.4 Validar con los usuarios

La validación con usuarios puede ser omitida si hasta entonces la técnica ha sido suficiente para demostrar la inviabilidad de la migración. En caso contrario es recomendable realizar el trabajo de campo con los usuarios para validar los resultados obtenidos en el Análisis de Compatibilidad.

Esta validación pretende comprobar los resultados de la técnica descrita, aplicando pruebas sobre las herramientas a migrar y una muestra de usuarios expertos, cuyos resultados proveerán las mejores prácticas a seguir y serán el soporte para predecir con propiedad la viabilidad de un proceso de migración.

Así pues, en esta sección se describe la planificación para las pruebas con usuarios y lo necesario para la medición y obtención de resultados.

Se requerirá la participación de, al menos, cinco (5) usuarios expertos y se necesitarán computadores que tengan instalado el software A y el software B.

Las pruebas seguirán las siguientes pautas:

- Se le informará al usuario que el objetivo de la prueba es evaluar la calidad de uso del software, pero nunca que él será el evaluado. Si el usuario comete algún fallo durante la prueba, no será culpa suya, sino del diseño del software
- El facilitador de la prueba necesitará anotar sus observaciones en la tabla de resultados
- La prueba la realizará cada usuario por separado, acompañado sólo del facilitador
- El usuario no debe disponer de más información que la necesaria para permitir su comportamiento espontáneo
- Al usuario se le debe instar a que durante la prueba piense en voz alta, debe decir todo lo que le pase por la cabeza. De hecho, durante la prueba, si el usuario pasa demasiado rato en silencio mirando el interfaz, el facilitador puede preguntar: “¿qué piensa?”
- El usuario deberá hacer lo que le pida el facilitador, expresando qué problemas encuentra, qué no entiende o qué cree que significa cada elemento
- Además, el usuario debe entender que la misión del facilitador es la de observador silencioso, y éste podrá orientarlo en la ejecución de sus tareas respondiendo sólo aquellas preguntas que tengan que ver con la ubicación de los elementos en la interfaz o cambios en la nomenclatura de los mismos, pero no ayudarlo en la total consecución de tareas
- Para evaluar las tareas representativas se le presentará al usuario un escenario compuesto de un ejercicio real del uso del software, por ejemplo, si el software implicado es un editor de texto se trabajará sobre un documento, el cual podría ser una carta
- La tarea representativa denominada t_{Ai} será ejecutada por el usuario sobre el software destino. Cabe destacar que las tareas siempre serán identificadas según el nombre que se le da en el software A, puesto que revelar el nombre que se le da a la tarea homóloga en B podría delatar de cierta forma su manera de ejecutarse.

Los resultados de las pruebas se registrarán en la Tabla 2-3 que se muestra a continuación:

Software A	Software B						
Nombre de t_{Ai}	Tarea Completada t_{Bi}	Tarea no completada	Tarea completada con asistencia	N° de preguntas t_{Bi}	Observaciones		
					Por familiaridad	Por transporte	No compatible
Tarea 1	X			No aplica			
Tarea 2	X			1		La tarea tenía otro nombre.	
....							
Tarea n	X			No aplica			

Tabla 2-3 Ejemplo de Resultados de las pruebas por usuario

Una tarea es completada cuando el usuario logra ejecutarla en su totalidad por sí mismo, sin necesidad de ayuda externa, sin importar el tiempo que tome para ello.

Se considera una tarea no completada cuando el usuario no es capaz de llevarla a cabo debido a la incompatibilidad en la manera de ejecutar las tareas.

Una tarea completada con asistencia tiene lugar si el usuario no puede ejecutarla totalmente por sí solo y necesita la orientación del facilitador para terminarla.

En el análisis de resultados se debe tomar en cuenta que:

- Para comprobar la familiaridad lo importante es que el usuario complete por sí mismo la tarea.
- Para verificar la compatibilidad por transporte son permitidas las preguntas que haga el usuario, cuyas respuestas le permitan transportar el conocimiento de A a B para completar la tarea, pues esto influirá en el reaprendizaje de la misma. Sin embargo, cuando el usuario se enfrenta al nuevo software puede descubrir por sí mismo lo que debe reaprender a través de un aprendizaje exploratorio.

Por otro lado, en el caso de la compatibilidad por transporte, cuando el usuario se ve obligado a reaprender, aparece un proceso de adaptación donde la productividad podría disminuir inicialmente. En la medida que el usuario se acostumbre a los cambios del nuevo software, la productividad irá en aumento.

2.4 Guía práctica para la aplicación de la técnica

Este apartado resume la técnica propuesta en una secuencia de pasos, que han sido extraídos a partir de la interpretación de la misma. Entonces, para aplicar esta técnica en un proceso de migración de software se sugiere orientarse de la siguiente manera:

1) Determinar el software origen y destino

Se debe tener claro cuáles son los productos de software que serán objeto de la migración.

Se recomienda, en la medida de lo posible, contar con varias opciones de software destino para tomar de ellas la que mejor se comporte en la aplicación de la técnica.

2) Seleccionar el conjunto de tareas representativas

Tomar el conjunto de tareas que representen la productividad del usuario experto en el uso del software origen, para ubicarlas luego en el software destino.

3) Evaluar la compatibilidad de los dos productos de software según sus modelos conceptuales

La compatibilidad será determinada según los criterios vistos en el Análisis de Compatibilidad (ver sección 2.3.3), que se resumen en:

- Establecer comparaciones entre las metáforas de interacción de ambos productos de software a través de la exploración visual de las interfaces. Esto consiste en observar la apariencia general de la interfaz que ofrece el software para interactuar con el usuario, comparando la distribución en el espacio de sus elementos, barras, menús, íconos y demás objetos
- Evaluar a través del recorrido cognitivo la compatibilidad entre tareas según las formas de realizarlas, nombres y metáforas de interacción asociadas a ellas. Recoger los resultados de la evaluación en la Tabla 2-2, para plasmar la compatibilidad del software en términos de familiaridad y transporte.

4) Validar con usuarios

Poner en práctica la técnica propuesta a través de la validación con usuarios en un escenario real de migración.

5) Tomar la decisión de migrar o no

Finalmente, decidir enfrentarse o no al proceso de migración según los resultados obtenidos con la aplicación de la técnica.

Capítulo 3: Marco Experimental

En este capítulo se plasma la fase experimental de la propuesta de este Trabajo Especial de Grado, pues se aplica la técnica diseñada para predecir los resultados de un proceso de migración en un caso de estudio, que deja constancia práctica de ella y de cómo puede ser ejecutada. Asimismo se presentan los resultados obtenidos por otros evaluadores en la aplicación de esta técnica para otros casos de estudios.

3.1 Aplicación de la técnica para una Migración de Navegadores Web

Este primer caso de estudio aplica la técnica con ayuda de la guía práctica para predecir una migración entre navegadores web, a fin de justificar que entre éstos existe una alta movilidad de usuarios y además muestran un rápido aprendizaje de uno cuando ya tienen conocimiento previo de otro, haciendo que la migración sea sencilla. Cabe destacar que, comparando diversos navegadores web, se observa un fuerte acercamiento entre sus modelos conceptuales.

En primera instancia se eligen como productos de software para este estudio el navegador **Internet Explorer 7** (IE7) por ser uno de los más populares en los últimos tiempos, y a **Mozilla Firefox 2.0** (MF2.0) por haber mostrado gran aceptación en comparación con otros navegadores, como Netscape®, Opera®, entre otros (w3Counter, 2007).

Windows Internet Explorer⁶ es un navegador de Internet producido por Microsoft® para su plataforma Windows y más tarde para Apple Macintosh. También conocido como IE (Internet Explorer) o MSIE (Microsoft Internet Explorer).

Mozilla Firefox es un navegador libre y de código abierto, fué desarrollado por la Corporación Mozilla y una comunidad de programadores voluntarios externos. Mozilla Firefox es un navegador web multiplataforma, que está disponible en versiones para Microsoft Windows®, Mac OS X® y Linux.

Cumpliendo con lo que establece la técnica, se selecciona como software **origen** Internet Explorer 7 y como software **destino** Mozilla Firefox 2.0.

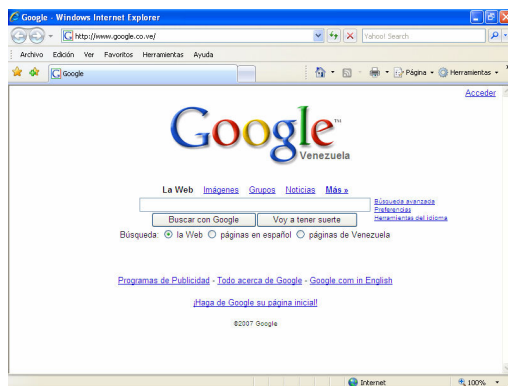
A continuación se presenta el conjunto de **tareas representativas** que serán analizadas para cada navegador:

⁶ Wikipedia (2006). *Navegador Web*. Consultado el 18 de marzo de 2007 en http://es.wikipedia.org/wiki/Navegador_web

- 1) Ir a una dirección web
- 2) Abrir Nueva Ventana
- 3) Abrir Nueva Pestaña
- 4) Ir Atrás o Regresar
- 5) Ir adelante
- 6) Actualizar
- 7) Agregar a Favoritos
- 8) Ver Historial

Seguidamente se comparan los dos productos de software, con el **análisis de compatibilidad** de sus metáforas de interacción y sus tareas representativas.

Internet Explorer 7



Mozilla Firefox 2.0

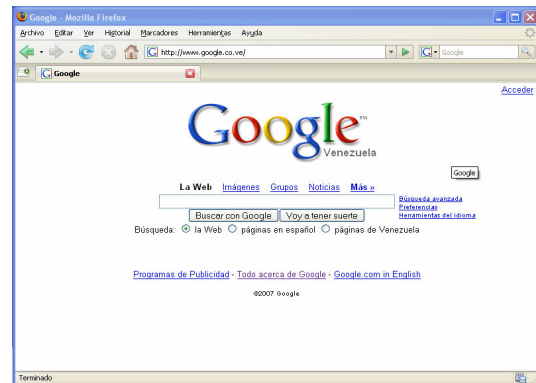


Figura 3-1 Metáfora de Interacción de Internet Explorer 7 y Mozilla Firefox 2.0

En la Figura 3-1 se observa el aspecto de ambas interfaces, así como la similitud entre los espacios de trabajo de IE7 y MF2.0.

En la parte superior se encuentra un conjunto de barras que ofrecen utilidades al usuario para interactuar con el software, mientras que en la parte inferior se observa una barra usada por el sistema para hacer notificaciones al usuario, dejando la parte central de la interfaz como el área de navegación donde se cargan las páginas web.

En cuanto a la distribución del espacio de trabajo se puede decir que, aunque poseen las mismas barras de interacción existen diferencias en la ubicación de las mismas. Esto se muestra en la Figura 3-2 y Figura 3-3.

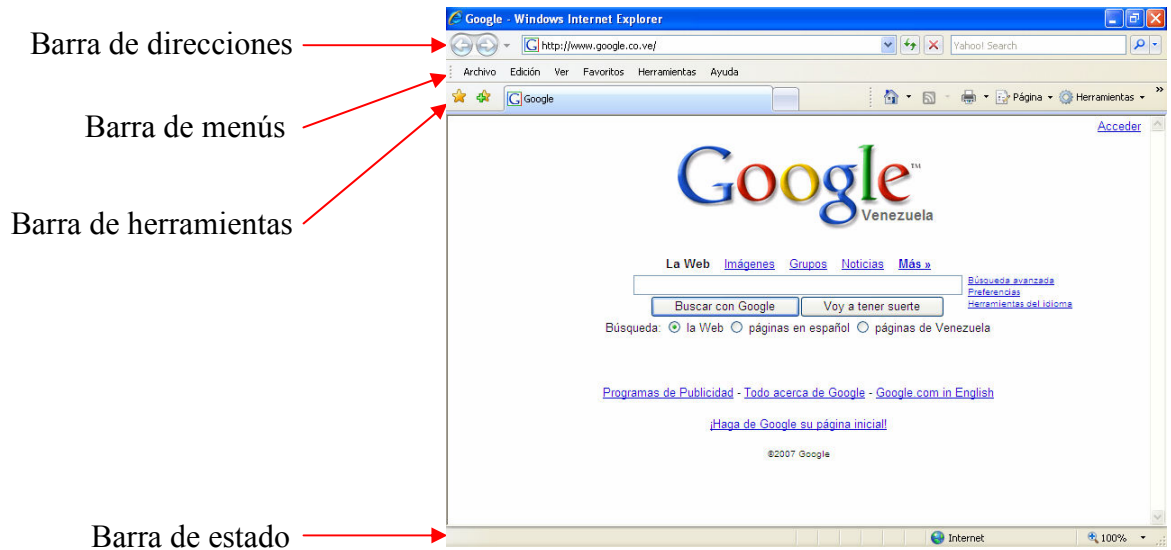


Figura 3-2 Barras de Internet Explorer 7

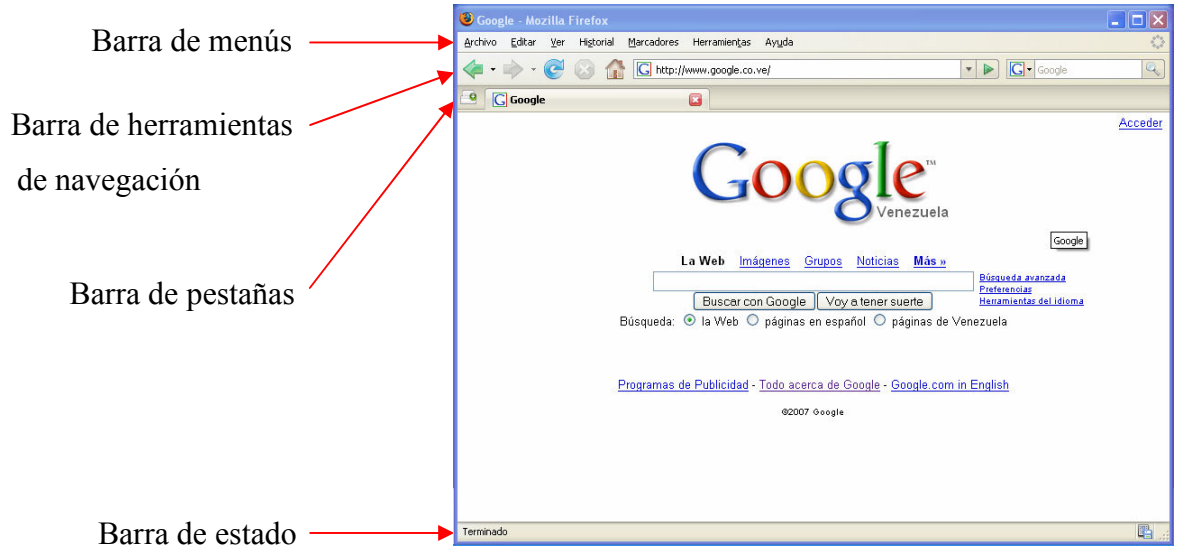


Figura 3-3 Barras de Mozilla Firefox 2.0

En cada barra se puede comparar la ubicación de los elementos de interacción que la componen, para lo cual se establecen semejanzas y diferencias, mostrando las imágenes de las barras de cada navegador a continuación:

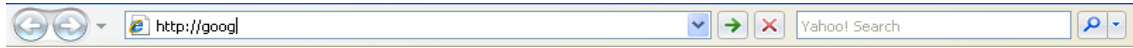


Figura 3-4 Barra de direcciones de Internet Explorer 7



Figura 3-5 Barra de herramientas de navegación de Mozilla Firefox 2.0

La barra de direcciones de IE7 resulta homóloga a la barra de herramientas de navegación de MF2.0 por poseer elementos en común, los cuales se analizan en la Tabla 3-1.

Cabe destacar que la barra de direcciones se encuentra sobre la barra de menús en IE7 y la barra de herramientas de navegación se ubica debajo de la barra de menús en MF2.0.

Nombre	Internet Explorer 7	Mozilla Firefox 2.0	Descripción
Ir atrás			Íconos representados con flechas que mantienen la misma silueta y función. Están ubicados en la misma posición en ambos software. Las pequeñas flechas negras muestran un historial de páginas visitadas.
Ir adelante			
Barra de dirección			En IE7 esta barra se ubica a la derecha de los íconos de ir atrás e ir adelante, pero en MF2.0 está a tres íconos más hacia la derecha de las flechas de atrás y adelante respectivas. La silueta es la misma para ambas metáforas.
Ir a dirección web			Este ícono se ubica en ambos navegadores justo al lado de la barra de dirección y se usa para visitar la página que se ha escrito en la misma. En IE7, después de cargada una página web este ícono se transforma en el botón de actualizar.
Actualizar			Aunque la silueta de estas metáforas no son exactamente iguales pueden asociarse, pues estas flechas intentan dibujar un ciclo o círculo para simular la recarga de la página actual.





			Estos íconos se encuentran en diferente posición dentro de la barra que los contiene.
Detener / Parar			Utiliza el símbolo de la equis (x) y mantiene la presencia del color rojo para detener la carga de una página web. En ambos navegadores estos íconos se ubican a la derecha del botón de actualizar.
Búsqueda			Este formulario de búsqueda se ubica en la misma posición para ambos navegadores. IE7 utiliza el motor de búsqueda de Yahoo® y MF2.0 el de Google®

Tabla 3-1 Barra de dirección de IE7 vs. Barra de Navegación de MF2.0

En la barra de menús de IE7 que se muestra en la Figura 3-6 se encuentran de izquierda a derecha los siguientes menús: *Archivo, Edición, Ver, Favoritos, Herramientas, Ayuda*; y en la Figura 3-7 de MF2.0 las opciones análogas respectivamente son: *Archivo, Editar, Ver, Historial, Marcadores, Herramientas, Ayuda*, a excepción de la opción adicional *Historial*.

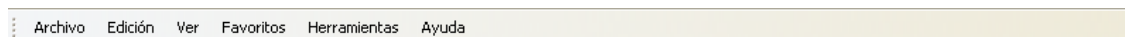


Figura 3-6 Barra de menús de Internet Explorer 7

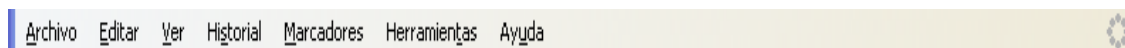


Figura 3-7 Barra de menús de Mozilla Firefox 2.0

La barra de herramientas de IE7 mostrada en la Figura 3-8 ofrece la posibilidad de tener varias pestañas abiertas en una misma ventana, pero además contiene algunos de los elementos más usados en las opciones de la barra de menús. En cambio, su análoga en MF2.0 sólo se encarga de la gestión de pestañas, ver Figura 3-9. Los elementos de estas barras son analizados en la Tabla 3-2.



Figura 3-8 Barra de herramientas de Internet Explorer 7

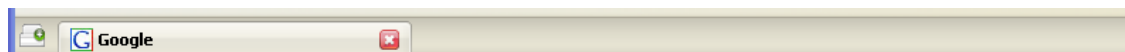

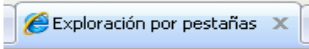
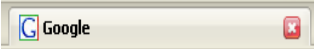







Figura 3-9 Barra de pestañas de Mozilla Firefox F2.0

Nombre	Internet Explorer 7	Mozilla Firefox 2.0	Descripción
Centro de Favoritos			En IE7 la estrella es un acceso directo al centro de favoritos. Mf2.0 no posee este ícono por defecto, pero es posible agregarlo a través de la personalización de barras. Sin embargo las siluetas de estos iconos son diferentes.
Agregar a Favoritos		—	En MF2.0 no existe ningún acceso directo para agregar a favoritos. La alternativa es seguir la ruta <i>Marcadores/Añadir página a marcadores</i> en la barra de menús.
Pestaña			Esta metáfora representa una pestaña, en la cual aparece el ícono y nombre de la página web que esté cargada en ella. Para cerrar la pestaña, ésta posee una x en la esquina superior derecha.
Carga			Este ícono representa la carga de una página web con un aro que gira, su silueta es la misma en ambos navegadores. En IE7 se ubica en la esquina superior izquierda de cada página, pero en MF2.0 se encuentra en el extremo derecho de la barra de menús.
Abrir nueva pestaña			Estos íconos no tienen siluetas similares, pero pueden ser asociados rápidamente porque ambos se encuentran al lado de las pestañas, en IE7 del lado derecho y en MF2.0 a la izquierda en la barra de pestañas.
Ir a Página de inicio (Home)			Este ícono utiliza una casa debido a su nombre en inglés <i>home</i> y sirve para ir a la página configurada como página







			de inicio, su silueta en ambos navegadores es igual.
RSS			Estos íconos se mantienen iguales, utilizan ondas para indicar que el contenido se difunde. Siempre está visible en IE7 y se ilumina de color naranja para indicar que una página posee contenido ya sindicado y es posible suscribirse. En MF2.0 no siempre está visible, aparece para indicar lo mismo, pero se ubica dentro de la barra de dirección.
Imprimir			Los íconos para imprimir poseen la misma silueta, pero en MF2.0 este ícono no se muestra por defecto, sin embargo se puede agregar a su barra de herramientas personalizándola.
Menú Página		—	Este elemento no está disponible en MF2.0, sin embargo las opciones que ofrece pueden encontrarse en los ítems de la barra de menús de MF2.0.
Menú Herramientas		—	Este elemento no está disponible en MF2.0, sin embargo las opciones que ofrece pueden encontrarse en los ítems de la barra de menús de MF2.0.

Tabla 3-2 Barra de herramientas de IE7 vs. Barra de pestañas de MF2.0

La barra de estado de IE7 muestra la palabra *Listo* cuando una página web se ha cargado completamente, en esa misma área van apareciendo los mensajes de los elementos de la página que carga y alguna indicación del número de los que faltan por cargar; luego aparece un símbolo indicando en qué lugar está esa página (Internet, Intranet local, Sitios de confianza y Sitios restringidos), cuando el navegador no puede ubicar la página en ninguno de los otros sitios, indica que está en Internet; seguidamente se ve una lupa con un porcentaje y una flechita desde la cual se puede escoger el tamaño en que redesea ver la página.

En MF2.0 cuando una página web está cargada, la barra de estado muestra la palabra *Terminado*, y en esa misma área el navegador informa el estado de la página a través de mensajes tal como lo hace IE7.

Ambas barras de estado muestran eventualmente íconos que indican la entrada a una página web con certificado de seguridad, bloqueo de *pop-ups*, entre otras.

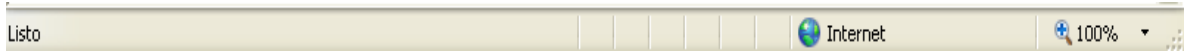


Figura 3-10 Barra de estado de Internet Explorer 7

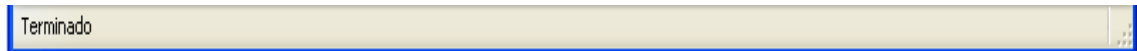


Figura 3-11 Barra de estado de Mozilla Firefox 2.0

Del análisis de las **metáforas de interacción** se puede concluir que, aunque las siluetas y la ubicación de algunos elementos de interacción difieren en ambos navegadores, tales elementos pueden ser proyectados, pues resulta fácil para el usuario acostumbrarse a esas diferencias, ya que va a poder emplear las mismas funcionalidades en una interfaz nueva donde es posible reutilizar el modelo mental que se ha formado con el uso de IE7. La afinidad entre estas metáforas de interacción evidencia la presencia de los conceptos de *affordance*, *constraints* y *mapping* propuestos por Norman. Todo esto constituye resultados favorables, lo cual es un buen indicativo inicial para continuar con el proceso de migración.

Lo siguiente será analizar y comparar la manera específica de utilizar ambas herramientas de software, mediante una muestra de tareas que representan las actividades más comunes que realiza un usuario experto con el software.

A partir de la lista de **tareas representativas** se analiza la compatibilidad de cada una de ellas con respecto a su homóloga en el software destino, mediante recorridos cognitivos para cada una de las formas de llevar a cabo las tareas, como se detalla en las siguientes tablas.

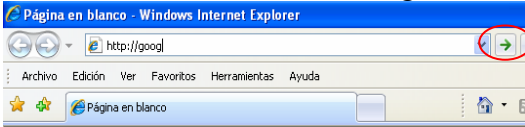
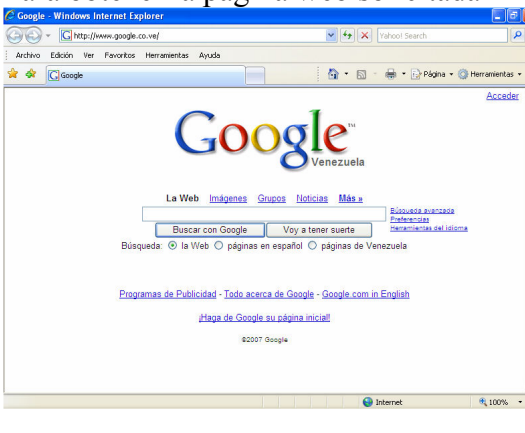
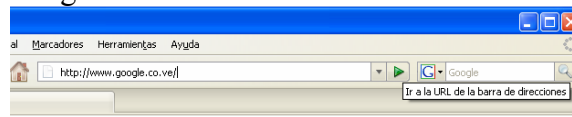
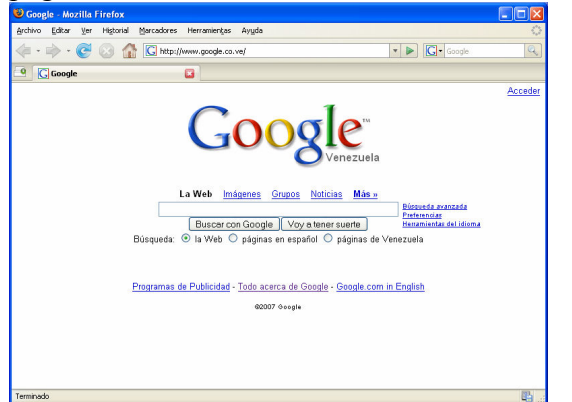
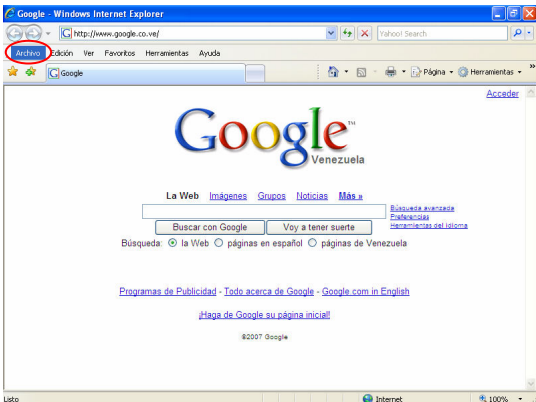
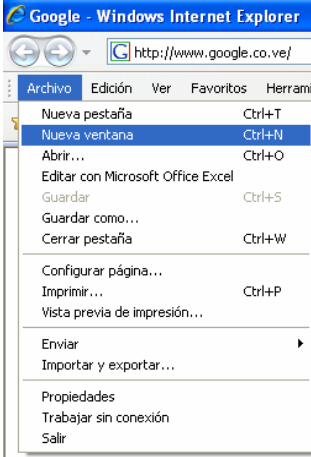
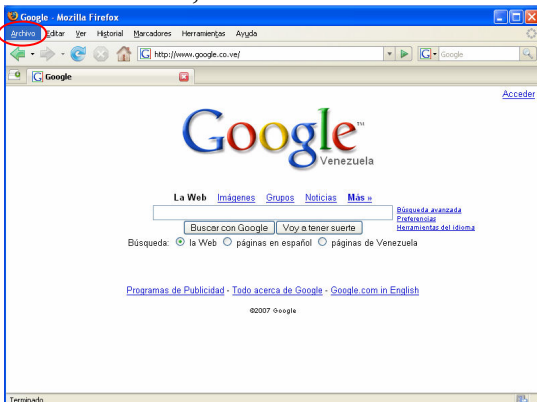
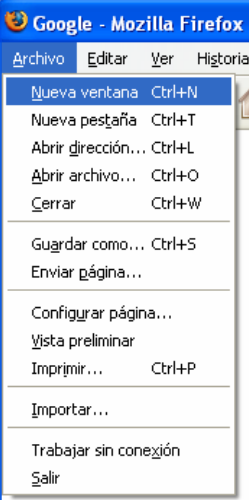
Análisis de compatibilidad de T₁	
Software A: Internet Explorer 7	Software B: Mozilla Firefox 2.0
Nombre de t_{Ai}: Ir a una dirección web	Nombre de t_{Bi}: Abrir dirección
<p>Forma de realizarla en A: Escribir la dirección web en la barra de direcciones y luego presionar la tecla <i>Enter</i> o hacer clic en el ícono de flecha verde, como se observa en la figura</p>  <p>Para obtener la página web solicitada</p> 	<p>Forma de realizarla en B: Para visitar una dirección web, ésta debe escribirse en la barra de direcciones y luego presionar <i>Enter</i> o hacer clic en el ícono de flecha verde, como se observa en la figura</p>  <p>Luego se despliega el contenido de la página visitada</p> 
<p>Compatibilidad: Familiar <u>X</u> Transportable _____ No compatible _____</p>	
<p>Observaciones: a pesar de que la barra de dirección no está ubicada en la misma posición en ambas herramientas, es posible realizar esta tarea con total familiaridad.</p>	

Tabla 3-3 Análisis de compatibilidad para la tarea: Ir a una dirección web

Análisis de compatibilidad de T₂

Software A: Internet Explorer 7	Software B: Mozilla Firefox 2.0
Nombre de t_{Ai}: Abrir nueva ventana	Nombre de t_{Bi}: Abrir nueva ventana
<p>1º Forma de realizarla en A: Hacer clic en la opción <i>Archivo</i> de la barra de menús.</p>  <p>Seguidamente el sistema despliega el siguiente menú, de las cuales se debe elegir la opción <i>Nueva ventana</i>.</p>  <p>2º Forma de realizarla en A: Presionar las teclas <i>Ctrl+ N</i>.</p> <p>3º Forma de realizarla en A: Hacer clic en el menú <i>Página</i> y luego en la opción <i>Nueva Ventana</i>.</p>	<p>1º Forma de realizarla en B: Hacer clic sobre la opción <i>Archivo</i> en la barra de menús,</p>  <p>Luego el sistema muestra un menú para seleccionar la opción <i>Nueva Ventana</i>.</p>  <p>2º Forma de realizarla en B: Con la combinación de teclas <i>Ctrl+N</i>.</p>

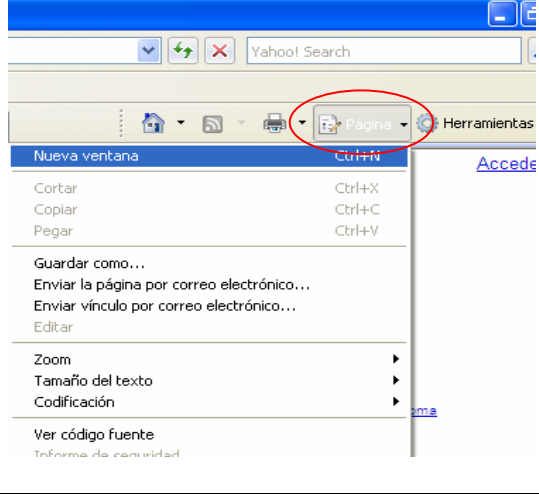
	
<p>Compatibilidad:</p> <p>1º forma: Familiar <u>X</u> Transportable ___ No compatible ___</p> <p>2º forma: Familiar <u>X</u> Transportable ___ No compatible ___</p> <p>3º forma: Familiar ___ Transportable ___ No compatible <u>X</u></p>	
<p>Observaciones:</p> <p>La 1º forma de realizar ambas tareas es familiar, sólo que la opción <i>Nueva Ventana</i> en IE7 y Firefox se encuentra en segunda posición y primera posición respectivamente, sin embargo, este detalle no impide la consecución de la tarea.</p> <p>En MF2.0 no se puede realizar la tarea de la 3º forma y se considera no compatible, porque hay que deshacer este conocimiento para aprender una nueva manera de llevar a cabo la tarea, ya sea de la 1º ó 2º forma.</p>	

Tabla 3-4 Análisis de compatibilidad para la tarea: Abrir nueva ventana

Análisis de compatibilidad de T₃

Software A: Internet Explorer 7

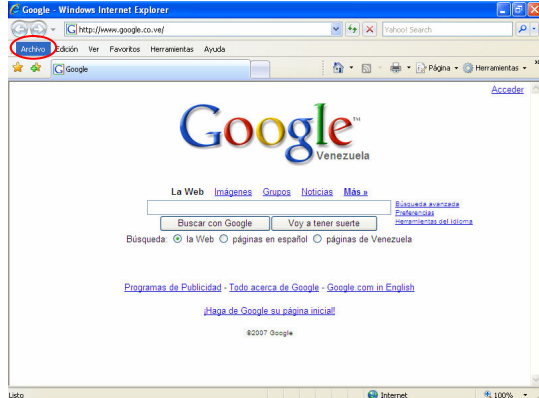
Software B: Mozilla Firefox 2.0

Nombre de t_{Ai}: Abrir nueva pestaña

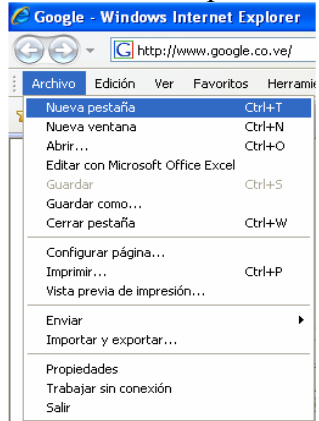
Nombre de t_{Bi}: Abrir nueva pestaña

1º Forma de realizarla en A:

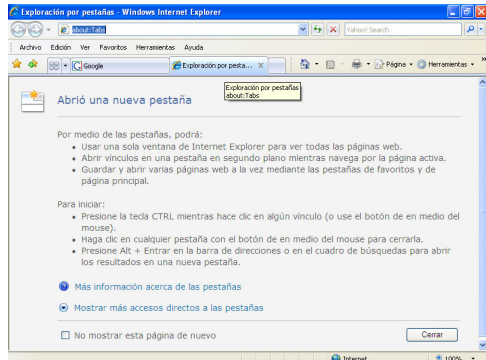
Hacer clic en *Archivo*, de forma similar a la tarea anterior,



Y de las opciones que despliega el sistema seleccionar la opción *Nueva pestaña*.



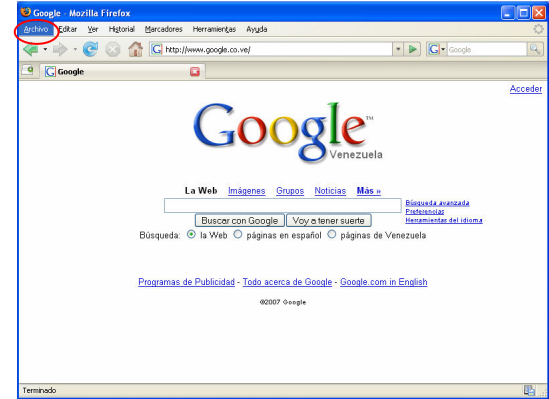
Para finalmente obtener el siguiente resultado:



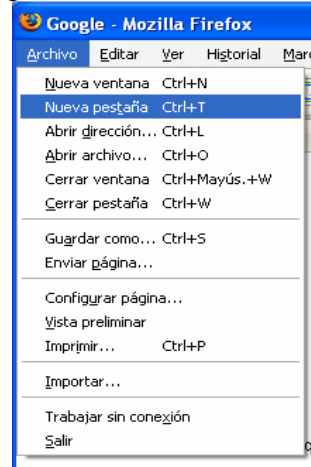
2º Forma de realizarla en A:

1º Forma de realizarla en B:

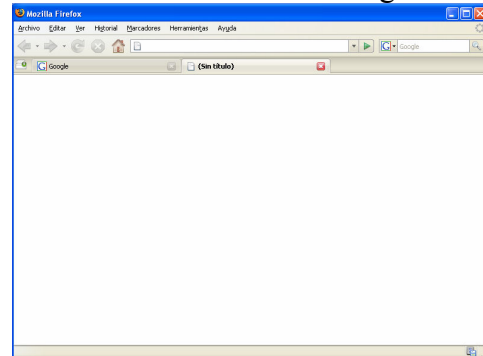
Hacer clic en *Archivo*



Luego escoger del menú la opción *Nueva pestaña*



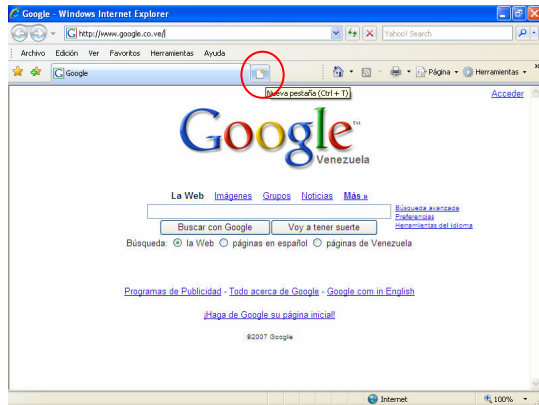
Y finalmente se obtiene lo siguiente:



2º Forma de realizarla en B:

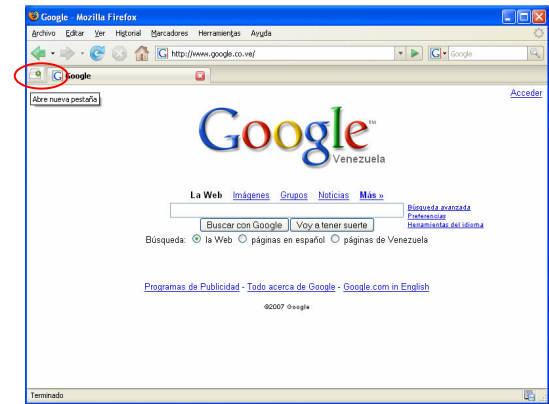
Otra forma de obtener el mismo resultado es presionando el ícono de *Nueva Pestaña*

Otra forma directa para obtener una *Nueva Pestaña* es hacer clic en el ícono que se resalta a continuación.



3º Forma de realizarla en A:
Utilizar el atajo presionando la combinación de teclas *Ctrl+T*.

ubicado en la barra de Pestañas.



3º Forma de realizarla en B:
Con la combinación de teclas *Ctrl+T*.

Compatibilidad:

1º forma: Familiar X Transportable ___ No compatible ___

2º forma: Familiar X Transportable ___ No compatible ___

3º forma: Familiar X Transportable ___ No compatible ___

Observaciones: para realizar ambas tareas se siguen los mismos pasos. Pero como en la tarea anterior, la opción *Nueva Pestaña* en IE7 y Firefox se encuentra en primera y segunda posición respectivamente, pero este detalle no le resta familiaridad a la ejecución de la tarea.

Tabla 3-5 Análisis de compatibilidad para la tarea: Abrir nueva pestaña

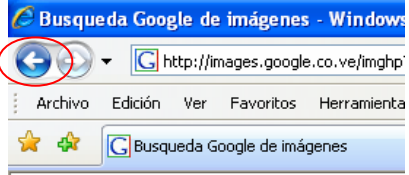
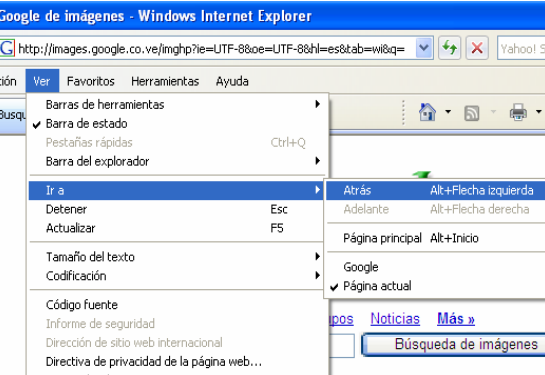

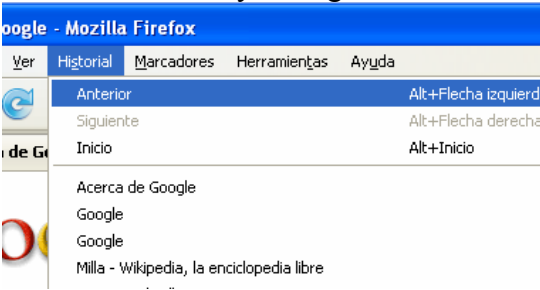
Análisis de compatibilidad de T₄	
Software A: Internet Explorer 7	Software B: Mozilla Firefox 2.0
Nombre de t_{Ai}: Ir atrás o Regresar	Nombre de t_{Bi}: Ir a la página anterior
<p>1º Forma de realizarla en A: Suponiendo que se ha navegado ya por varias páginas dentro de una misma pestaña, para regresar a la pestaña inmediatamente anterior, se hace clic en la flecha blanca orientada hacia la izquierda.</p>  <p>2º Forma de realizarla en A: Seguir la siguiente ruta <i>Ver/Ir a/Atrás</i> y hacer clic en <i>Atrás</i>.</p>  <p>3º Forma de realizarla en A: Presionar la combinación de teclas <i>Alt+Flecha izquierda</i>.</p>	<p>1º Forma de realizarla en B: Luego de haber navegado en una página se puede presionar el ícono de flecha izquierda para volver a la última página visitada.</p>  <p>2º Forma de realizarla en B: También mediante la opción <i>Historial</i> de la barra de menús y se elige <i>Anterior</i></p>  <p>3º Forma de realizarla en B: Con la combinación de teclas <i>Alt+Flecha Izquierda</i>.</p>
<p>Compatibilidad: 1º forma: Familiar <u>X</u> Transportable ___ No compatible ___ 2º forma: Familiar ___ Transportable <u>X</u> No compatible ___ 3º forma: Familiar <u>X</u> Transportable ___ No compatible ___</p>	
<p>Observaciones: para realizar esta tarea de la segunda forma, se requiere deshacer el conocimiento previo, ya que en el menú <i>Ver</i> de MF no se encuentra la opción de <i>Ir atrás</i>; sino que se ubica en el menú <i>Historial</i> y con el nombre de <i>Anterior</i>.</p>	

Tabla 3-6 Análisis de compatibilidad para la tarea: Ir atrás o Regresar


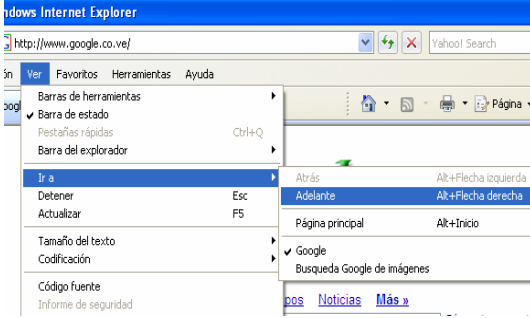

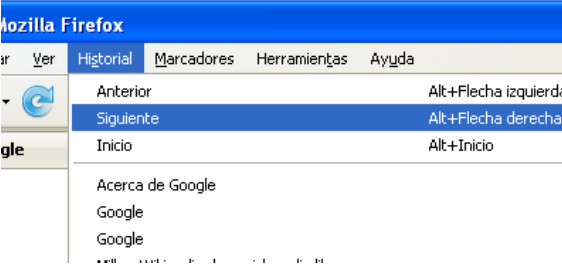
Análisis de compatibilidad de T₅	
Software A: Internet Explorer 7	Software B: Mozilla Firefox 2.0
Nombre de t_{Ai}: Ir adelante	Nombre de t_{Bi}: Ir a la página siguiente
<p>1º Forma de realizarla en A: Luego de visitar páginas anteriores es posible visitar entonces las siguientes, para lo cual se hace clic en la flecha orientada hacia la derecha.</p>  <p>2º Forma de realizarla en A: Seguir la ruta <i>Ver/Ir a/Adelante</i> y hacer clic en <i>Adelante</i>.</p>  <p>3º Forma de realizarla en A: Presionar la combinación de teclas <i>Alt+Flecha derecha</i>.</p>	<p>1º Forma de realizarla en B: Una vez realizada la tarea anterior, se activa el ícono de flecha derecha para ir a la página siguiente</p>  <p>2º Forma de realizarla en B: Mediante el menú <i>Historial</i> opción <i>siguiente</i></p>  <p>3º Forma de realizarla en B: Presionar la combinación de teclas <i>Alt+Flecha derecha</i>.</p>
<p>Compatibilidad: 1º forma: Familiar <u>X</u> Transportable ___ No compatible ___ 2º forma: Familiar ___ Transportable <u>X</u> No compatible ___ 3º forma: Familiar <u>X</u> Transportable ___ No compatible ___</p>	
<p>Observaciones: para realizar esta tarea de la segunda forma, se requiere deshacer el conocimiento previo, ya que en el menú <i>Ver</i> de MF no se encuentra la opción de <i>Ir adelante</i>; pues se ubica en el menú <i>Historial</i> con el nombre de <i>Siguiente</i>.</p>	

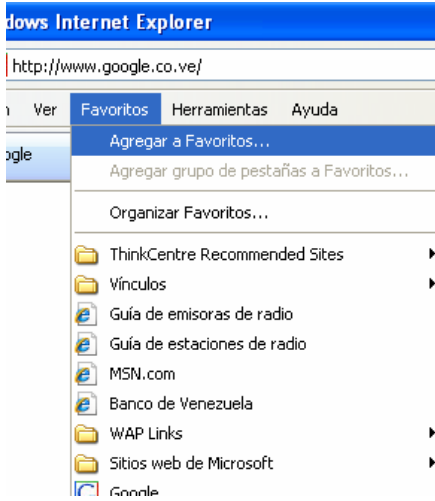
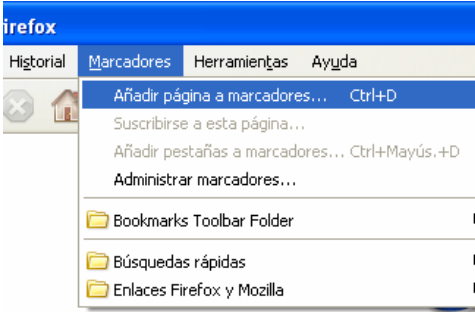

Tabla 3-7 Análisis de compatibilidad para la tarea: Ir adelante

Análisis de compatibilidad de T₆

Software A: Internet Explorer 7	Software B: Mozilla Firefox 2.0
Nombre de t _{Ai} : Actualizar	Nombre de t _{Bi} : Recargar esta página
<p>1º Forma de realizarla en A: Hacer clic en el ícono conformado por dos flechas verdes curvas.</p>  <p style="text-align: center;"></p> <p>Al actualizar, el sistema muestra en la barra de estado, un indicador de carga que se va llenando poco a poco.</p>   <p style="text-align: center;">Listo</p>	<p>Forma de realizarla en B: Se presiona el ícono de flecha circular</p>  <p>Al recargar el sistema muestra un indicador de carga en la barra de estado</p>  
<p>2º Forma de realizarla en A: Hacer clic en la opción <i>Ver</i> de la barra de menús y luego en <i>Actualizar</i></p>  	<p>2º Forma de realizarla en B: Hacer clic en el menú <i>Ver</i> y seleccionar <i>Recargar</i></p>  
<p>3º Forma de realizarla en A:</p>	

Presionar la tecla <i>F5</i> .	3° Forma de realizarla en B: Presionar la tecla <i>F5</i> o la combinación de teclas <i>Crtl +R</i> .
Compatibilidad: 1° forma: Familiar <u>X</u> Transportable <u> </u> No compatible <u> </u> 2° forma: Familiar <u> </u> Transportable <u>X</u> No compatible <u> </u> 3° forma: Familiar <u>X</u> Transportable <u> </u> No compatible <u> </u>	
Observaciones: Se puede decir que la 1° forma de realizar esta tarea es familiar aunque los íconos parezcan distintos, pues en realidad proyectan una silueta que sugiere un giro o círculo que es reconocible por el usuario. En la 2° forma cambia el nombre de la tarea.	

Tabla 3-8 Análisis de compatibilidad para la tarea: Actualizar

Análisis de compatibilidad de T₇	
Software A: Internet Explorer 7	Software B: Mozilla Firefox 2.0
Nombre de t_{Ai}: Agregar a Favoritos	Nombre de t_{Bi}: Añadir página a Marcadores
<p>1º Forma de realizarla en A: Hacer clic en la opción <i>Favoritos</i> de la barra de menús, y seguidamente clic en <i>Agregar a Favoritos</i>.</p>  <p>Al hacer esto, el sistema muestra un diálogo para colocarle un nombre a la página que se está guardando como favorita o aceptar el nombre sugerido, luego se hace clic en <i>Agregar</i> y listo.</p>	<p>1º Forma de realizarla en B: Hacer clic en el menú <i>Marcadores</i> y luego a <i>Añadir página a Marcadores</i>.</p>  <p>Luego de hacer clic, el sistema mostrará una ventana <i>Añadir Marcador</i> que permite escribir el nombre de la página a agregar y escoger la carpeta dónde se guardará</p>  <p>2º Forma de realizarla en B: Presionar la combinación de teclas <i>Ctrl+D</i></p>

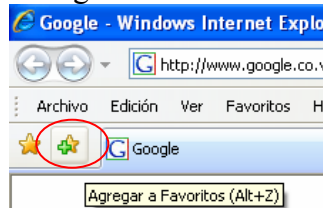


2º Forma de realizarla en A:

Utilizar un atajo, presionando la combinación de teclas *Alt+Z*.

3º Forma de realizarla en A:

Otra manera es hacer clic en el ícono indicado a continuación para obtener el cuadro de diálogo mostrado anteriormente.



Compatibilidad:

1º forma: Familiar Transportable No compatible

2º forma: Familiar Transportable No compatible

3º forma: Familiar Transportable No compatible

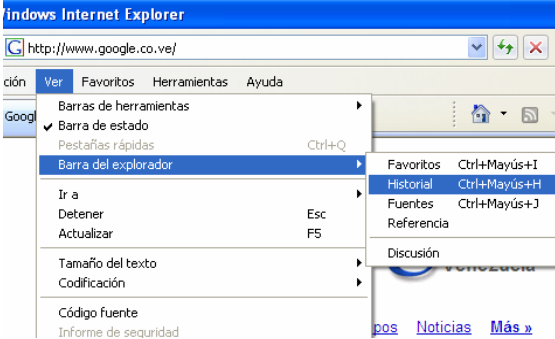
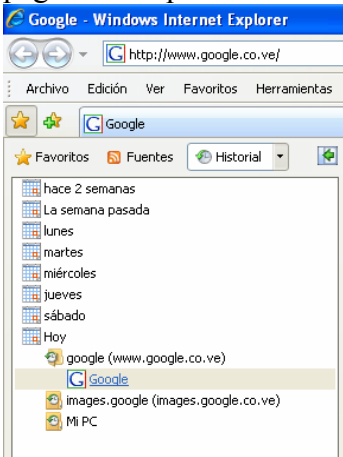


Observaciones: la 1º forma de realizar esta tarea es transportable, porque implica reaprender el nombre de la tarea (de Favoritos a Marcadores).

La 2º forma también requiere un reaprendizaje porque cambia la combinación de teclas del *shortcut*.

No hay manera de ejecutar en MF2.0 la 3º forma de realizar la tarea, por lo tanto se considera una forma no compatible, pues hay que deshacer ese conocimiento para aprender una nueva manera de llevar a cabo la tarea, ya sea de la 1º ó 2º forma.

Tabla 3-9 Análisis de compatibilidad para la tarea: Agregar a favoritos

Análisis de compatibilidad de T₈

Software A: Internet Explorer 7	Software B: Mozilla Firefox 2.0
Nombre de t _{Ai} : Ver Historial	Nombre de t _{Bi} : Historial
<p>1º Forma de realizarla en A: Para ver el historial de páginas visitadas se debe seguir la ruta <i>Ver/Barra del explorador/Historial</i> y hacer clic en la opción <i>Historial</i>.</p>  <p>Para obtener la siguiente respuesta del sistema, donde se puede hacer clic en la página web que se desea volver visitar.</p>  <p>2º Forma de realizarla en A: Presionar la combinación de teclas <i>Ctrl+Mayús+H</i>.</p> <p>3º Forma de realizarla en A: Para obtener el mismo resultado, hacer clic en el menú <i>Herramientas</i> y seguir la ruta <i>Barra de herramientas/Historial</i>.</p>	<p>1º Forma de realizarla en B: Seguir la ruta <i>Historial/Mostrar en la barra lateral</i></p>  <p>El sistema desplegará una barra lateral en la parte izquierda de la ventana con carpetas que contienen las páginas visitadas días atrás.</p>  <p>2º Forma de realizarla en B: Presionar la combinación de teclas <i>Ctrl+H</i>.</p>

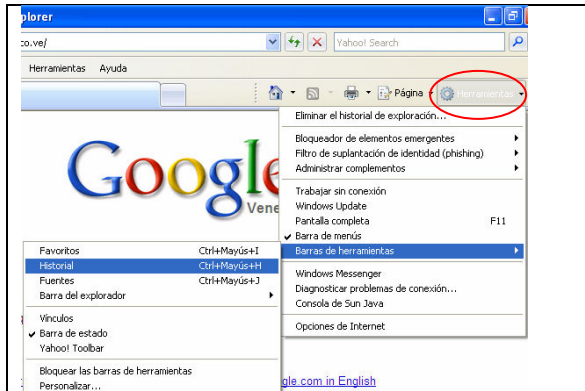
	
<p>Compatibilidad: 1º forma: Familiar ___ Transportable <u>X</u> No compatible ___ 2º forma: Familiar ___ Transportable <u>X</u> No compatible ___ 3º forma: Familiar ___ Transportable ___ No compatible <u>X</u></p>	
<p>Observaciones: La 1º forma es transportable porque la opción de <i>Ver Historial</i> se encuentra en distintos menús para ambas herramientas. En la 2º forma cambia la combinación de teclas, y para la 3º forma no existe una manera parecida de realizarla en MF2.0.</p>	

Tabla 3-10 Análisis de compatibilidad para la tarea: Ver historial

Luego de realizar los recorridos cognitivos se pudo apreciar que existen varias formas de realizar una misma tarea, éstas pueden ser familiares, transportables o no compatibles, de manera que, por ejemplo, una tarea puede no ser totalmente familiar sino más bien medio familiar y medio transportable, lo cual puede ser expresado como 50% familiar y 50% transportable.

En este mismo orden de ideas, se totaliza la asignación de familiaridad, transporte y no compatibilidad en términos de porcentaje por cada forma de realizar las tareas, para finalmente determinar los tipos de compatibilidad que presenta el Navegador Mozilla Firefox 2.0.

t _i	1° forma de realizar la tarea.	2° forma de realizar la tarea.	3° forma de realizar la tarea.	%F	%T	%NC
t ₁	F	-	-	100%		
t ₂	F	F	NC	66,66%		33,33%
t ₃	F	F	F	100%		
t ₄	F	T	F	66,66%	33,33%	
t ₅	F	T	F	66,66%	33,33%	
t ₆	F	T	F	66,66%	33,33%	
t ₇	T	T	NC		66,66%	33,33%
t ₈	T	T	NC		66,66%	33,33%
Leyenda				Total % F	Total % T	Total%NC
F = Familiar T = Transportable NC = No Compatible				58,33 %	29,17 %	12,5 %

Tabla 3-11 Resultados del análisis de compatibilidad por tareas entre IE7 y MF2.0

La Tabla 3-11 revela que todas las tareas representativas de IE7 se pueden ejecutar en MF 2.0, ya sea de forma familiar o por transporte, a pesar de que existan algunas formas no compatibles para ciertas tareas.

El hecho de que el 87,5% esté repartido entre familiaridad y transporte pronostica teóricamente un proceso de migración favorable entre IE7 y MF2.0 para el conjunto de tareas analizadas.

Para comprobar estos resultados, se recurre a la **validación con usuarios**, para lo cual se cuenta con la participación de cinco (5) usuarios expertos en el uso de IE7, quienes cuentan con edades comprendidas entre 20 y 40 años y navegan en Internet en sus trabajos u oficinas. Estos usuarios realizarán las siguientes actividades con la supervisión del facilitador:

1. Ir a la dirección www.ucv.ve

2. Abrir una nueva ventana y visitar la dirección www.cantv.net
3. Abrir una nueva pestaña y visitar la dirección www.google.com y luego navegar (realizar una búsqueda de actualidad). Visitar el primer resultado arrojado por Google
4. Ir a la página anterior
5. Ir a la página siguiente
6. Actualizar página actual
7. Añadir la página actual a favoritos
8. Ver el historial de páginas.

La observación del facilitador sobre el comportamiento de cada usuario en la ejecución de las actividades descritas, permite comprobar la compatibilidad entre IE7 y MF2.0 para el conjunto de tareas representativas seleccionado. Los resultados de este trabajo de campo son recogidos en las siguientes tablas:

Usuario # 1 Nivel Experto							
IE7	MF2.0						
Nombre de T_i	Tarea Completada T_{Bi}	Tarea no completada	Tarea completada con asistencia	Nº de preguntas T_{Bi}	Observaciones		
					Por familiaridad	Por transporte	No compatible
Ir a una dirección Web	x			0	El usuario por sí mismo se ubico en la barra de direcciones.		
Nueva Ventana	x			0	A través de la opción del menú archivo.		
Nueva Pestaña	x			0	Por medio del ícono de pestaña de la barra de pestaña.		
Ir Atrás o regresar	x			0	Mediante el ícono de flecha izquierda.		
Ir adelante	x			0	Mediante el ícono de flecha derecha.		
Actualizar	x			0	Asociación por ícono.		
Agregar a Favoritos			x	1		Al usuario se le ayudó con el nombre de ésta tarea (Favoritos = Marcadores) para que pudiera completarla.	
Ver Historial	x			0		El usuario por sí sólo asoció la palabra <i>Historial</i> de la barra menú para poder ejecutar la tarea, a pesar de que en IE7 se realiza de forma distinta.	

Tabla 3-12 Resultados de las actividades para el usuario #1

Usuario # 2 Nivel experto							
IE7	MF2.0						
Nombre de T_i	Tarea Completada T_{Bi}	Tarea no completada	Tarea completada con asistencia	Nº de preguntas T_{Bi}	Observaciones		
					Por familiaridad	Por transporte	No compatible
Ir a una dirección Web	x			0	El usuario por sí mismo se ubico en la barra de direcciones.		
Nueva Ventana	x			0	A través de la opción del menú archivo.		
Nueva Pestaña	x			0	Por medio del menú archivo.		
Ir Atrás o regresar	x			0	Mediante el ícono de flecha izquierda.		
Ir adelante	x			0	Mediante el ícono de flecha derecha.		
Actualizar	x			0	Asociación por ícono.		
Agregar a Favoritos			x	1		Al usuario se le ayudó con el nombre de ésta tarea (Favoritos = Marcadores) para que pudiera completarla.	
Ver Historial			x	1		Al usuario se le ayudó diciéndole que se ubicara en el menú historial.	

Tabla 3-13 Resultados de las actividades para el usuario #2

Usuario # 3 Nivel experto							
IE7	MF2.0						
Nombre de T_i	Tarea Completada T_{Bi}	Tarea no completada	Tarea completada con asistencia	N° de preguntas T_{Bi}	Observaciones		
					Por familiaridad	Por transporte	No compatible
Ir a una dirección Web	x			0	El usuario por sí mismo se ubico en la barra de direcciones.		
Nueva Ventana			x	2		El usuario preguntó por el menú <i>Página</i> . Se le indicó ubicarse en el menú <i>Archivo</i> .	
Nueva Pestaña	x			0	Asociación por ícono.		
Ir Atrás o regresar	x			0	Mediante el ícono de flecha izquierda.		
Ir adelante	x			0	Mediante el ícono de flecha derecha.		
Actualizar	x			0	Asociación por ícono.		
Agregar a Favoritos			x	1		El usuario preguntó por el acceso directo a Favoritos.	
Ver Historial			x	1		Al usuario se le ayudó diciéndole que se ubicara en el menú historial.	

Tabla 3-14 Resultados de las actividades para el usuario #3

Usuario # 4 Nivel experto							
IE7	MF2.0						
Nombre de T_i	Tarea Completada T_{Bi}	Tarea no completada	Tarea completada con asistencia	N° de preguntas T_{Bi}	Observaciones		
					Por familiaridad	Por transporte	No compatible
Ir a una dirección Web	x			0	Resultó familiar la interacción con la barra de direcciones.		
Nueva Ventana			x	1		El usuario preguntó por el menú <i>Página</i> . Finalmente logró realizar la tarea a través del menú <i>Archivo</i> .	
Nueva Pestaña	x			0	Realizada a través del <i>shortcut</i> .		
Ir Atrás o regresar	x			0	Mediante el ícono de flecha izquierda.		
Ir adelante	x			0	Mediante el ícono de flecha derecha.		
Actualizar	x			0	Realizada a través del <i>shortcut</i> .		
Agregar a Favoritos			x	1		Al usuario se le indicó que en MF2.0 los <i>Favoritos</i> se conocen como <i>Marcadores</i> .	
Ver Historial	x			0		El usuario ubicó rápidamente el menú <i>Historial</i> .	

Tabla 3-15 Resultados de las actividades para el usuario #4

Usuario # 5 Nivel experto							
IE7	MF2.0						
Nombre de T_i	Tarea Completada T_{Bi}	Tarea no completada	Tarea completada con asistencia	N° de preguntas T_{Bi}	Observaciones		
					Por familiaridad	Por transporte	No compatible
Ir a una dirección Web	x			0	El usuario realizó la tarea por sí solo.		
Nueva Ventana	x			0		El usuario tomó su tiempo para explorar por sí solo las opciones del menú hasta lograr la tarea.	
Nueva Pestaña	x			0	Asociación por ícono.		
Ir Atrás o regresar	x			0	Utilizó el ícono de flecha izquierda.		
Ir adelante	x			0	Utilizó el ícono de flecha derecha.		
Actualizar	x			0	Reconoció el ícono de recargar.		
Agregar a Favoritos	x			0		El usuario exploró la barra de menús y se mostró dudoso al utilizar el menú Marcadores, pero finalmente lo logró sin preguntar.	
Ver Historial	x			0		El usuario ubicó por sí solo el menú <i>Historial</i> .	

Tabla 3-16 Resultados de las actividades para el usuario #5

A continuación, en la Tabla 3-17 se totalizan, en términos de familiaridad y transporte, los resultados obtenidos en las pruebas aplicadas a los usuarios, para ser comparados con los resultados arrojados en el análisis de compatibilidad por tareas entre IE7 y MF2.0 vistos en la Tabla 3-11.

Participante (P _i)	N° Tareas completadas por familiaridad	N° Tareas completadas por transporte	N° Tareas completadas por transporte con asistencia	N° Tareas no completadas
P1	6	1	1	0
P2	6	0	2	0
P3	5	0	3	0
P4	5	1	2	0
P5	5	3	0	0
Total	27	5	8	0

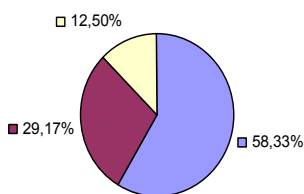
Tabla 3-17 Resultados de la validación con usuarios

En la validación con usuarios, se analizaron 8 tareas por usuario, para un total de de 40 (100%) tareas, de las cuales 27 (67,5%) fueron completadas de forma familiar; 5 se ejecutaron transportando conocimiento y 8 necesitaron ayuda, lo que da un total de 13 (32,5%) tareas completadas por transporte, también se aprecia que las tareas no completadas ocupan un 0%.

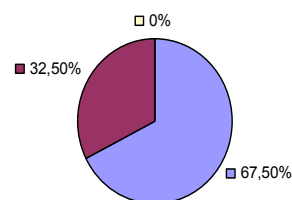
El transporte de conocimiento puede experimentarlo el usuario por sí solo o con un poco de ayuda que despeje sus dudas y le permita llevar a cabo sus tareas. En la práctica se observó que los usuarios se sentían en la capacidad de enfrentar el nuevo software y realizaban las preguntas oportunas para completar todas las tareas asignadas en el navegador MF2.0, aunque no siempre consiguieron las mismas técnicas de interacción que usan en IE7 el aprendizaje exploratorio los condujo a la proyección del modelo mental que ya poseían.

Los resultados de la validación con usuarios superaron la expectativa generada con el análisis de compatibilidad por tareas. El 58,33% de familiaridad pronosticado fue superado en la práctica, llegando a un 67,5% demostrado en la pruebas de usuarios; así también, el 29,17% destinado a conocimiento transportable alcanzó según el trabajo de campo un 32,5%, dejando sin efecto al 12,5% correspondiente a la no compatibilidad en la forma de realizar algunas tareas para este caso de estudio particular. Esta comparación se aprecia en la Figura 3-12.

Resultados del análisis de compatibilidad entre IE7 y MF2.0



Resultados de la validación con usuarios



- Tareas completadas por Familiaridad
- Tareas completadas por Transporte
- Tareas no completadas

Figura 3-12 Resultados de la validación teórica vs. la práctica

Es importante señalar que el 12,5% de no compatibilidad obtenido teóricamente no se reflejó en la práctica, y más bien fue repartido entre los porcentajes de familiaridad y transporte, debido a que los participantes no desistieron al no poder realizar una tarea, sino que espontáneamente buscaron la manera de realizarla explorando el nuevo software o preguntando al facilitador. Esta situación resulta favorable al proceso de migración.

Sin embargo, con una muestra diferente, de usuarios expertos pero menos atrevidos, podría suceder que les cueste más deshacer su conocimiento previo en algunas tareas, y al no conseguir la misma manera de realizarlas en el nuevo software se muestren reacios a aprender nuevas formas de interacción, dando como resultado la frustración en la ejecución de la tarea, lo cual no desaprueba el 12,5% de no compatibilidad teórica obtenido a través de los recorridos cognitivos.

Finalmente se puede decir que los productos de software analizados son compatibles, y que es factible emprender un proceso de migración desde IE7 a MF2.0 con la certeza de que el usuario podrá reutilizar el modelo mental que se formó con el uso de IE7 para enfrentar el manejo de MF2.0. De manera que la técnica aplicada apoya la decisión de iniciar esta migración augurando resultados positivos.

3.2 Resultados de la aplicación de la técnica en otras herramientas de software

Esta sección presenta pequeños casos de estudio donde se aplica la técnica sobre varias herramientas de software.

El grupo docente de la materia Interacción Humano – Computador correspondiente a la Licenciatura de Computación de la Universidad Central de Venezuela, propuso a sus alumnos, como tema de exposición dentro de la materia, la aplicación de la técnica propuesta en este T.E.G., pero enmarcándola dentro del contexto de migración de software propietario a libre, realizando las siguientes actividades en un plazo de un mes:

- Escoger libremente dos aplicaciones homólogas a partir de <http://alts.homelinux.net/>
- Realizar las pruebas propuestas por las tesistas
- Determinar si es factible la migración entre las herramientas homólogas
- Realizar un reporte técnico de la experiencia y los resultados obtenidos.

Para que los alumnos cumplieran con su asignación se les hizo llegar el documento descriptivo de la técnica, adicionalmente se le prestó asistencia teórica a algunos de ellos.

Participaron un total de 7 grupos, conformados por un mínimo de 2 y un máximo de 4 alumnos cada uno, de los cuáles sólo 2 grupos recibieron asesoría personal para aplicar la técnica.

De los grupos participantes, se seleccionaron los 4 mejores que cumplieron a cabalidad con todas las fases de aplicación de la técnica. A continuación, se presentan los resultados obtenidos por estos grupos de alumnos, sobre diferentes productos de software, para un conjunto específico de tareas representativas. Cabe destacar, que de estos 4 grupos, sólo el que trabajó con Microsoft Word y OpenOffice Writer recibió asistencia directa, mientras que los demás grupos trabajaron por su cuenta siguiendo efectivamente las pautas del documento guía.

3.2.1 Acrobat Reader vs. KPDF

El estudio realizado por los alumnos entre *Acrobat Reader*⁷ y *KPDF*⁸ arrojó los resultados mostrados en la Tabla 3-18, en lo cuales se puede apreciar que es posible efectuar una migración entre ambas herramientas, ya que la compatibilidad se muestra mayormente familiar con un 85,5% y por transporte en tan sólo un 14,5%.

Resultados de Compatibilidad por tareas entre Acrobat Reader y KPDF			
Nombre de t_{Ai}	t_{Ai} es familiar a t_{Bi}	t_{Ai} es transportable a t_{Bi}	t_{Ai} no es compatible con t_{Bi}
Abrir Documento	X		
Guardar Documento	X		
Imprimir Documento	X		
Copiar		X	
Atrás/Adelante	X		
Zoom in/out	X		
Buscar	X		
Total	85,5%	14,5%	0%

Tabla 3-18 Resultados del análisis de compatibilidad por tareas entre Acrobat Reader y KPDF

3.2.2 Winzip vs. 7-Zip

La compatibilidad entre los compresores de archivos *Winzip*⁹ y *7-Zip*¹⁰ se describe en la Tabla 3-19, donde se observa que un proceso de migración entre estas dos aplicaciones puede darse en un 80% de forma familiar y con un 20% de transporte de conocimiento.

⁷ Más información sobre el visor de pdf Acrobat Reader en <http://www.adobe.com/la/products/acrobat>

⁸ Sitio web oficial de KPDF <http://kpdf.kde.org/>

⁹ Sitio web oficial de Winzip <http://www.winzip.com/es/>

¹⁰ Sitio web oficial de 7-Zip <http://www.7-zip.org/>

Resultados de Compatibilidad por tareas entre Winzip y 7-Zip			
Nombre de t _{Ai}	t _{Ai} es familiar a t _{Bi}	t _{Ai} es transportable a t _{Bi}	t _{Ai} no es compatible con t _{Bi}
Evaluar archivo en busca de errores	X		
Extraer archivo	X		
Comprimir archivo		X	
Cambiar idioma	X		
Manejar archivos con diferentes extensiones	X		
Total	80%	20%	0%

Tabla 3-19 Resultados del análisis de compatibilidad por tareas entre Winzip y 7-Zip

3.2.3 MSN Messenger vs. Alvaro's MSN

El análisis de compatibilidad aplicado por los evaluadores a las herramientas de mensajería instantánea *MSN Messenger*¹¹ y *Alvaro's MSN*¹² mostró que es posible desarrollar una migración que será familiar en un 60% y requerirá reaprendizaje en un 40%. Esto se detalla en la Tabla 3-20.

¹¹ Más información de MSN Messenger en

<http://messenger.imagine-live.com/Messenger/Launch80/?Locale=es-es>

¹² Más información de Alvaro's MSN en <http://www.amsn-project.net/index.php>

Resultados de Compatibilidad por tareas entre MSN Messenger y Alvaro's MSN			
Nombre de t _{Ai}	t _{Ai} es familiar a t _{Bi}	t _{Ai} es transportable a t _{Bi}	t _{Ai} no es compatible con t _{Bi}
Cambiar fuente	X		
Enviar zumbido		X	
Iniciar video conferencia	X		
Cambiar imagen	X		
Enviar archivo		X	
Total	60%	40%	0%

Tabla 3-20 Resultados del análisis de compatibilidad por tareas entre MSN Messenger y Alvaro's MSN

3.2.4 Microsoft Office Word vs. Open Office Writer

Un proceso de migración entre los editores de texto *Microsoft Office Word*¹³ y *Open Office Writer*¹⁴ es factible según los resultados obtenidos por los evaluadores, la Tabla 3-21 muestra, que para el conjunto de tareas analizadas, la aceptación de los usuarios se daría en forma familiar en un 71,5% y con transporte de conocimiento en un 28,5%.

¹³ Más información de Microsoft Office Word en <http://office.microsoft.com/es-es/word/default.aspx>

¹⁴ Más información sobre Open Office Writer en <http://www.openoffice.org/index.html>

Resultados de Compatibilidad por tareas entre Microsoft Office Word y Open Office Writer			
Nombre de t _{Ai}	t _{Ai} es familiar a t _{Bi}	t _{Ai} es transportable a t _{Bi}	t _{Ai} no es compatible con t _{Bi}
Copiar	X		
Pegar	X		
Insertar tabla		X	
Ortografía y gramática	X		
Guardar	X		
Vista preliminar	X		
Copiar formato		X	
Total	71,5%	28,5%	0%

Tabla 3-21 Resultados del análisis de compatibilidad por tareas entre Microsoft Office Word y Open Office Writer

3.3 Efectividad de la técnica

Para validar la efectividad de esta propuesta se aplicó una encuesta anónima a los alumnos que utilizaron la técnica en los casos de estudio presentados anteriormente.

La encuesta se presenta en la Tabla 3-22, la cual posee preguntas dirigidas a determinar la aceptación y utilidad de la técnica desarrollada, cuyas respuestas van del 1 al 6, donde el 1 representa lo menos efectivo y el 5 lo más efectivo, mientras que un 6 significa no aplica.

Una vez aplicada la encuesta, se analizaron las respuestas obtenidas, mostrando en los siguientes gráficos la frecuencia de las respuestas dadas a cada pregunta.

Así pues, los resultados de la pregunta #1 (Figura 3-13) reflejan que la mayoría de los encuestados consideran fácil la aplicación de la técnica.

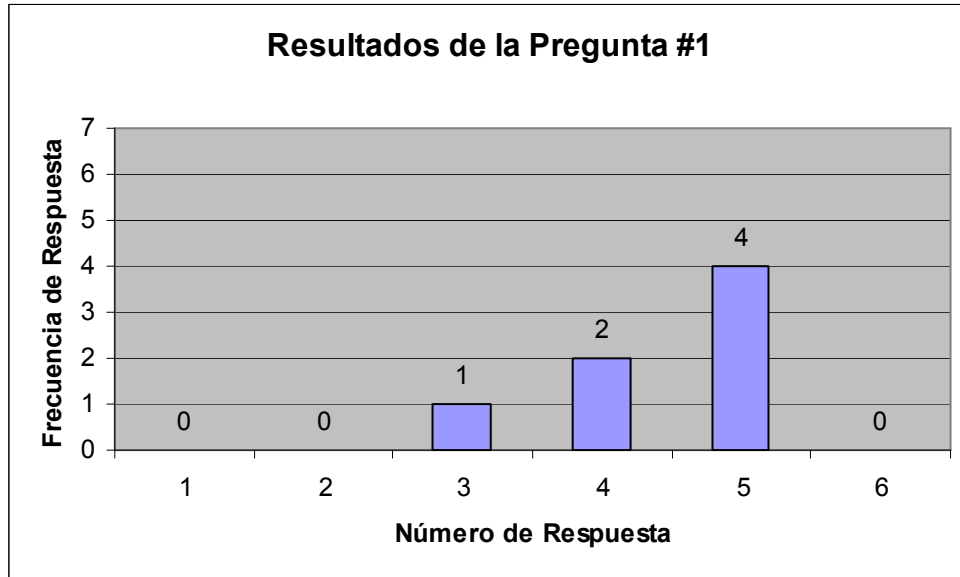


Figura 3-13 Resultados de la Pregunta #1

La pregunta #2 (Figura 3-14) indica que, la mayoría de los encuestados otorga 4 en una escala de sencillez del 1 al 5, al lenguaje utilizado para describir la técnica.

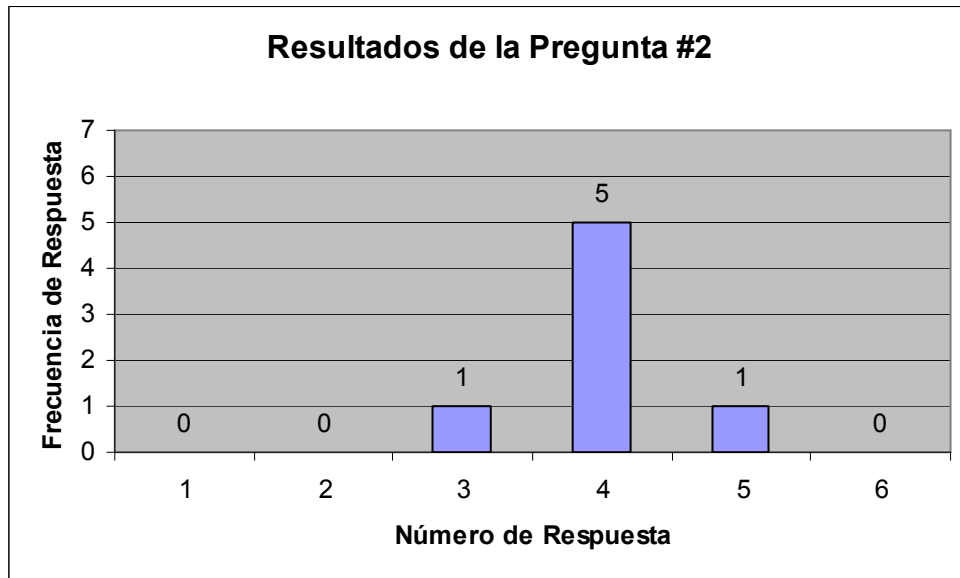


Figura 3-14 Resultados de la Pregunta #2

Según los resultados de la pregunta #3 (Figura 3-15), la mayoría de los encuestados considera que las actividades propuestas en la técnica fueron adecuadas.

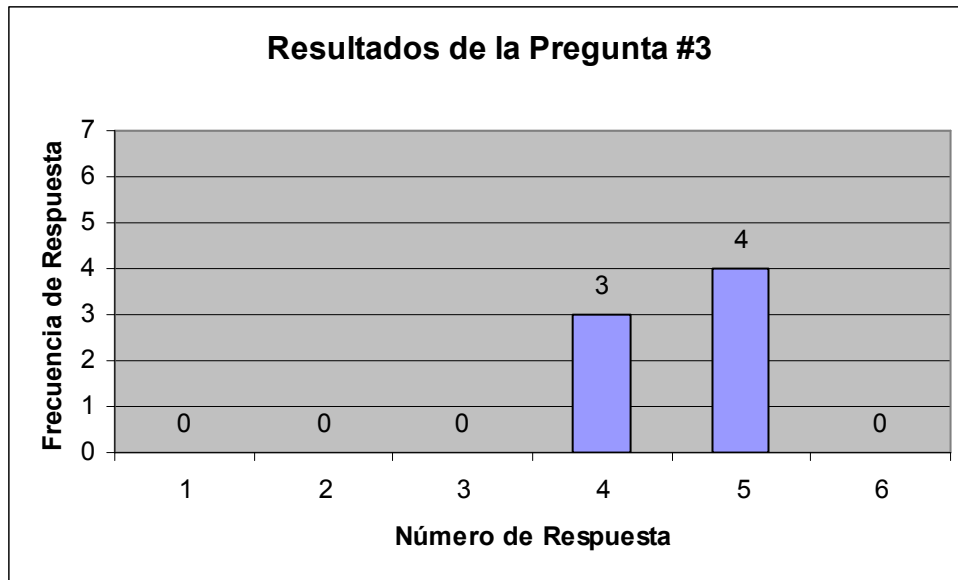


Figura 3-15 Resultados de la Pregunta #3

Las frecuencias de respuesta para la pregunta #4 (Figura 3-16) indican que la mayor parte de los encuestados considera útil el recorrido cognitivo como vía para determinar la compatibilidad entre los productos de software.

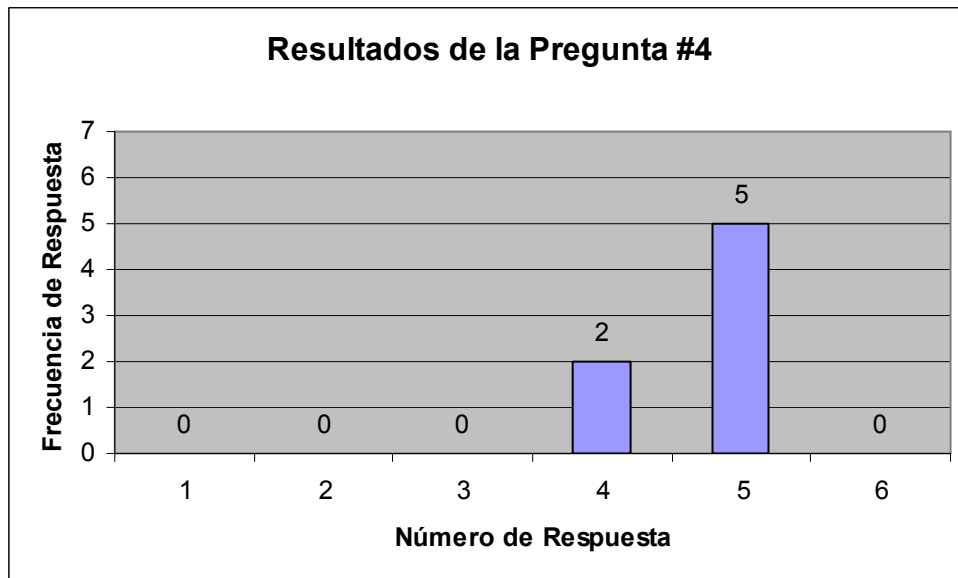


Figura 3-16 Resultados de la Pregunta #4

Los resultados de la pregunta #5 (Figura 3-17) muestran que la mayor parte de los encuestados consideró fácil de entender el concepto de compatibilidad por familiaridad.

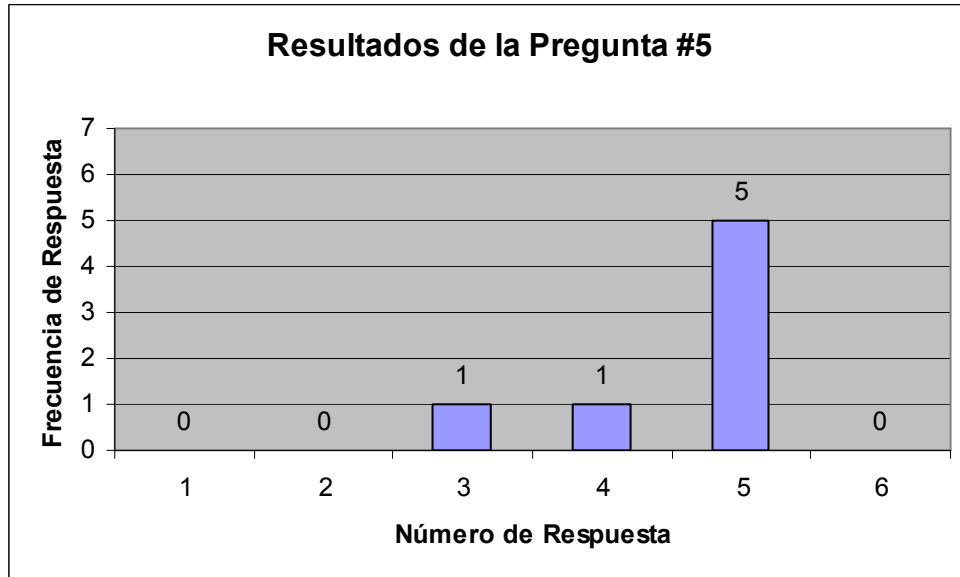


Figura 3-17 Resultados de la Pregunta #5

La mayoría de los encuestados consideró fácil de entender el concepto de compatibilidad por transporte, según los resultado de la pregunta #6 (Figura 3-18).

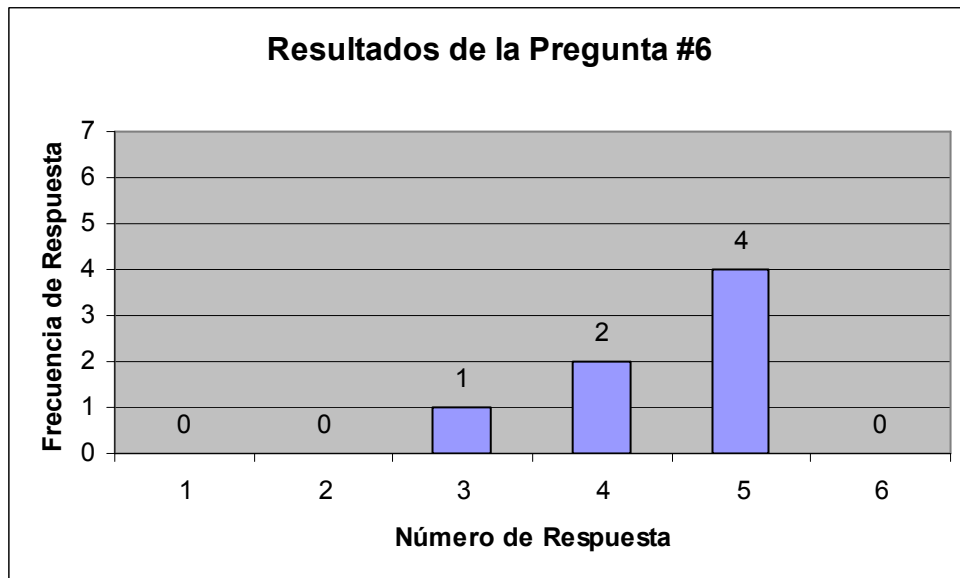


Figura 3-18 Resultados de la Pregunta #6

Los resultados de la pregunta #7 (Figura 3-19) reflejan que la mayoría de encuestados opina que la comparación de modelos conceptuales para la migración de software fue adecuada.

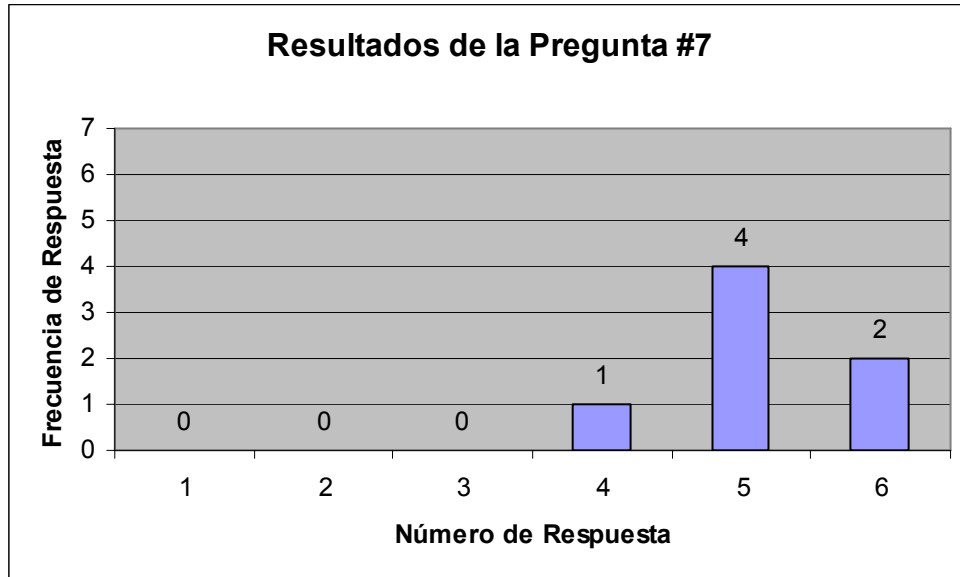


Figura 3-19 Resultados de la Pregunta #7

Los resultados de la pregunta #8 (Figura 3-20) referente a si la técnica permitió realizar una predicción concreta del resultado de la migración, muestran que la mayoría de las respuestas de los usuarios se encuentran repartidas entre la opción 4 y 5 sobre una escala del 1 al 5 de totalmente desacuerdo - totalmente de acuerdo.

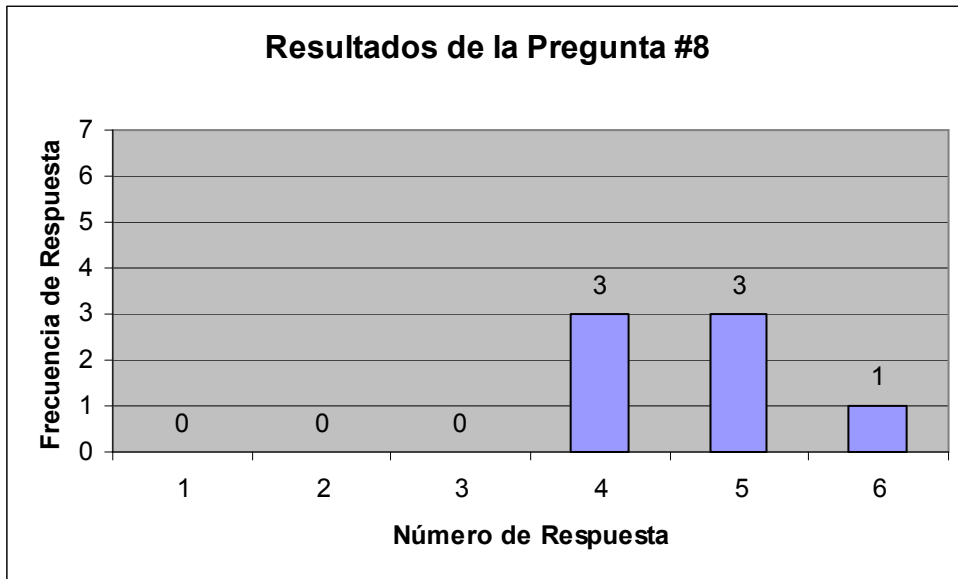


Figura 3-20 Resultados de la Pregunta #8

Según los resultados de la pregunta #9 (Figura 3-21), la mayor parte de los encuestados considera adecuado recomendar esta técnica a otras personas que deseen realizar una migración de software.

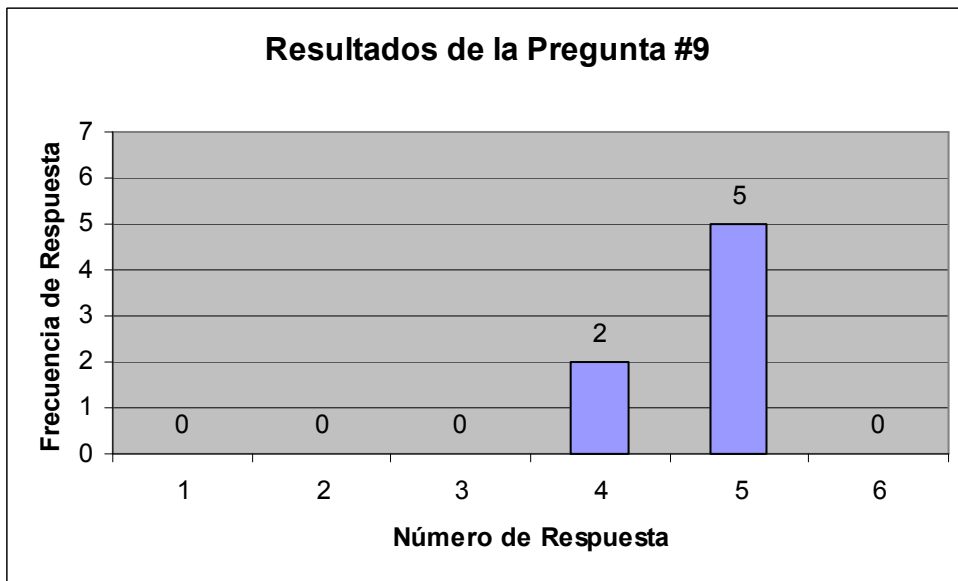


Figura 3-21 Resultados de la Pregunta #9

Como puede observarse, las respuestas preferidas oscilan entre 3, 4, 5 y 6, siendo más comunes la 4 y 5, lo que indica que, la percepción de los encuestados sobre la técnica se acerca a catalogarla como realmente efectiva.

Finalmente, se muestra el resultado total de la encuesta en la Figura 3-22, en la cual se puede apreciar de forma visual que los encuestados muestran tendencia a dar la respuesta más favorable, con lo cual se concluye que la técnica resulta útil, efectiva y fácil de aplicar.

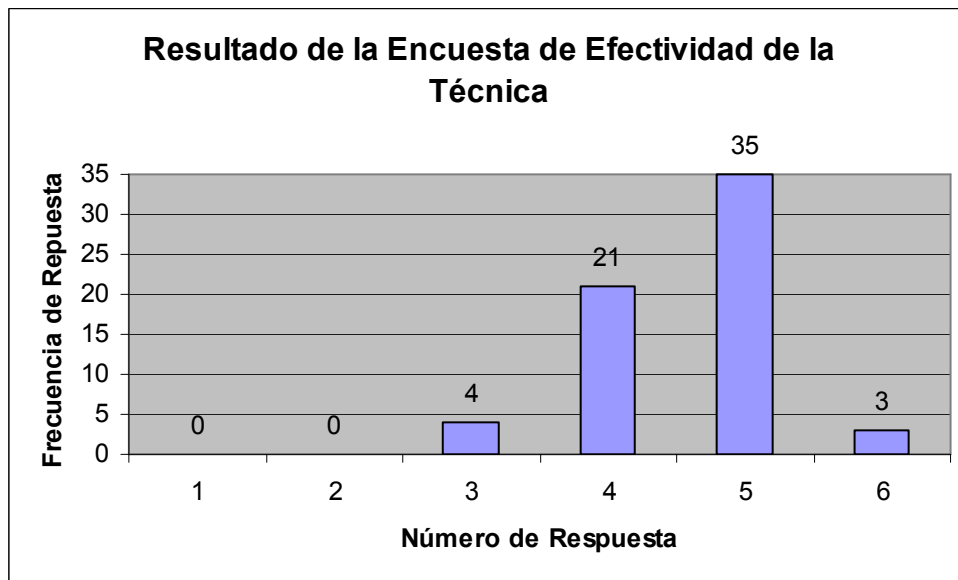


Figura 3-22 Resultado de la Encuesta de Efectividad de la Técnica

Conclusiones

Todo proceso de migración de software debe ser planificado, tomando en cuenta los aspectos humanos, para lograr la aceptación del usuario final.

En este sentido, el objetivo de este Trabajo Especial de Grado fue alcanzado, dando como resultado el desarrollo de una técnica para predecir los resultados de un proceso de migración, que permite al usuario enfrentarse y superar la transición por sí mismo a través de un aprendizaje exploratorio, la cual podrá ser aprovechada por empresas, instituciones o personas que deseen abordar una migración sobre terreno seguro sin arriesgar su inversión, evitándoles gastos innecesarios como producto del fracaso de la migración o pagos de cursos de entrenamiento.

Por consiguiente, conocer muy bien el modelo mental que posee el usuario experto sobre un software, es decir, el conocimiento previo, constituye la base para migrar hacia otro sistema cuyo modelo conceptual sea acorde. Mientras más compatible sea el software destino respecto al software origen, el modelo mental del usuario no sufrirá muchos cambios y podrá ser reutilizado. Así se tiene que, la clave para determinar los resultados de una migración está en la proyección del modelo mental sobre el software destino.

Dentro de este marco, para la elaboración de la propuesta presentada se aprovecharon algunas técnicas de usabilidad, que si bien indican cuando un software es usable, fueron aplicadas en este caso para estudiar la compatibilidad entre el sistema origen y el destino, verificando que una herramienta puede ser tan usable como la otra si están basadas en las mismas o similares técnicas de interacción.

Por otro lado, la productividad de los usuarios no debe verse afectada por una migración, sin embargo ésta puede disminuir con el cambio de software, pero si se predice un buen resultado para la migración, la productividad será recuperada en un corto tiempo donde el usuario se adapte a la proyección de las estructuras cognitivas que posee.

La técnica fue aplicada en un caso de estudio sobre navegadores web para predecir los resultados de una migración de Internet Explorer 7 hacia Mozilla Firefox 2.0, y para su validación, fue oportuna la participación de un grupo de usuarios que ayudaron a constatar, en un escenario real, los pronósticos obtenidos en el correspondiente análisis de compatibilidad por tareas, que incluso fueron superados.

Además, la efectividad de la técnica fue comprobada a través de una encuesta realizada a un grupo de personas que experimentaron su aplicación en varios casos de estudio.

Las personas que aplicaron la técnica, contaron con un documento explicativo de la misma, al cual se refirieron como fácil de entender y de aplicar. Cabe resaltar que, estas personas, casi no solicitaron asistencia a los creadores de la técnica, obteniendo resultados favorables, según su criterio y análisis al usarla; con lo cual fue posible predecir el resultado de los procesos de migración de ciertas herramientas de software para un conjunto propio de tareas específicas. Esto mostró buenas experiencias al seguir la técnica, lo que puede indicar que ésta posee una buena usabilidad de la misma para predecir resultados de migraciones en diferentes entornos, sin necesidad de personal capacitado, más allá que expertos en la herramientas implicadas.

Finalmente, resulta claro que el nivel de compatibilidad revelado por la técnica va a determinar el nivel de éxito en un proceso de migración de software.

Recomendaciones

Para todo proceso de migración es recomendable fomentar el interés en el nuevo software entre los usuarios finales, resaltando las ventajas que pueda ofrecer, para evitar el rechazo injustificado ante el cambio de software.

El tiempo que requiere un usuario para ejecutar una tarea específica en un software puede considerarse como una medida para determinar la productividad. En este sentido se sugiere que, al validar con usuarios la técnica propuesta, se tome en cuenta las siguientes recomendaciones:

- El evaluador puede medir el tiempo que utiliza el usuario para llevar a cabo la tarea i (t_i perteneciente al conjunto de tareas representativas) tanto en el software origen (A) como en el software destino (B) para ser comparados.
- Si el tiempo es mayor en B y las tareas fueron completadas con o sin asistencia, indicará que, aunque el proceso de migración sea factible, afectará en la productividad del usuario. En este caso, el usuario invertirá tiempo en un proceso de aprendizaje para acostumbrarse al nuevo software, hasta que con la práctica finalmente logre alcanzar el nivel de productividad que poseía con el software original.
- Si el tiempo de ejecutar las tareas en B es igual o menor que en A se tendrán más puntos a favor del éxito del proceso de migración, pues esto asegura la rápida asimilación del nuevo sistema por parte del usuario.

Por otra parte, sería recomendable luego de finalizar las pruebas que validan la técnica, aplicar una encuesta de satisfacción a los usuarios con la finalidad de saber si se sintieron a gusto o les pareció traumático el cambio, esto conducirá la aceptación o rechazo del nuevo software.

La técnica está planteada para predecir el resultado de las migraciones de un software A hacia un software B, sin embargo, si el sistema B no existiera es posible utilizar la técnica para el diseño futuro de éste, como apoyo a la fase de levantamiento de requerimientos de B.

Referencias

Answer (2007). *Dictionary Answers*. Consultado el 26 de junio de 2007 en <http://www.answers.com/mapping?cat=health>

Arano, Graciela & Asteggianti, Silvana (1999). *Las Metáforas en la interacción Hombre – Computador*. Consultado el 03 de marzo de 2007 en www.crnti.edu.uy/05trabajos/interface/hcisil.doc

Cañas, José, Salmerón, Ladislao & Gámez, Pilar (2001). *El factor humano*. España: Universidad de Granada. Consultado el 10 de junio de 2006 en <http://griho.udl.es/ipo/pdf/02FacHum.pdf>

Cueva, J.M. *Métricas de Usabilidad en la Web*. Consultado el 26 de marzo de 2007 en <http://www.di.uniovi.es/~cueva/asignaturas/doctorado/2004/MetricasUsabilidad.pdf>

Dix, A., Finlay, J., Abowd, G. & Beale R. (1993). *Human-Computer Interaction*. NJ: Prentice Hall, Englewood Cliffs, (1st edition).

Figuera Y, Júpiter (1993). *Matemática 8º grado*. Venezuela: Colegial Bolivariana.

GRIHO a: Grupo de Investigación en Interacción Humano Ordenador (2005). *Modelo mental y modelo conceptual*. Consultado el 17 de junio de 2006 en <http://griho.udl.es/mpiuua/modelosmyc.htm>

GRIHO b: Grupo de Investigación en Interacción Humano Ordenador (2005). *Usabilidad: ¿En qué momento se ha de considerar?*. Consultado el 17 de junio de 2006 en <http://griho.udl.es/mpiuua/consideracion.htm>

Hom, James (2003). *The usability Methods Toolbox*. Consultado el 20 de septiembre de 2005 en <http://jthom.best.vwh.net/usability/>

ISO (2000). *International standards for HCI and usability*. Consultado el 28 de marzo de 2006 en http://www.usabilitynet.org/tools/r_international.htm#9126-1

Johnson, Laird (1983). *Mental models: towards a cognitive science of language, inference, and consciousness*. Cambridge, Mass: Harvard University Press. Consultado el 12 de junio de 2006 en http://www.filos.unam.mx/POSGRADO/seminarios/pag_robertp/paginas/cognitivism.html#modelos

Lipschutz, Seymour (1991). *Teoría de conjuntos y temas afines*. México: Mc Graw Hill.

López, Olga L. (1996). *Gran diccionario enciclopédico visual*. Colombia: LTDA

Lorés Jesús, Granollers Toni & Perdrix F (2004). *Incorporación de Usuarios en la evaluación de la Usabilidad por Recorrido cognitivo*. Consultado el 10 de marzo de 2007 en <http://griho.udl.es/publicacions/2004/Recorrido-Cognitivo-INTERACCION.PDF>

Lorés, Jesús. & Granollers Toni & Lana Sergi (2002). *Introducción a la Interacción persona ordenador*. Consultado el 10 de marzo de 2007 en <http://griho.udl.es/ipo/pdf/01Introd.pdf>

Lorés, Jesús. & Gimeno Joan Manuel (2001). *Metáforas, estilos y paradigmas*. Consultado el 10 de marzo de 2007 en <http://griho.udl.es/ipo/pdf/03Metafo.pdf>

Mercovich, Eduardo (2000). *Ponencia sobre Diseño de Interfaces y Usabilidad: cómo hacer productos más útiles, eficientes y seductores*. Consultado el 10 de junio de 2006 en <http://planeta.gaiasur.com.ar/infoteca/siggraph99/disen-de-interfaces-y-usabilidad.html>.

Mercovich a, Eduardo (2000). *Cómo hacer un test de usabilidad de un sitio*. Consultado el 09 de mayo de 2006 en <http://planeta.gaiasur.com.ar/infoteca/siggraph99/test-de-usabilidad-de-un-sitio.html>.

Michlmayr, Martín (2005). *Software Process Maturity and the Success of Free Software Projects*. Consultado el 4 de junio de 2006 en http://pdf2html.spawncamp.net/pdf2html.php?url=http://www.cyrius.com/publications/michlmayr-process_maturity_success.pdf.

Ministerio de Ciencia y Tecnología (2005). *Software Propietario, Uso y adquisición en la Administración Pública venezolana*. Caracas: Gráficas Franco.

Moore, G. A. (1995). *Inside the Tornado: Marketing Strategies from Silicon Valley's Cutting Edge*. New York: HarperBusiness.

Netproject (2003). *Directrices IDA de migración a software de fuentes abiertas*. Consultado el 06 de enero de 2006 en http://www.csi.map.es/csi/pdf/Directrices_IDA_software_libre_ESv1_r.pdf

Nielsen, Jakob (1994). *How to Conduct a Heuristic Evaluation*. Consultado el 7 de junio de 2006 en http://www.useit.com/papers/heuristic/heuristic_evaluation.html.

Nielsen, Jakob (1994). *Usability Engineering*. Academic Press Professional, Boston, MA.

Nielsen (1994). *Inspecciones*. Consultado el 05 de junio de 2006 en <http://is.ls.fi.upm.es/udis/miembros/xavier/usabilityframework/tecnicas/inspecciones.php>

- Nielsen, Jakob (2001). *Usability Metrics*. Consultado el 26 de marzo de 2007 en <http://www.useit.com/alertbox/20010121.html>
- Nielsen, Jakob (2005). *Ten Usability Heuristics*. Consultado el 26 de marzo de 2007 en http://www.useit.com/papers/heuristic/heuristic_list.html
- Norman, Donald. A (1983). "*Some observations on mental models*" en *Mental models* (GENTNER D. y STEVENS A., eds.). NJ: Lawrence Erlbaum, Hillsdale.
- Norman, Donald. A. (1986) *Cognitive Engineering en User centered system design: new perspectives on human-computer interaction*. NJ: Lawrence Erlbaum Associates, Hillsdale.
- Norman, Donald A. (1990). *The design of everyday things*. Capítulo 4, pág 81.
- Norman, Donald. A. (1999). *The invisible computer: why good products can fail, the personal computer is so complex, and information appliances are the solution*. Cambridge Mass: MIT Press.
- Poggioli, Lisette (1989). *Estrategias cognoscitivas: una perspectiva teórica*. Consultado el 19 de enero de 2007 en <http://www.fpolar.org.ve/poggioli/poggio13.htm>
- Pressman, Roger S. (2002). *Ingeniería del Software, un enfoque práctico* (quinta edición). Madrid: Mc Graw Hill.
- Sidar (2000). *Pero.. ¿qué es, realmente, la usabilidad?*. Consultado el 28 de marzo de 2006 en <http://www.sidar.org/recur/desdi/traduc/es/visitable/quees/usab.htm>.
- Sidar a (2000). *Recopilación de métodos de Usabilidad*. Cinsultado el 28 de marzo de 2006 en <http://www.sidar.org/recur/desdi/traduc/es/visitable/Herramientas.htm>
- Smith, S & Ugen, D & Mankoski, A & Frishberg, N & Pedersen, N & Benson, C (2001, julio). *GNOME Usability Study Report*. Consultado el 3 de mayo de 2006 desde http://developer.gnome.org/projects/gup/ut1_report/report_main.html.
- Shneiderman, B (1987). *Designing the User Interface*, Addison-Wesley.
- W3Counter (2007). *Global Web Stats*. Consultado el 8 de junio de 2007 en <http://www.w3counter.com/globalstats.php?date=2007-05-20>