



03063

**UNIVERSIDAD NACIONAL AUTÓNOMA  
DE MÉXICO**

**POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN**

**“SISTEMA PARA EL ANÁLISIS AUTOMATIZADO  
DE LA INTERACCIÓN DE UN USUARIO  
CON SITIOS WEB”**

**T E S I S**

**QUE PARA OBTENER EL GRADO DE:**

**MAESTRO EN CIENCIAS  
(COMPUTACIÓN)**

**P R E S E N T A :**

**CÉSAR CÁRDENAS DESALES**

**DIRECTOR DE TESIS:**

**DR. FERNANDO GAMBOA RODRÍGUEZ**

**México, D.F. Julio de 2005**

m345324



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

*A Dios:*

*Mi Señor, tú eres todo para mí:  
el amanecer de mi oscuridad,  
mi felicidad y mi fuerza,  
mi esperanza y mi vida misma.*

*Te ofrezco el trabajo de mis manos,  
de mi inteligencia y de mi corazón.*

*A mi Madre:*

*Con todo mi amor y mi ternura,  
aquí está el fruto  
del esfuerzo de nosotros dos.*

*Sigo siendo el niño  
que nunca te ha dejado de amar  
y de necesitar  
ni un instante de mi vida.*

# Índice

## Introducción

## Capítulo I. Antecedentes de la automatización de la evaluación de interfaces de usuario

- 1.1 Definición de usabilidad
- 1.2 Evaluación de usabilidad
- 1.3 Métodos de evaluación de usabilidad
  - 1.3.1 Métodos de inspección
    - 1.3.1.1 Inspección de estándares y consistencia
    - 1.3.1.2 Evaluación heurística
  - 1.3.2 Métodos de análisis
    - 1.3.2.1 Análisis cognitivo
    - 1.3.2.2 GOMS
  - 1.3.3 Pruebas empíricas
  - 1.3.4 Métodos de indagación directa de la experiencia del usuario
    - 1.3.4.1 Entrevistas
    - 1.3.4.2 Cuestionarios a usuarios
- 1.4 Problemática de las técnicas no automatizadas de evaluación de interfaces
- 1.5 Evaluación de usabilidad en interfaces web
- 1.6 Diferencia entre usabilidad y accesibilidad
- 1.7 Objetivo de la tesis
- 1.8 Conclusión y discusión

## Capítulo II. Estado del arte de la automatización de evaluación de interfaces de usuario

### 2.1 Herramientas existentes para interfaces WIMP

2.1.1 TkReplay

2.1.2 UsAGE

2.1.3 QUIP

2.1.4 KALDI

### 2.2 Herramientas existentes para interfaces web

2.2.1 Bobby

2.2.2 LIFT

2.2.3 DoctorHTML

2.2.4 La suite WebMetrics

2.2.4.1 La herramienta WebVIP

2.2.4.2 La herramienta VisVIP

2.2.4.3 La herramienta WebCAT

2.2.4.4 La herramienta WebSAT

2.2.4.5 La herramienta TreeDec

2.2.4.6 El formato FLUD

2.2.5 WET

### 2.3 Aspectos no cubiertos por enfoques anteriores

2.3.1 Herramientas de registro y/o reproducción de eventos de interacción del usuario con la interfaz

2.3.2 Herramientas para verificación de ajuste a normatividad

### 2.4 Conclusión y discusión

## **Capítulo III. Proceso de desarrollo**

### **3.1 Especificación**

#### **3.1.1 Contribución y relevancia**

3.1.1.1 Aportaciones en el área de interacción humano - máquina

3.1.1.2 Aportaciones en el área de interfaces adaptables al usuario

### **3.2 Diseño**

#### **3.2.1 Arquitectura general**

#### **3.2.2 Recolección de datos de acciones del usuario**

#### **3.2.3 Análisis de las acciones del usuario**

#### **3.2.4 Comparación de secuencias de eventos de interacción**

#### **3.2.5 Visualización del comparativo de secuencias de eventos de interacción**

### **3.3 Implementación**

#### **3.3.1 Módulo de captura de datos**

##### **3.3.1.1 Instalación de WebVIP**

##### **3.3.1.2 Instrumentación de las páginas de la prueba**

##### **3.3.1.3 Realización de pruebas**

##### **3.3.1.4 Generación de los datos en formato FLUD Simplificado usando CutFLUD**

#### **3.3.2 Módulo de comparación de secuencias de eventos de interacción**

##### **3.3.2.1 Propuesta de una versión simplificada de FLUD**

##### **3.3.2.2 Realización de las comparaciones con ShFLUD**

##### **3.3.2.3 El Proceso de comparación con ShFLUD**

#### **3.3.3 Módulo de visualización de comparaciones de secuencias de eventos de interacción**

#### **3.3.4 Detalles de la implementación**

##### **3.3.4.1 Lenguajes de programación**

##### **3.3.4.2 Generadores de compiladores**

##### **3.3.4.3 Herramientas en general**

### **3.4 Conclusión y discusión**

## Capítulo IV. Resultados

- 4.1 Plan de pruebas
- 4.2 Desarrollo de las pruebas
- 4.3 Interpretación de los resultados
  - 4.3.1 Caso del usuario 1
  - 4.3.2 Usuarios restantes
- 4.4 Reporte general de usabilidad
  - 4.4.1 Descripción de la prueba
  - 4.4.2. Validación de la prueba
    - 4.4.2.1. El sistema
    - 4.4.2.2. Los participantes
  - 4.4.3 Resultados
    - 4.4.3.1. Análisis de las pruebas
    - 4.4.3.2. Calificación del sistema
  - 4.4.4 Conclusiones
- 4.4 Resumen

## Capítulo V. Conclusiones

- 5.1 Automatización del proceso de evaluación de interfaces de usuario
- 5.2 Ventajas y problemáticas y del enfoque
- 5.3 Futuras direcciones de investigación

## Apéndice A. Especificación del formato FLUD Simplificado

## Apéndice B. Cuestionario de perfil de usuario

## Apéndice C. Protocolo de bienvenida

## Apéndice D. Cuestionario de usabilidad

## Apéndice E. Especificación de las tareas del usuario

## Apéndice F. Descripción de los instrumentos

## Apéndice G. Secuencias de eventos en formato FLUD Simplificado generadas durante las pruebas

## Referencias

# INTRODUCCIÓN

A pesar de ser una disciplina de reciente nacimiento, la ingeniería del software ha tenido un crecimiento considerable en décadas recientes. Las exigencias de una pujante industria global han presionado en su demanda de soluciones para la inmensa cantidad de problemas. La sociedad se hace cada día más dependiente de los progresos que le brindan las ciencias de la computación y las tecnologías de la información.

En la ingeniería del software, como en cualquier área de la ingeniería, se requiere asegurarse de que todos los productos lanzados al mercado cumplen con las expectativas de calidad que se tienen sobre ellas. Así se podrán maximizar beneficios y se minimizar riesgos.

La usabilidad es uno de los aspectos esenciales que se deben cuidar en todo sistema de software. Es necesario verificar que los usuarios puedan interactuar con el sistema de manera que puedan realizar su cometido eficientemente, de manera que sean y se sientan productivos.

Si bien el estudio de la usabilidad ya lleva cierto tiempo de iniciado, el crecimiento acelerado de Internet ha traído consigo nuevos retos a la ingeniería del software. Han surgido nuevos problemas que resolver para la comunidad de investigación en interacción humano - computadora.

Es de hacer notar el hecho de que a pesar de que los avances en el estudio de la usabilidad han sido sustanciales, aún queda mucho por crear y descubrir. Existe un amplio sector de la industria del software que todavía no percibe la urgente necesidad de incorporar estudios de usabilidad a sus procesos y metodologías de desarrollo.

La evaluación de la usabilidad de la interfaz de usuario de un sistema ha sido tradicionalmente ubicada en las etapas de verificación y validación del proceso de desarrollo del software. Ello debido a que durante ellas se valora el grado en que un sistema cumple con criterios específicos de calidad. Sin embargo es necesario mencionar que la evaluación de la interfaz no debe limitarse a dichas etapas. El descubrimiento de problemas de usabilidad en tales circunstancias acarrea un costo mayor que cuando estos son descubiertos en etapas más tempranas.

Uno de los argumentos con que se intenta justificar la ausencia de estudios de usabilidad en los proyectos de desarrollo de software es que su aplicación implica elevados costos. Esencialmente los gastos son debidos a la necesidad de contar con instalaciones especiales y personal calificado.

Es en este contexto que se ha tenido a bien elaborar una propuesta para brindar una alternativa viable a aquellas organizaciones con presupuesto o personal limitado. La idea principal tratada a lo largo de esta tesis consiste en automatizar parte de los procesos de evaluación de usabilidad, de manera que se puedan abatir tiempos y costos.



Con la finalidad de no repetir trabajo ya realizado se llevó a cabo una investigación a fin de encontrar procesos o técnicas susceptibles de automatización que hubiesen sido exploradas con anterioridad.

El trabajo estará enfocado a la automatización de los procesos de recolección de datos y análisis de las acciones del usuario, en el ámbito de las pruebas con usuarios. Esto es, se proponen técnicas para poder recopilar datos acerca de la interacción de un usuario con una interfaz, con la finalidad de brindar técnicas que analizan tales datos. La intención de ello es proporcionar al ingeniero de usabilidad elementos de decisión que le permitan emitir un veredicto acerca de la usabilidad de la interfaz.

Aunque es cierto que ya antes se han publicado trabajos que exploran las posibilidades mencionadas, estos adolecen de uno o más de los siguientes aspectos fundamentales que sí son abarcados en este trabajo:

- a) Análisis aplicable a interfaces web
- b) Herramientas de análisis que van más allá de la recopilación de datos crudos acerca de la interacción del usuario con interfaz
- c) Enfoque orientado a la posibilidad de intercambio de datos de pruebas entre colegas mediante la utilización de formatos estandarizados

Adicional a los tres aspectos mencionados, la propuesta consiste en una herramienta capaz de realizar comparaciones entre la forma en que interactuaron varios usuarios. Así son obtenidos datos que incluso pueden ser visualizados y que indican puntos de coincidencia y discrepancia entre ellos. Una utilidad inmediata de es la posibilidad de que un ingeniero de usabilidad proponga lo que a su juicio considera “la manera más adecuada de llevar a cabo una tarea con el sistema a evaluar”, y la compare con las acciones de los usuarios finales al momento de utilizar el sistema. Con esto es posible determinar un patrón de coincidencias y diferencias en que incurren los usuarios. Se podría con ello demostrar la hipótesis del ingeniero, o por el contrario, evidenciar fallas en el diseño de la interfaz o de la prueba.

El presente trabajo de tesis se ha dividido en cinco capítulos. En el primero de ellos se pone en contexto la investigación, presentándose las definiciones necesarias, y dando una descripción general de los distintos métodos de evaluación de usabilidad que existen. En el segundo capítulo se presenta el estado del arte en lo relacionado a la evaluación automatizada de interfaces de usuario, mencionando varios enfoques existentes, en conjunto con sus características, beneficios y desventajas. En el tercer capítulo se describe el proceso de desarrollo de una serie de técnicas y herramientas que son el aporte fundamental de esta tesis, mencionando principalmente los detalles tomados en cuenta durante su especificación, diseño e implementación. El capítulo cuarto explora la aplicación de las ideas surgidas en este trabajo a un estudio de usabilidad real, probando de esta manera que es viable su utilización en entornos industriales. Por último, el quinto capítulo menciona las conclusiones derivadas de los puntos tratados a lo largo de todo el trabajo de tesis, indicando los beneficios y desventajas del enfoque tomado. A su vez se ofrecen líneas de investigación complementarias. De manera adicional se incluyen apéndices en que se colocan documentos relevantes.

# Capítulo I

## Antecedentes de la automatización de la evaluación de interfaces de usuario

Contrario a una idea generalizada, el diseño y evaluación de la interfaz-usuario de un sistema es uno de los aspectos esenciales de su desarrollo y no sólo una cuestión cosmética. En efecto, existe una gran cantidad de profesionales que piensa que diseñar interfaces-usuario es un aspecto que se da naturalmente, tan sólo por desarrollar software y usar el sentido común [TULLIS 1993]. Desafortunadamente no hay nada más alejado de la realidad: el diseño de una interfaz-usuario es un proceso iterativo, que requiere de modelos y procesos formales. En ese proceso es necesario evaluar con usuarios su correcto funcionamiento y usabilidad. "Usabilidad", o "desarrollo centrado en el usuario", son términos que cada vez más se encuentran en la literatura de la ingeniería de software. Sin embargo, para efectos de claridad, en el presente documento hemos decidido introducir el término. Esto se hace a continuación.

### 1.1 DEFINICIÓN DE USABILIDAD

El concepto de usabilidad es definido por el ISO (International Organisation for Standardisation, Oficina Internacional para la Estandarización) de la siguiente forma:

"El grado en que un producto puede ser usado por usuarios específicos para lograr metas específicas con efectividad, eficiencia y satisfacción en el contexto específico de uso." [ISO 1992]

Es necesario precisar algunos de los conceptos mencionados en la definición anterior, provenientes de la misma fuente:

- **Efectividad.** Exactitud y completitud con la cual los usuarios logran metas específicas. Esto es, la medida en que los usuarios logran cumplir sus metas completamente o parcialmente.
- **Eficiencia.** Recursos usados en relación con la exactitud y completitud con la cual los usuarios logran metas específicas. En general se puede describir como el gasto de recursos del usuario para lograr metas específicas: tiempo, esfuerzo intelectual, etc.
- **Satisfacción.** Libertad de confort o falta de confort, y actitud positiva para usar el producto. Es decir, la impresión que el uso del producto deja en el usuario: le hace sentir productivo, seguro, eficiente, etc.; o por el contrario, le ocasiona estrés, ansiedad, inseguridad etc.

Por otra parte Jakob Nielsen -una de las autoridades en la materia- enuncia su definición de usabilidad de la siguiente manera [NIELSEN 2003]:

*“La usabilidad es un atributo de calidad enfocado a observar la facilidad de uso de las interfaces de usuario. La palabra usabilidad también se refiere a los métodos para mejorar la facilidad de uso durante el proceso de diseño”*

Es de hacer notar el hecho de que el término definido en ambos casos - redactados en inglés- es *usability*. En español aún no se ha aprobado por la Real Academia de la Lengua Española alguna traducción o palabra equivalente que tenga toda la amplitud expresiva de aquella palabra. Sin embargo, y debido a que el término *usabilidad* ha tenido una amplia aceptación entre la comunidad que realiza investigación principalmente en el área de interacción humano – máquina, se ha elegido utilizar el término a fin de simplificar la disertación que se hace a lo largo de todo este trabajo de tesis.

Nielsen [NIELSEN 2003] menciona cinco componentes cualitativos de la usabilidad, a saber:

- **Aprendizaje.** La facilidad con que los usuarios pueden realizar sus tareas básicas la primera vez que se utilizan el diseño.
- **Eficiencia.** Una vez que los usuarios han aprendido el diseño, ¿qué tan rápido pueden realizar sus tareas?
- **Memorización.** Cuando los usuarios retornan al diseño después de un periodo de no tener contacto con él, ¿qué tan fácil es para ellos volver a tener su nivel de eficiencia?
- **Errores.** ¿Cuántos errores cometen los usuarios, qué tan severos son esos errores, y con qué facilidad se pueden recuperar de los errores?
- **Satisfacción.** ¿Qué tan placentero es para los usuarios el uso del diseño?

Los puntos antes mencionados forman los pilares de la usabilidad, por lo que los diseñadores de interfaces e ingenieros de usabilidad no deben permitirse pasarlos por alto a menos de que busquen complicar las tareas de sus usuarios.

## 1.2 EVALUACIÓN DE USABILIDAD

Cada una de las personas que trabajan de una forma u otra con una interfaz, ya sea durante su creación o su uso, tienen lo que se llama un *modelo cognitivo* de ella, es decir, las cosas que asumen y conocen acerca de la interfaz [GILLAN 1990]. Tómese en cuenta lo siguiente:

- Cuando un diseñador construye una interfaz, en su mente se lleva a cabo todo un proceso creativo, que lo lleva a mecanizar e integrar a su legado de conocimiento todas las ideas que va plasmando. Así, cuando ha terminado de crear la interfaz, cree estar seguro de que es la adecuada debido a que

desde su particular punto de vista ya ha cumplido con todos los requisitos funcionales que se planteaban. De esta manera, puede darse el caso de que el modelo de la interfaz no sea el mismo para el diseñador que para el usuario, siendo esta situación la regla más que la excepción.

- El punto de vista del programador es aún más disparatado con respecto al del usuario, ya que el programador conoce el sistema profundamente, y lo concibe desde un punto de vista técnico, conociéndolo aún en sus vicisitudes más intrincadas.
- Los usuarios del sistema, por otro lado, prefieren en general ignorar los detalles acerca del funcionamiento interno de las aplicaciones que usan.

El problema de la diversidad de modelos cognitivos acerca de la interfaz estriba en que el único aspecto que será relevante al usuario final será precisamente la interfaz y la forma en que puede llegar a manejarla para poder tener una productividad óptima. El programador podrá pensar en el sistema como un conjunto de algoritmos, y el diseñador podrá sentirse orgulloso de su técnica de manejo del color o de la tipografía empleada. Sin embargo, estos aspectos impactarán poco en el desempeño del usuario si el grupo de desarrollo no se ha preocupado por la usabilidad.

Por lo enunciado anteriormente se hace evidente la necesidad de verificar que en un producto determinado las ideas y puntos de vista asumidos por cada uno de los implicados coincide con el de los otros, de manera que al no haber diferencias sustanciales relativas a su concepción de la interfaz, se reduce la probabilidad de que existan problemas de usabilidad en la interfaz. Para poder satisfacer dicha necesidad de verificación los expertos han desarrollado técnicas para la evaluación de interfaces de usuario, que se definen como sigue:

*"Metodologías para la medición de los aspectos de usabilidad de la interfaz de usuario de un sistema y para la identificación de problemas específicos. La evaluación de usabilidad es una parte importante del proceso del diseño de la interfaz de usuario, que consiste de ciclos iterativos de diseño, elaboración de prototipos y evaluación." [DIX 1998; NIELSEN 1993]*

La evaluación de usabilidad es un proceso en el que se requieren varias actividades, que dependen del método elegido; algunas de ellas son [IVORY 2001]:

- Captura. Compilación de datos de usabilidad, como pueden ser tiempo en que se completan las tareas, errores, violaciones de guías de diseño, y valoración subjetiva.
- Análisis. Interpretación de los datos de usabilidad para identificar problemas de usabilidad en la interfaz.
- Crítica. Elaboración de sugerencias o mejoras para mitigar problemas.

El aspecto medular de esta tesis se encuentra en el ámbito de la evaluación de usabilidad. El enfoque principal es generar una propuesta para la automatización de este proceso, con especial interés en las dos primeras actividades: captura y análisis.

El proceso de ingeniería de usabilidad es enriquecido por una amplia gama de conocimientos provenientes de diversas áreas como son: factores humanos, ergonomía, ciencia cognoscitiva, psicología, etc. Cada una de estas áreas tiene un interés bien definido, y cuenta con su propia comunidad de investigación. El presente trabajo de tesis no pretende ser un tratado acerca de cada aspecto de la usabilidad. Se limita su alcance al campo de la evaluación de interfaces de usuario, enfocándose especialmente en generar una propuesta para la automatización de este proceso. Se pone especial interés en las dos primeras actividades mencionadas por Ivory: captura y análisis.

## **1.3 MÉTODOS DE EVALUACIÓN DE USABILIDAD**

La evaluación de usabilidad puede realizarse usando varios métodos, que pueden ser clasificados en tres grandes categorías, cada uno con sus aspectos positivos y negativos, orientados a metas distintas [NIELSEN 1993a]:

- Métodos de inspección
- Métodos de análisis
- Pruebas empíricas

Existen otro tipo de pruebas que son también pilares importantes para la labor del ingeniero de usabilidad, no mencionados en la lista anterior, los llamados métodos de indagación directa de la experiencia del usuario [DE MARSICO 2003; PEARROW 2000].

La aplicación de cada una de las pruebas de cualquiera de las categorías mencionadas no excluye a otras. No hay una técnica que proporcione toda la información requerida, sino que en lugar de ello, los resultados de ellas suelen ser complementarios. De igual forma, no es forzoso que una prueba lleve a cabo una sola vez y para siempre. Lo adecuado consiste en un proceso recurrente en el que se van detectando problemas de usabilidad e implementado mejoras a lo largo del ciclo de vida de la interfaz.

Es posible que en etapas tempranas de la existencia del sistema, se pueda trabajar con prototipos, es decir, con bosquejos semifuncionales de la interfaz. Sobre un prototipo se pueden comenzar a realizar algunas pruebas antes de que la versión final de la interfaz quede terminada. La ventaja de esto se aprecia en la detección temprana de los errores y fallas de usabilidad, de manera que el impacto de las correcciones sobre un diseño final se vea minimizado.

### **1.3.1 Métodos de inspección**

Se utilizan las habilidades de uno o más especialistas de usabilidad para inspeccionar la interfaz con la intención de generar una lista de potenciales problemas de usabilidad. Algunos de los métodos más conocidos pertenecientes a esta categoría son:

- Inspección de estándares y consistencia [UPA 2005]
- Evaluación heurística [NIELSEN 1992]

La ventaja inmediata de estas técnicas es que son más baratas y rápidas, en comparación con otras, y se pueden usar en cualquier momento del ciclo de desarrollo. Sin embargo, al no emplear usuarios reales, es difícil evaluar el desempeño de los usuarios.

#### 1.3.1.1 Inspección de estándares y consistencia

Aplicado a interfaces web, que son a las que se enfoca este trabajo de tesis, también puede ser llamado análisis del código fuente de páginas web.

Esta técnica se enfoca en analizar no al usuario ni a su comportamiento, sino a la interfaz en sí. Se revisa el código que ejecuta o describe la interfaz (por ejemplo código de un lenguaje de etiquetas, como HTML para una interfaz web), verificando que cumpla con ciertas guías y recomendaciones de estilo ampliamente aceptadas.

Contrario a otras técnicas descritas en esta sección, el análisis de código fuente de páginas web presenta oportunidad excelente para implementar la automatización del proceso de evaluación de interfaces de usuario. La realización manual de este tipo de pruebas es una tarea laboriosa, propensa a errores y omisiones. Por otra parte, su automatización mediante el uso de alguna herramienta resulta ser un proceso casi mecánico que requiere poco esfuerzo.

#### 1.3.1.2 Evaluación heurística

Esta es también una técnica valiosa para comenzar el proceso de verificación de la usabilidad de una interfaz. Una heurística es una regla o recomendación muy aceptada.

Una evaluación heurística es realizada por un experto que analiza un sitio y lo califica en base a una lista estandarizada de heurísticas. La implementación de esta técnica tiene un costo relativamente bajo, y requiere una preparación poco compleja, debido a que la lista de heurísticas de usabilidad es la misma para muchas situaciones.

Una vez terminada la evaluación heurística, la persona que la haya llevado a cabo emite un reporte con las deficiencias que encontró, y un listado de sugerencias que describen los cambios que deben llevarse a cabo en la interfaz.

### 1.3.2 Métodos de análisis

También son implementados por expertos en usabilidad, que elaboran la simulación de un usuario usando el modelo propuesto. Las dos técnicas más populares de esta categoría son:

- Análisis cognitivo [RIEMAN 1995]
- GOMS [CARD 1983]

Cada técnica tiene sus objetivos propios, como puede ser la evaluación de la eficiencia de la interfaz en el caso de GOMS, o la determinación de la facilidad de aprendizaje de un sistema usando el Análisis Cognitivo.

La aplicación de estos métodos, al igual que con los métodos de inspección, resultan ser barata y rápida, durante cualquier fase del ciclo de desarrollo. Si embargo se requiere contar con especialistas con amplia experiencia, con un entendimiento profundo de los usuarios y sus tareas.

#### 1.3.2.1 Análisis cognitivo

El análisis cognitivo es una técnica de evaluación de interfaces de usuario. Es llevada a cabo por uno o más evaluadores que inspeccionan una interfaz de usuario guiados por una lista de tareas, evaluando básicamente la facilidad de aprendizaje y de comprensión de la interfaz. Por lo general la interfaz bajo revisión es presentada en forma de un prototipo impreso o dibujado en papel, aunque no se descarta su uso en interfaces totalmente terminadas. Para poder llevar a cabo el estudio, además de la lista de tareas es necesario tener a la mano el perfil de usuario especificando el conocimiento que posee el usuario acerca del dominio de la aplicación [RIEMAN 1995].

Como parte de la realización de una prueba de este tipo, es responsabilidad de los evaluadores dar respuesta a cuatro cuestiones esenciales:

- ¿El usuario tiene una meta clara? Es decir, indicar si el usuario sabe qué es lo que desea hacer.
- ¿El usuario puede darse cuenta de que la opción correcta está disponible? Esto es, verificar si el usuario puede ver o entender las opciones con las que cuenta para interactuar.
- ¿El usuario puede asociar las posibles acciones a tomar con los efectos que desea obtener?
- ¿Si el usuario ha elegido y realizado la acción correcta, se puede dar cuenta del progreso que está haciendo para terminar su tarea?

Una vez que se han respondido las preguntas anteriores se tendrán elementos para poder emitir diversas consideraciones relativas a la usabilidad de la interfaz, y proponer mejoras.

El análisis cognitivo es una técnica muy valiosa durante etapas tempranas del proceso de desarrollo del software. Es posible utilizarla repetidamente sobre distintos prototipos que van incorporando y acumulando paulatinamente las mejoras propuestas. La detección oportuna de problemas de usabilidad evita su posposición hasta etapas en que su descubrimiento y corrección acarrearía un costo mucho mayor.

#### 1.3.2.2 GOMS

Más que una técnica de evaluación de interfaces de usuario, GOMS es un modelo que describe el conocimiento que un usuario debe tener para poder llevar a cabo tareas en un dispositivo o sistema [KIERAS 1999].

GOMS fue propuesto por primera vez en 1983 por Card, Moran y Newell [CARD 1983]. Las siglas GOMS significan metas, operadores, métodos y reglas de selección (del inglés Goals, Operators, Methods, and Selection Rules). Básicamente

un modelo GOMS consiste de la descripción de los métodos requeridos para llevar a cabo las metas establecidas.

Los modelos GOMS suelen utilizarse para predecir la usabilidad de una interfaz de la siguiente manera: después de la realización de un análisis de la tarea preliminar y de la elaboración de un primer diseño de la interfaz, el diseñador usará un modelo aplicable para encontrar problemas de usabilidad en la interfaz.

### 1.3.3 Pruebas empíricas

Se realizan mediante la observación cuidadosa de usuarios intentando llevar a cabo una serie de tareas utilizando el sistema bajo prueba. Generalmente las sesiones de los usuarios son grabadas en video para poder realizar medidas del tiempo tomado para cada tarea, y para permitir una posterior observación detallada [PEARROW 2000]. A pesar de sus excelentes resultados, su costo de aplicación puede resultar gravoso para organizaciones con presupuesto limitado. Son conocidas en la literatura bajo el concepto de evaluación de comportamiento o evaluación clásica de usabilidad.

Este tipo de pruebas se basan en una larga tradición de investigación etológica y psicológica. Su principal característica consiste en la observación del usuario mientras realiza las tareas que le han sido asignadas para la prueba. A partir de las observaciones se elaboran conclusiones acerca de las cualidades y defectos de la interfaz.

A pesar de la complejidad que presenta su realización, esta es una práctica muy difundida entre los expertos investigadores del área de Interacción Humano Máquina, por la gran cantidad de información valiosa que proporciona.

Cuando una organización ha decidido llevar a cabo este tipo de evaluación, ha de seguir toda una metodología para lograr obtener el máximo beneficio de los recursos invertidos. A continuación se exponen los pasos a seguir.

I.- Infraestructura.- Lo primero que se requiere es tener un laboratorio, que como mínimo cuenta con los siguientes muebles y equipo:

- Una estación de trabajo con una configuración de software y hardware similar a la que se presume tendrán los usuarios de la interfaz
- Una silla para el participante
- Una silla para el moderador de la prueba
- Una silla para el encargado de cronometrar la prueba(opcional)

El laboratorio también puede estar equipado con una videograbadora, micrófonos, y equipos de comunicaciones, para realizar monitoreo desde una sala de control. En la figura I.1 se muestra la configuración común de un laboratorio de pruebas sencillo. Se pueden observar los diferentes elementos de infraestructura necesarios para la realización de pruebas, recientemente listados.

II.- Plan de trabajo.- Para poder llevar a cabo la prueba, debe haber un plan de trabajo establecido, en el que se tengan en cuenta los siguientes aspectos:



### a) Propósito

Se describe qué aspecto de la interfaz se está probando, aclarando puntos como: ¿Qué hace el programa? ¿Hacia quién está dirigido el programa? ¿Cuál es la visión del sistema, programa o sitio web? ¿Por qué razón se está realizando la evaluación de usabilidad? La utilidad de lo aquí mencionado estriba en aclarar los objetivos del sistema bajo prueba. Se busca llegar a una puesta en común con el cliente, con la finalidad de evitar confusiones y malos entendidos debidos a puntos de vista distintos, o ideas mal concebidas.

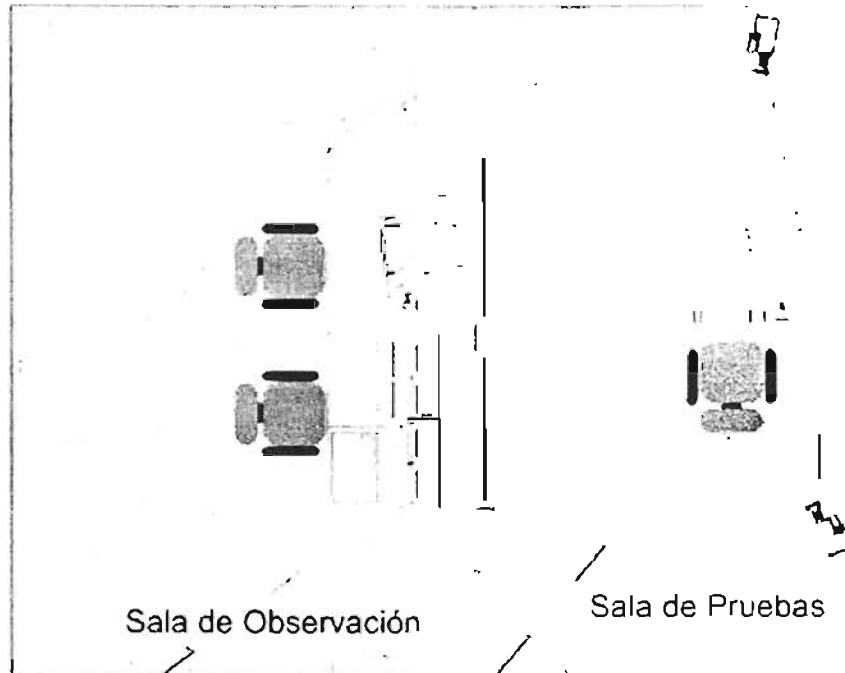


Figura I.1 Laboratorio de pruebas de usabilidad

### b) Planteamiento del problema

Se enuncia de manera corta el tipo de cosas que se desea resolver, resumiendo en una o dos líneas

### c) Perfiles de usuario

Se caracteriza al grupo de usuarios que forman parte de la población, tomando de entre ellos solamente una muestra representativa; sobre esta muestra se elegirán las personas que presentarán la prueba.

### d) Metodología

Es una explicación de los pasos a llevar a cabo durante las sesiones de la prueba, sin llegar al nivel de detalle en que se describa cada una de las tareas que han de llevarse a cabo.

#### e) Probar el ambiente y equipo de prueba

Este paso tiene la finalidad de detectar posibles inconvenientes que se podrían presentar durante las sesiones en que se esté llevando a cabo la prueba. Del mismo modo, incluir una clara descripción de este ambiente dentro del reporte final será importante si se desea replicar el ambiente en algún momento.

#### f) Personal de la prueba

Definitivamente es imposible llevar a cabo una prueba de este tipo con sólo una persona realizando todas las actividades requeridas. La lista de potenciales miembros del equipo de trabajo puede incluir los siguientes:

- Monitor de pruebas. La persona que interactúa con los participantes, modera la sesión, y lee el guión bajo el que estarán trabajando los usuarios.
- Registrador de datos. Recaba por escrito toda la información y resultados luego de que los participantes realizan sus tareas.
- Encargado del cronómetro. Mide el tiempo que ha tomado a los participantes completar las tareas, e indica en qué momento se les ha agotado el tiempo.
- Operador de video. Maneja la cámara de video, haciéndose cargo de colocar y editar las cintas o dispositivos de almacenamiento.
- Especialista del sitio, programa o sistema. Una persona que conoce todos los detalles relativos al sistema que se está probando. Tiene autoridad para juzgar si en usuario ha tenido éxito o fracaso en tareas cuestionables.

Es necesario dar al personal una explicación detallada de las labores a que han sido asignados, y de asegurarse de que la hayan entendido completamente antes de iniciar con las sesiones con usuarios.

#### g) Métricas de Evaluación

Aquí se debe explicar los métodos mediante los cuales se determinará el éxito o fracaso para cada tarea, así como las restricciones que deben respetar los participantes. Algunos criterios comunes suelen ser:

- Número de clics que el usuario requirió para concluir una tarea.
- Ruta de clics tomada por el usuario, que es difícil de tomar, a menos que se utilice algún tipo de software especializado.
- Tiempo total transcurrido. Se puede especificar un límite de tiempo máximo que un usuario puede tomarse para cumplir con una tarea.

#### h) Lista de Tareas

Se presentan detalladamente las tareas que se han elegido para intentar obtener respuestas a las problemática expresada en apartados anteriores. Cada tarea tiene varios elementos:

- Guión o texto. Un breve texto que instruye al usuario acerca de la tarea que debe llevar a cabo. Este texto es leído en voz alta al usuario, y es de vital importancia que sea corto y fácil de recordar.
- Condiciones de éxito. Cada tarea debe tener una meta no ambigua.

- Estado inicial. Se especifica el estado en que se encuentra la computadora y si es necesario también la sesión de usuario cuando la tarea empieza.

#### i) Resultados

Esta sección se agregará hasta que se hayan finalizado las pruebas de usabilidad. Generalmente en esta sección se muestra un resumen que contiene indicadores cuantitativos acerca de las tareas: cantidad de éxitos y fracasos, tiempo transcurrido, número de clics. Los indicadores pueden ser expresados en porcentajes o promedios.

#### j) Discusión

Esta es una sección de formato libre, en la que se pueden anotar observaciones, comentarios hechos por los participantes a lo largo de la prueba.

#### k) Recomendaciones de cambios

No se puede negar la importancia de esta sección, pues el fin último de las pruebas es precisamente encontrar una forma de corregir los problemas que pueda presentar la interfaz. En esta sección se enuncian los cambios que se considera deben ser implementados como resultado de la información recabada.

#### l) Apéndices y agregados

En esta sección se adjuntan documentos que sean o hayan sido relevantes para las pruebas.

Como se pudo notar a lo largo de esta sección la complejidad de este tipo de pruebas radica no en la forma en que se han de procesar los datos, sino en la forma en que se obtendrán. La limitante estriba en la necesidad de un ambiente costoso y complejo para la realización de este tipo de pruebas. Afortunadamente, la posibilidad de automatizar alguna o varias fases de este proceso ya ha sido explorada en varias ocasiones. En secciones posteriores se describen algunos enfoques propuestos.

### **1.3.4 Métodos de indagación directa de la experiencia del usuario**

Este tipo de evaluación de usabilidad goza de amplia aceptación entre los ingenieros de usabilidad, en parte debido a la experiencia positiva acumulada por la comunidad de investigación en interacción humano - computadora. Las dos principales técnicas de esta categoría son:

- Cuestionarios a usuarios
- Entrevistas

#### **1.3.4.1 Entrevistas**

La entrevista es un diálogo interpersonal directo cuyo tema es la interfaz en cuestión. Por lo general la entrevista va dirigida por una serie de preguntas o tópicos elaborados por el moderador de la prueba, siendo el entrevistado un usuario de la

interfaz. Durante el intercambio de ideas se pueden tratar asuntos como son: expectativas del usuario, necesidades, pensamientos, experiencias previas, y cualquier otro aspecto relevante [PEARROW 2000].

Debido a la naturaleza de esta prueba, difícilmente puede pensarse o plantearse un proceso de automatización para ella. Gran parte del éxito en su aplicación de debe al talento que muestre el entrevistador para dirigir al entrevistado, a fin de obtener información de gran calidad y relevancia.

#### 1.3.4.2 Cuestionarios a usuarios

Esta es una de las prácticas más típicas y extendidas. Se componen de una serie de preguntas dirigidas a obtener un modelo de la percepción que tienen el usuario acerca de la interfaz, que generalmente son de alto nivel.

Aspectos vitales para la obtención de resultados que sean de utilidad (que sean exactos y objetivos) para el realizador de la prueba son:

- Adecuada elección de las preguntas que ha de responder el usuario, justificando cada una de ellas con un punto específico que desea aclarar o resolver.
- La elección de un formato de pregunta que proporcione el indicador que se espera: preguntas abiertas para mediciones puramente cualitativas, preguntas de opción múltiple para cuantificar algún aspecto de la interfaz, etc.
- La correcta redacción de las preguntas, ya que un planteamiento tendencioso podría arrojar resultados poco objetivos.
- Elección de la forma en que se van a calificar o interpretar los resultados: generación de estadísticas, gráficas, reportes, informes, sugerencias, etc. Las preguntas abiertas pueden arrojar información muy valiosa, pero suelen ser difíciles de interpretar y cuantificar.

Existe una serie de cuestionarios que cuentan con amplia aceptación entre la comunidad de expertos evaluadores de interfaz. Una recopilación de los más importantes se encuentra en la tabla I.1 [PERLMAN 2001].

La aplicación de cuestionarios no exige que estos se encuentren impresos en papel. Las respuestas de los cuestionarios bien pueden provenir de lugares distantes de la tierra usando algún medio de telecomunicación, ya que muchos de los sistemas actuales no se centran en usuarios de una sola latitud. Antes bien reciben accesos incluso de los lugares más remotos del mundo.

La aplicación automatizada de este tipo de prueba no presenta mayor complicación que la de realizar una aplicación común que muestra las preguntas de la batería. Colecta las respuestas, con la posibilidad de emitir estadísticas y reportes de los datos obtenidos.

<b>Acrónimo</b>	<b>Instrumento</b>	<b>Referencia</b>	<b>Institución</b>
QUIS	Questionnaire for User Interface Satisfaction	Chin et. al. 1988	Maryland
PUEU	Perceived Usefulness and Ease of Use	Davis 1989	IBM
NAU	Nielsen's Attributes of Usability	Nielsen 1993a	BellCore
NHE	Nielsen's Heuristic Evaluation	Nielsen 1993b	BellCore
CSUQ	Computer System Usability Questionnaire	Lewis 1995	IBM
ASQ	Alter Scenario Questionnaire	Lewis 1995	IBM
PHUE	Practical Heuristics for Usability Evaluation	Perlman 1997	OSU
PUTQ	Purdue Usability Testing Questionnaire	Lin et. al. 1997	Purdue

Tabla 1.1. Cuestionarios con amplia aceptación entre la comunidad de expertos evaluadores de interfaces

#### **1.4 Problemática de las técnicas no automatizadas de evaluación de interfaces**

Las técnicas que comúnmente se usan para asegurar la usabilidad generalmente son llevadas a cabo por un experto que posee el conocimiento adecuado para su planeación y realización exitosa. Sin embargo, la cantidad de expertos que existen es mucho menor a la deseable para una industria del software en pleno crecimiento [NEWMAN, 1995].

Como es de suponerse, la contratación de un experto que obtenga y analice manualmente la interacción de los usuarios mediante pruebas de laboratorio es un proceso lento y costoso. Por esa razón, la comunidad académica de Interacción Humano-Computadora (IHC) se ha dado a la tarea de generar herramientas que si bien no automatizan completamente dicho proceso de evaluación, si facilitan la compilación de información y la presentación de resultados. Esto se hace de modo tal que un desarrollador, con algunos conocimientos de diseño de interfaces, sea capaz de hacer sus propios diagnósticos.

En este contexto, uno de los enfoques más trabajados consiste en automatizar el análisis de la interacción de un usuario con un sistema, comparando sus acciones con respecto a una interacción definida como "óptima". Esto es, cuando un usuario utiliza un sistema y recorre sus diferentes funciones, ¿qué tan bien lo hace? Esto es susceptible de ser medido si el equipo de desarrollo establece cuál es la mejor secuencia de acciones y a partir de ella calcula la distancia entre las acciones realizadas por el usuario y las establecidas. Una interfaz en la que el usuario sigue de manera autónoma la ruta óptima de acciones, es una interfaz que puede suponerse bien diseñada. Es decir, existe evidencia de que el usuario encuentra su lógica de funcionamiento y organización fácilmente entendible. Por el contrario, una navegación errática indica problemas por parte del usuario para poder comprender la lógica de funcionamiento.

Este tipo de sistemas de evaluación automática se ha vuelto de especial interés para la web en donde. Por una parte, los usuarios pueden abandonar un

sistema en cualquier momento, de modo que una interacción sencilla y eficiente es de primera importancia; Por otro lado, un sitio se puede ver como un lugar de experimentación de bajo costo, en el sentido de que queda abierto a la interacción de miles de usuarios.

Para la evaluación de este tipo de sistemas se acostumbra realizar el análisis de los registros de eventos generados durante la interacción del usuario con el programa. Desafortunadamente la cantidad de información a revisar es muy grande, pues por lo general se toma un evento de interacción como un evento de bajo nivel (como pueden ser movimientos del ratón, pulsaciones de teclas). Esto lleva al verdadero problema: encontrar una manera de abstraer toda esa gran cantidad de información, de modo que el análisis se haga sobre aspectos de alto nivel (por ejemplo tareas a desarrollar), y no los movimientos del cursor. La Figura 1.2 muestra una categorización de eventos en base a su semántica: de bajo, medio o alto nivel semántico.



Figura 1.2 Jerarquía de eventos generados durante la interacción de un usuario con un sistema

Como puede observarse en la figura 1.2, se plantea un sistema que agrupe eventos de bajo nivel para posteriormente identificarlos como un paso concreto con un significado. A su vez agrupa estos pasos para tratar de identificar que tarea está haciendo el usuario, y en base a esa identificación, calcular la distancia entre los pasos óptimos, y los realizados por el usuario.

## 1.5 Evaluación de interfaces web

Las interfaces de usuario para sitios web tienen características especiales que las diferencian profundamente de las interfaces WIMP (*windows, icons, pointer and mouse*; ventanas, iconos, apuntador y ratón).

Las diferencias vienen dadas desde el mismo paradigma de interacción. Comúnmente las interfaces web se encuentran limitadas en un modelo de interacción impuesto por el protocolo de comunicaciones entre el cliente y el servidor. Un límite adicional es la simplicidad que impone el HTML como lenguaje para creación de interfaces, con su reducido conjunto de elementos de interacción (cuadros de texto, cuadros de verificación, botones de radio, cuadros de lista, hiperenlaces, botones de envío de información, los controles de manejo de ventanas proporcionados por el agente de usuario o navegador).

Es por ello que se han desarrollado técnicas de evaluación de interfaces tipo web que se orientan a las necesidades específicas de los usuarios de estas.

Es de notar que las diferencias entre las interfaces web y las WIMPS no son abismales. A pesar de que existe una gran cantidad de puntos de divergencia entre ellas, los aspectos comunes hacen que las mismas técnicas de evaluación puedan aplicarse a ambos tipos de interfaces. Se conserva la esencia de la prueba, pero variando los detalles de la implementación.

## 1.6 Diferencia entre usabilidad y accesibilidad

Aunque ambos conceptos pudieran parecer similares en primera instancia, no lo son ya que se considera a la accesibilidad como un pilar fundamental de la usabilidad.

La accesibilidad consiste en asegurarse de que el contenido es accesible, o sea, asegurarse de que el contenido puede ser navegado y leído por cualquiera, sin importar su localización, experiencia, o el tipo de tecnología computacional que usa. Popularmente la accesibilidad se discute en relación a las personas con discapacidades, porque es un grupo particularmente propenso a estar en desventaja si se dejan de lado los principios de la accesibilidad [PEARROW 2000].

Actualmente el W3C (World Wide Web Consortium, Consorcio de la World Wide Web) impulsa un proyecto llamado Web Accessibility Initiative [W3C 2005] – iniciativa de accesibilidad en web, en español-, mejor conocido por sus siglas en inglés: WAI. Las metas principales de este esfuerzo son las de establecer una serie de guías que se pretende sean usadas por los creadores de páginas web. Las guías, contenidas en un documento llamado Web Content Accessibility Guidelines 1.0 [W3C 2005b] (guías para la accesibilidad de contenido web) son descritas de la siguiente forma:

*“La meta primaria de estas guías es promover la accesibilidad. Sin embargo, ellas también mejorarán la disponibilidad del contenido web para todos los usuarios, sin importar qué agente usen (por ejemplo, navegador web, navegador web parlante, teléfono móvil, computadora personal dentro de un vehículo, etc.) o las limitaciones bajo las que operen (por ejemplo, ambiente ruidoso, salas poco iluminadas o iluminadas en demasía, un ambiente en el que no se deban usar las manos, etc.).”*

Sin embargo, el documento en cuestión no define el término accesibilidad, aunque sí da la definición de contenido accesible, indicándolo como sigue:

*“El contenido es accesible si puede ser usado por alguien con alguna discapacidad”*

## **1.7 Objetivo de la tesis**

Se desarrollará un entorno de trabajo mediante el cual sea posible la captura de eventos de la interacción del usuario con un sistema web en un formato estándar. Habiendo dispuesto de estos datos, se propondrá un esquema de comparación automatizada de secuencias de eventos, de manera que se pueda emitir información acerca de la similitud entre ellas. Las secuencias de eventos a comparar podrán provenir directamente de la interacción de un usuario con una interfaz, o bien ser definidas expresamente por un ingeniero de usabilidad.

## **1.8 Conclusión y discusión**

En el presente capítulo se han introducido los conceptos básicos sobre los que se discurre a lo largo de la tesis. Se ha presentado al lector una descripción somera de los aspectos esenciales de la usabilidad y de la evaluación de usabilidad tal y como se lleva a cabo actualmente por los expertos.

En subsecuentes capítulos se exploran las oportunidades de automatización de algunas formas de la evaluación de usabilidad. Se habrán de describir enfoques que incluso ya son aplicados en la industria del software, y se explorarán las posibilidades de innovación y mejora de ellos.



## Capítulo II

# Estado del arte de la automatización de evaluación de interfaces de usuario

El presente trabajo, como casi cualquier trabajo de investigación actual, no existe en un vacío, se fundamenta y toma referencias de otros enfoques que ya han sido probados y que han demostrado sus cualidades y deficiencias. A continuación se mencionarán algunos de ellos.

### 2.1 Herramientas de evaluación automatizada de interfaces WIMP

A lo largo de esta sección se hará un breve recuento de los enfoques que ya han sido tomados por diferentes investigadores para plantear esquemas de automatización de evaluación de interfaces de usuarios del tipo WIMP (*windows, icons, pointer and mouse*; ventanas, íconos, apuntador y ratón).

#### 2.1.1 TkReplay

TkReplay es una herramienta desarrollada por Charlie Crowley, del Departamento de Ciencias de la Computación de la Universidad de Nuevo México [CROWLEY, 2005].

Su funcionalidad básica es la de registrar el flujo de eventos de la interacción de un usuario con una interfaz desarrollada con el marco de trabajo Tcl/Tk, que permite desarrollar interfaces gráficas de usuario que funcionan en múltiples plataformas. Habiendo registrado los eventos, se puede hacer una reejecución de ellos, de manera que quien esté realizando la prueba puede ver nuevamente las acciones realizadas por el usuario en una sesión.

El enfoque que explícitamente se ha indicado para TkReplay es el de la ejecución de pruebas de regresión. En pocas palabras las pruebas de regresión pueden ser pensadas como aquellas que nos sirven para asegurarnos de que una vez que tal prueba se ha aplicado satisfactoriamente a un programa, se ha garantizado que tiene el comportamiento deseado. De esa manera se considera que llegado el momento en que se han de introducir cambios, se pueden aplicar nuevamente las pruebas. La finalidad de las pruebas es saber si las modificaciones han afectado nocivamente el comportamiento del programa.

Una de las limitaciones que presenta esta herramienta, y que no es privativa de ella, sino que se extiende a todas las herramientas de este tipo que se han realizado, es que no utiliza un formato de registro de eventos que esté pensado para ser estándar. Tampoco usa un modelo de eventos consistente con otros marcos de trabajo. Esto se debe a las diferencias que existen entre la forma en que tienen tantas similitudes como diferencias wxWindows [WXWINDOWS, 2005], XWindow [XWINDOW, 2005], Qt [TROLLTECH QT, 2005], AWT [SUN AWT, 2005], Swing [SUN SWING, 2005], Tcl/Tk [Tcl/Tk, 2005], etc. Las diferencias van desde el modelo

de generación, captura y tratamiento de eventos, hasta el nivel de abstracción en que se desenvuelven y las plataformas en las que funcionan.

### 2.1.2 UsAGE

Esta herramienta fue desarrollada por Dana L. Uehling del NASA / Goddard Space Flight Center y Karl Wolf de Century Computing, Inc. A diferencia de TkReplayer, UsAGE (User Action Graphing Effort) si fue desarrollado teniendo en mente la realización de pruebas de usabilidad automatizadas.

UsAGE es un programa que en esencia realiza la captura de los eventos de la interacción entre el usuario y el sistema. La finalidad de ello es poder realizar no sólo la reejecución posterior de ellos, sino una visualización de las diferencias existentes entre el comportamiento presentado por dos usuarios diferentes. Los grupos de eventos comparados bien pudieran ser los datos de un experto contra los de un usuario novato. La figura II.1 tomada de [UEHLING, 1995] es una muestra de las capacidades de ese programa, que consisten básicamente en acomodar las secuencias de eventos del usuario como del experto, para poder comparar visualmente las coincidencias y diferencias que existen entre ambas secuencias:

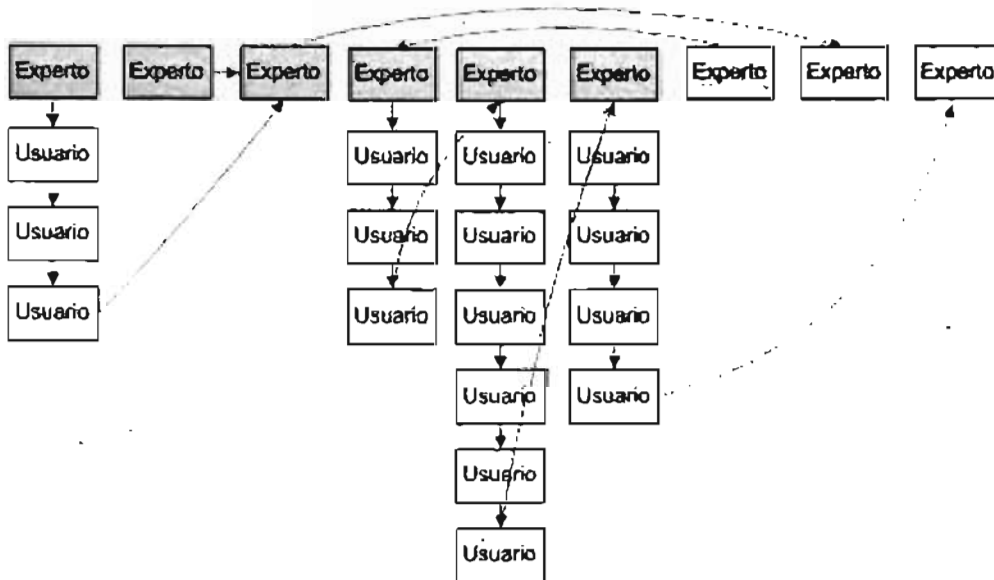


Figura II.1. Reconstrucción de una gráfica de UsAGE de comparación entre nodos de novato y experto

Su implementación está íntimamente ligada a la biblioteca y al modelo de eventos de TAE (Transportable Applications Environment ) Plus, un marco de trabajo para la generación de interfaces gráficas de usuario. Y es esta una de las debilidades más radicales de esta herramienta, ya que no permite que la misma idea pueda ser fácilmente transportada a marcos de trabajo diferentes.

Otro aspecto que habrá de ser tomado en cuenta por quien desee utilizar UsAGE es que el marco de trabajo TAE sólo se encuentra disponible para su adquisición para personas que viven dentro del territorio de los Estados Unidos de América.

Un aspecto a considerar, y del que también adolecen el resto de las herramientas que se mencionan es que todas trabajan con eventos de bajo nivel, como son el movimiento del puntero del ratón, la opresión y liberación de una tecla, entre otras. Hilbert [HILBERT, 1999] menciona que en el caso del análisis de secuencias de eventos de bajo nivel debidos a la interacción de un usuario particular, la suma de las partes no es el todo. Para elaborar conclusiones más acertadas y valiosas se requiere de información acerca del contexto, que no siempre viene incluida en la captura de eventos discretos.

### 2.1.3 QUIP

El proyecto QUIP o Quantitative User Interface Profiling (análisis cuantitativo de interfaces de usuario) es una herramienta desarrollada por Brian Helfrich y James A. Landay, de la Universidad de California en Berkeley [HELFRICH, 1999]

El documento en el que se describe esta herramienta poco se profundiza en aspectos como:

- a) La forma de obtener el programa
- b) El tipo de licencia con que se distribuye el programa
- c) El tipo de interfaces que soporta (web, WIMP)
- d) El tipo de eventos de interfaz que puede registrar
- e) El formato de representación de los datos de interacción del usuario

Para poder usar QUIP, el programador debe instrumentar la aplicación objeto de estudio, modificando el código fuente para realizar, además de las propias tareas del programa, el proceso de recolección de datos de interacción. Los datos obtenidos para cada acción incluyen un identificador de sesión de prueba, una marca de tiempo (en milisegundos), información descriptiva acerca de la ubicación dentro de la interfaz en que tuvo lugar la acción. Si es aplicable, el texto proporcionado por el usuario mediante el teclado también es recabado. Todos los datos son almacenados en una base de datos SQL Server.

La figura II.2 muestra cómo se grafican las secuencias de acciones realizadas por los usuarios. La parte coloreada con diagonales corresponde a la ruta tomada por el diseñador, en tanto que las otras flechas representan acciones tomadas por el resto de los usuarios. La parte de mayor grosor corresponde una mayor cantidad de usuarios que recorrieron esa ruta. El color de los arcos corresponde al tiempo promedio transcurrido entre las dos acciones.

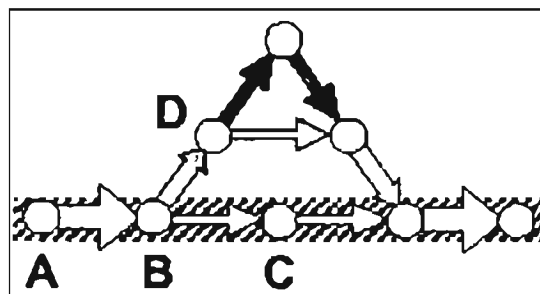


Figura II.2. Perfil de uso mostrado por QUIP

A pesar de que en primera instancia las ideas usadas en QUIP suenan bastante prometedoras, existen varias desventajas que pueden mencionarse:

- Fuera del documento en el que se propone y describe QUIP de forma somera. No existe ninguna descripción o especificación adicional, ni se ofrece alguna forma de obtener al menos una versión de prueba de algún programa ya instrumentado
- El campo de aplicación de QUIP parece estar orientado hacia los denominados clientes pesados (*fat-clients*), que son programas en que hay poca distribución de responsabilidades. En este tipo de interfaces existe solamente una capa, que contiene lógica de manejo de datos, de negocio y de control. De esta forma, se tendría que adicionar al código de presentación y de control fragmentos de manejo de datos de sesión, lo que a la larga resulta poco escalable.
- La modificación manual del código de la aplicación es un aspecto delicado que podría inyectar errores sutiles al sistema.
- El procedimiento no puede ser clasificado como totalmente automático. A pesar de que automatiza el proceso de recolección y análisis de datos, el proceso de instrumentación de la prueba sigue estando sujeto a particularidades de la interfaz.

#### 2.1.4 KALDI

KALDI es un conjunto de siglas para referirse con mayor brevedad a la herramienta cuyo nombre es Keyboard/Mouse Action Logger and Display Instrument, cuya traducción en español sería instrumento para el registro y despliegue de acciones de teclado/ratón.

KALDI fue desarrollada por Darren Jeffrey McRostie [MCROSTIE 1998], como una nueva propuesta de herramienta para la realización de la llamada ingeniería de usabilidad asistida por computadora. KALDI consiste de varios módulos:

- a) Herramienta de captura en vivo (Live Capture Tool). Se encarga del registro de eventos y captura de pantalla, utilizando para ello el entorno de trabajo Java AWT. Durante la sesión de pruebas con el usuario se capturan todos eventos y se realiza una grabación en video de lo que se muestra en pantalla. La información de esta manera registrada es enviada a través de la red hacia el segundo componente, la herramienta de monitoreo en vivo.
- b) Herramienta de monitoreo en vivo (Live Monitoring Tool). Recibe los datos y las capturas de pantalla, provenientes de la herramienta de captura en vivo, y los guarda en un dispositivo de almacenamiento permanente. El realizador de la prueba puede visualizar los eventos registrados en una notación gráfica mostrando la descripción y duración de las acciones registradas. Se puede observar también una reproducción en video de la salida del programa y los movimientos del ratón realizados por el usuario.

c) Herramienta de análisis. Se usa cuando ha concluido la sesión de prueba, para analizar los datos almacenados y proporcionar al ingeniero de usabilidad las siguientes características:

- Indexar los datos leídos por el usuario, el nombre de la tarea y la hora de registro
- Navegar y realizar búsquedas sobre los datos indexados
- Desplegar los datos de cada evento
- Permitir el agrupamiento jerárquico de eventos en clasificaciones o tareas
- Desplegar los eventos a diferentes niveles de detalle

KALDI no ha sido completamente terminada, ya que su desarrollo únicamente se realizó hasta el punto en que puede ser mostrada como "prueba del concepto". KALDI es un prototipo que demuestra la factibilidad de la construcción de una herramienta de su tipo. Debido a esto muchas características que originalmente se proponen sólo están parcialmente implementadas. En el mismo documento en que se describe KALDI, se mencionan sus limitaciones; a continuación se mencionan las más destacables:

a) Herramienta de captura en vivo:

- La herramienta de captura no funciona con aplicaciones que usan más de una ventana
- La herramienta de captura sólo funciona con componentes de AWT. A pesar de que la propia interfaz de KALDI fue desarrollada usando los componentes de Java Swing, la captura de datos que ofrece sólo se ajusta al modelo de eventos de AWT.
- La captura de eventos no se puede iniciar o detener de manera remota desde la herramienta de monitoreo
- Sólo reporta la posición del puntero del ratón, pero no la forma que este tiene.

b) Herramienta de monitoreo en vivo. No implementa ninguno de los puntos de su especificación funcional.

c) Herramienta de análisis:

- Solo permite la creación de tareas y acciones, pero no provee alguna forma de editar tareas o grupos de acciones
- La indexación por tiempo no está implementada

A pesar de que como herramienta KALDI es una buena propuesta, el enfoque con que realiza la automatización de la evaluación automatizada de interfaces presenta algunas desventajas, como son:

- Sólo se puede usar en interfaces tipo WIMP desarrolladas en Java.

- No se ajusta a otros modelos de eventos fuera de AWT, por lo que se tiene un escenario semejante al de UsAGE: no permite que la misma idea pueda ser fácilmente transportada a entornos de trabajo diferentes.

## **2.2 Herramientas existentes para interfaces web**

La mayor parte de los programas disponibles actualmente para realizar evaluación automática de interfaces de usuario para la WWW lo hacen confrontando el código HTML de la página web con una serie de guías preestablecidas. Con ello se busca asegurarse de alguna manera de la accesibilidad del contenido. Por ejemplo, indican al diseñador de la interfaz cuando ha olvidado poner textos alternos a las imágenes incluidas en la página, para tener certeza de que las personas con debilidad visual tengan al menos una porción de la idea que intentaba transmitirse usando tal figura.

En general, esos productos cumplen con reglas establecidas por grupos de reconocidos investigadores, o con estándares como el IEEE Std 2001-1999 [IEEE 1999].

El defecto principal que tienen herramientas de este tipo es el que solamente analizan código HTML, ya sea obtenido estáticamente o generado mediante alguna tecnología de generación dinámica de páginas web. Carecen de las características que se requerirían para hacer un análisis a mayor profundidad. Se pasa por alto la obtención resultados en los que se tomaran cuenta aspectos como metas y objetivos del usuario, descubrimiento del instante en que un usuario ha terminado una tarea, análisis o categorización del contenido de la página, historia de la interacción del usuario, etc.

En un estudio empírico reciente Ivory [IVORY 2002] describe como tras el uso de herramientas de este género no se observó un impacto apreciable en los resultados de las pruebas con usuarios. Incluso describe que en ocasiones, en vez llevar a la interfaz a una mejora, se le introdujeron más fallas.

A continuación se hace una descripción no exhaustiva de las herramientas de este tipo más conocidas:

### **2.2.1 CAST Bobby**

Es un servicio gratuito proporcionado por el CAST (Center for Applied Special Technology, Centro de Tecnología Especial Aplicada). En general realiza una labor de análisis de sitios web con el fin de identificar y eliminar barreras de acceso, en especial para personas con alguna discapacidad. En el mismo sitio también se distribuye una versión comercial de este software [CAST 2005].

### **2.2.2 UsableNet LIFT**

Desarrollado por la compañía UsableNet [USABLENET 2005], es una herramienta para la realización de pruebas de usabilidad y accesibilidad en distintas categorías: portabilidad, colores, fuentes, consistencia, uso, optimización y consistencia de imágenes, revisión de marcos, y validación de código.

### 2.2.3 DoctorHTML

Producido por Imagiware, DoctorHTML realiza pruebas como son: soporte del navegador para distintas etiquetas HTML, revisión de fuentes, análisis de tablas, imágenes y formas, y verificación de hiperenlaces [IMAGIWARE 2005].

### 2.2.4 La suite WebMetrics

La suite WebMetrics es un conjunto de programas desarrollados por el grupo de visualización y usabilidad (*Visualization and Usability Group*, VUG) del Instituto Nacional de Estándares y Tecnología (*National Institute of Standards and Technology*, NIST) de los Estados Unidos de América. [NIST 2005]

La finalidad del NIST al crear WebMetrics es la de explorar la incorporación de herramientas y técnicas que ayuden al proceso de evaluación de sitios web. La intención es hacer que las pruebas de usabilidad se realicen de manera rápida, remota y automatizada. Los autores de este conjunto de programas recomiendan que las ideas planteadas en WebMetrics sean usadas e inclusive comercializadas [NIST 2005b].

La primera versión de WebMetrics fue liberada en 1998, después de un año de desarrollo. A partir de entonces esta ha sido enriquecida con mayor funcionalidad, hasta llegar a la versión 2, puesta a disposición del público en general en Octubre de 2002.

WebMetrics consiste de seis programas y una especificación de formato de archivo. Sus componentes se mencionan a continuación y se describen en las secciones subsiguientes:

- WebVIP
- VizVIP
- WebCAT
- WebSAT
- TreeDec
- VizVIP
- Formato FLUD

#### 2.2.4.1 La herramienta WebVIP

WebVIP es una herramienta para incrementar el campo de aplicación de las pruebas de usabilidad con usuarios. WebVIP permite al experto en usabilidad instrumentar con rapidez un sitio web para realizar sobre él pruebas locales o remotas utilizando técnicas automatizadas.

La instrumentación consiste en adecuar las páginas de que consiste el sitio para posibilitar la recolección de datos de la interacción de los usuarios.

Durante el proceso de instrumentación, el ingeniero de usabilidad elige qué tipo de eventos de usuario se registrarán. Las elecciones de eventos corresponden a rutinas JavaScript que se incorporarán al código de una réplica de cada página del sitio web bajo prueba. Posteriormente se realiza una copia de todas las páginas

instrumentadas hacia una ubicación especial dentro del alcance del servidor web. Las páginas del sitio web permanecerán intactas, y los usuarios trabajarán con las páginas instrumentadas. Al finalizar la interacción del usuario con el sitio se creará un archivo de registro de eventos que contiene la secuencia de eventos ocurridos a lo largo de la interacción del usuario con el sitio.

La versión actual WebVIP solamente puede instrumentar sitios de prueba de páginas HTML estáticas.

Los archivos generados cumplen con la especificación del Formato FLUD [CUGINI 2002], y pueden ser usados por cualquier aplicación que sea capaz de manipularlos.

El nombre de WebVIP proviene de su nombre descriptivo: Web Variable Instrumenter Program, en español programa instrumentador de variables web.

#### 2.2.4.2 La herramienta VisVIP

VisVIP es una herramienta que puede leer archivos en formato FLUD, para presentar una visualización tridimensional de la ruta seguida por un usuario durante su interacción con un sitio web. VizVIP automáticamente genera un grafo bidimensional del sitio, en el que los vértices representan páginas web, y las aristas son los enlaces entre ellas.

El nombre de VisVIP proviene de su nombre descriptivo: WebVIP Visualization Tool, en español herramienta de visualización de WebVIP

#### 2.2.4.3 La herramienta WebCAT

Cuando un sitio web tiene su contenido distribuido a través de varias categorías, es conveniente conocer si el modelo mental de los usuarios acerca de la ubicación de los elementos coincide con el del diseñador de la interfaz. Una falla en este aspecto dará como resultado situaciones en que los usuarios no pueden encontrar las opciones que buscan debido a que mentalmente las ubican en un lado distinto al que fue considerado por el diseñador.

WebCAT o Web Category Análisis Tool –herramienta de análisis de categorías- permite realizar pruebas a un esquema de categorización dentro de un sitio web. Se busca determinar el nivel de comprensión que tienen los usuarios acerca de las categorías y elementos.

Lo que WebCAT presenta es en realidad una variación de la técnica de ordenamiento de cartas que funciona de manera interactiva.

#### 2.2.4.4 La herramienta WebSAT

WebSAT o Web Static Analyzer Tool –herramienta de análisis estático de web- es una herramienta orientada a la inspección de la composición del código HTML de las páginas web. Su objetivo es detectar potenciales problemas de usabilidad.



El funcionamiento de WebSAT es similar al de otras herramientas como Bobby [CAST 2005], LIFT [USABLENET 2005] y DrHTM [IMAGIWARE 2005].

WebSAT puede realizar análisis de páginas web aisladas o de sitios enteros, basándose en varias recomendaciones de usabilidad, como son:

- Las reglas de usabilidad propias de WebSAT
- Las reglas especificadas por el estándar del IEEE 2001-1999 [IEEE 1999]

#### 2.2.4.5 La herramienta TreeDec

En esencia TreeDec es una herramienta para incorporar guías de navegación que muestran al usuario el subárbol lógico en el cual se encuentra una página determinada, es decir, la ruta de acceso a dicha página.

El propósito principal de TreeDec es el de permitir a los diseñadores del sitio mantener una estructura lógica de árbol del sitio web en un solo archivo. Se usa ese archivo para controlar la adición sistemática de guías de navegación en las páginas web del sitio.

#### 2.2.4.6 El formato FLUD

FLUD (*Framework for Logging Usability Data*, marco de trabajo para registro de datos de usabilidad) es un formato de archivo propuesto por el NIST para el registro de los datos de sesiones de pruebas de usabilidad.

Durante mucho tiempo se ha intentado sin éxito obtener información de usabilidad a partir del análisis de los datos contenidos en los registros de transacciones de los servidores web. Ello se debe principalmente a que:

- La existencia de mecanismos que usan memoria caché para servir contenido web: caché en navegadores, caché en servidores web, caché en *proxies*, etc. Tales mecanismos, evitan dejar pasar todas las peticiones de un cliente hacia un servidor previendo que estuviesen solicitando contenido repetido. Esto imposibilita al servidor web almacenar la cantidad real de interacciones que ha tenido con alguna página alguna vez proporcionada por él.
- La carencia de medios para identificar de manera certera la interacción de un usuario con el servidor. Esto viene dado como consecuencia del uso de tecnologías como la asignación dinámica de direcciones IP mediante DHCP por parte del proveedor de servicios de Internet del cliente. Esto quiere decir que en algún momento, el servidor puede registrar la identidad de un usuario mediante su dirección IP. Sin embargo, el usuario puede regresar en una sesión posterior con una IP totalmente distinta, de suerte tal que el servidor no puede identificar que se trata de la misma persona.

Si se desea un estudio más detallado de los aspectos que impiden una recolección confiable de datos de usabilidad mediante técnicas convencionales de registro de transacciones en un servidor web, ya se han llevado a cabo estudios con una mayor profundidad. Algunos de los ellos están mencionados en la sección de referencias [DAVIDSON, 1999].

El formato FLUD ha construido una abstracción de nivel medio acerca de la actividad de un usuario (web), capturando comportamiento como el clic del puntero del ratón, la solicitud de una página web o la elección de un cuadro de selección.

El formato FLUD tiene un alcance claramente definido por sus creadores, que lo limitan indicando que es únicamente un formato de archivo, que incluye los elementos sintácticos y su significado. El formato FLUD únicamente define elementos generales para describir aspectos como son:

- Operaciones comunes realizadas por el navegador
- Operaciones comunes realizadas por el sistema de ventanas
- Operaciones comunes realizadas dentro de la página web

En el estado de desarrollo en que se encuentra, el formato FLUD aún no incorpora una forma de trabajar con páginas que consisten de varios marcos o *frames*.

La figura II.3 muestra la estructura básica de un archivo en formato FLUD.

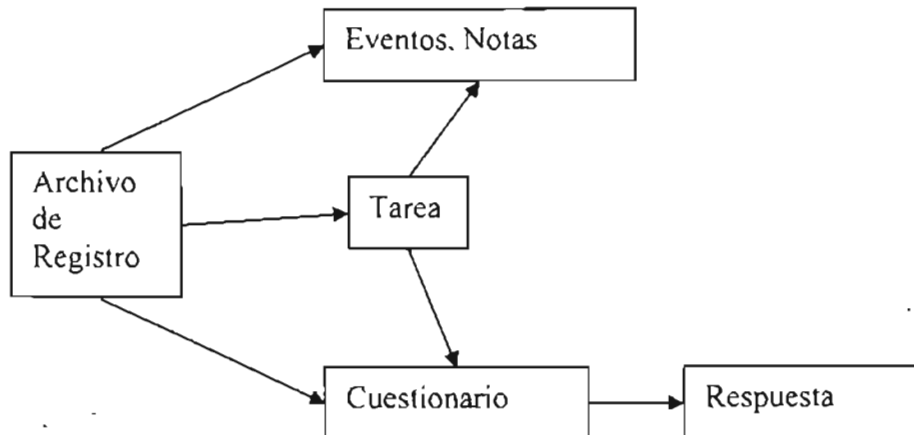


Figura II.3. Estructura básica de un archivo en formato FLUD

Un elemento *logfile* o archivo de registro es la unidad semántica mayor. Representa la sesión de un usuario interactuando con un sistema que pueda generar registros en formato FLUD. Cada *logfile* contiene una cabecera, la cual brinda una descripción general de la prueba: nombre del usuario, tipo de agente de usuario, hora de inicio de la prueba, etc.

Posterior a la cabecera del archivo viene una secuencia de registros o bloques de datos que describen cualquiera de las siguientes unidades semánticas: tareas(*tasks*), eventos(*events*), notas(*notes*), cuestionarios(*questionnaires*).

A su vez, las tareas pueden estar compuestas de eventos, notas y cuestionarios, y los cuestionarios de respuestas.

Un archivo también puede contener elementos *Ad\_hoc*, que son eventos definidos por el usuario para los fines que mejor le convengan. Estos elementos le

proporcionan al formato un punto de extensión que puede ser aprovechado para satisfacer necesidades específicas para una prueba u organización determinada.

Los autores del formato FLUD define varios niveles en que un archivo puede acatar la especificación:

- Nivel sintáctico. El archivo puede ser analizado de acuerdo a las reglas gramaticales que se mencionan en el Apéndice A.
- Nivel de sintaxis definida. Cumple con el nivel sintáctico y no contiene elementos Ad\_hoc.
- Nivel semánticamente verdadero. Se cumple que todo lo que está en el archivo en verdad sucedió, aunque hubiera omisiones. Se define con mayor detalle como sigue:
  - Cumple con el nivel sintáctico.
  - Todas las tareas, cuestionarios, eventos y notas representados en el archivo ocurren tal y como se describen.
  - Los elementos Ad\_hoc representados en el archivo ocurren tal y como se describen, de acuerdo a la semántica implícita en el generador de archivos.
  - Todos los elementos identificadores del archivo sirven efectivamente como identificadores; hay un mapeo exacto entre los elementos sintácticos y las entidades de la sesión de la cual provienen.
- Nivel semántico de estado completo. Es semánticamente verdadero y además todos los eventos definidos realmente ocurrieron en la sesión registrada en el archivo. Se exceptúan aquellos eventos que involucran una operación continua, como puede ser: el movimiento del apuntador del ratón, posiciones intermedias durante el movimiento de una barra de desplazamiento, del redimensionamiento o movimiento de una ventana.
- Nivel semánticamente completo. Es semánticamente verdadero y todos los eventos realmente ocurrieron en el archivo.

La implementación actual de WebVIP puede reconocer los siguientes eventos básicos que forman parte del formato FLUD:

- POINTERPRESS. Se ha hecho un clic con el puntero del ratón (o dispositivo de señalamiento, en caso de no tratarse específicamente de un ratón).
- POINTERRELEASE. El puntero del ratón ha sido liberado.
- POINTERMOVE. El dispositivo de señalamiento se ha movido.
- KEYPRESS. Se ha pulsado una tecla.
- KEYRELEASE. Se ha liberado una tecla que estaba presionada.
- ENTERWIDGET. El dispositivo de señalamiento ha entrado al área que abarca un elemento o control de interfaz (widget), por ejemplo, sobre un cuadro de texto o una casilla de verificación.

- LEAVEWIDGET. El dispositivo de señalamiento ha abandonado el área que abarca un elemento o control de interfaz (widget) , por ejemplo, sobre un cuadro de texto o una casilla de verificación.
- ENTERWINDOW. El dispositivo de señalamiento ha entrado al área de la pantalla correspondiente a una ventana.
- LEAVEWINDOW. El dispositivo de señalamiento ha abandonado al área de la pantalla correspondiente a una ventana.
- SIX\_DOF. El usuario manipula un dispositivo con seis grados de libertad. Tales dispositivos son usados frecuentemente en aplicaciones de tres dimensiones para modelar transformaciones rígidas.

Los eventos de la lista anterior son reconocidos sobre siguientes controles de interfaz:

- Button. Botón
- Radio. Conjunto de opciones de selección exclusiva (botón de radio)
- Checkbox. Casilla de verificación
- Textbox. Cuadro de texto
- Menu. Menú
- Slider. Barra de desplazamiento
- SscrollbarH, scrollbarV. Barras de desplazamiento horizontal o vertical
- Handle. La especificación no explica de qué control se trata
- Link. Enlace

WebVIP es capaz de inferir los siguientes eventos de mediano nivel.

- Eventos relacionados con las ventanas:
  - Open. Se ha abierto la ventana
  - Close. Se ha cerrado la ventana
  - Icon. Se ha minimizado la ventana
  - de\_icon. Se ha maximizado o restaurado el tamaño de la ventana
  - .move. Se ha movido la ventana
  - Resize. Se ha modificado el tamaño de la ventana
  - Raise. La ventana se ha hecho completamente visible, pudiendo inclusive haber cubierto el área de visibilidad de otras
  - Lower. La ventana ha sido cubierta parcial o totalmente por otra
  - Focus. La ventana está lista para aceptar la entrada del usuario
  - Blur. La ventana no aceptará entrada de datos por parte del usuario
- Eventos relacionados con la navegación web:
  - Print. Se ha impreso una página.
  - Page\_locate. La ubicación de una página web ha cambiado dentro de la ventana, posiblemente debido a que el usuario ha utilizado un enlace interno, una barra de desplazamiento, o porque ha cambiado el tamaño de la ventana.
  - Newpage. Los eventos alrededor de la recuperación y despliegue de una nueva página web. Consta de tres componentes, a saber:

- Request. Se solicita la página
- Loading. La página se está cargando
- Complete. La página se ha cargado completamente.

### Ventajas del uso del formato FLUD

Habiendo visto algunos aspectos de la forma tradicional en que los servidores web registran sus transacciones, y que funcionan en detrimento de la calidad de la información de usabilidad, encontramos en el formato FLUD una buena solución a algunos de los problemas que se mencionaron.

Para que un producto de la ingeniería sea considerado de calidad, ha de apegarse siempre que sea posible a los estándares que regulen el área a la que está enfocado. Respecto a la ingeniería del software, algunos estándares han ido emergiendo lentamente, fruto de las experiencias de las hasta ahora pocas generaciones de profesionales del ramo que han ido acumulando. Las mejores prácticas de la profesión sirven de guía para la elaboración de los proyectos del presente y del futuro.

Siendo FLUD una propuesta de una institución cuya labor es precisamente emitir lineamientos, guías y estándares. Se indica esto ya que el NIST regula muchas de las actividades referentes a tecnología en los Estados Unidos. Esto brinda un gran respaldo para todas las organizaciones que quisieran adoptar este marco de trabajo para sus labores cotidianas de aseguramiento de usabilidad.

El uso del formato FLUD para un estudio de usabilidad permitiría a una organización generar y analizar datos e información de evaluaciones de usabilidad. Esos datos analizarse con herramientas de diversa naturaleza, ya sea por el tipo de análisis que efectuaran, por la plataforma en que se ejecutan o por el público al que están dirigidos sus resultados.

El beneficio inmediato que reporta el uso de un formato normalizado para los datos es la facilidad con que estos se pueden intercambiar o compartir entre personas o aplicaciones con diferentes motivos. Así se desvanece la necesidad de mayores adecuaciones.

### **2.2.5 WET**

La Web Event-logging Tool o herramienta de registro de wventos web, mejor conocida como WET, es una herramienta desarrollada por la División de Ingeniería de Experiencia de Usuario de los Laboratorios AT&T [ETGEN 1999].

Su funcionamiento es similar al de WebVIP, descrito anteriormente. En algunos aspectos, la forma en que WET realiza la recolección es más sencilla. Mientras WebVIP incrusta el código JavaScript de recolección de datos dentro de cada página web de manera estática, WET simplemente coloca una referencia al archivo JavaScript que contiene las rutinas para recolección de eventos.

Aunque la diferencia entre WebVIP y WET parece sutil no lo es tanto, pues el enfoque de WET facilita su integración con tecnologías de generación de contenido dinámico como: Servlets, JSP, ASP, ColdFusion, etc. La ventaja del enfoque de

WET radica en que usando las instrucciones de inclusión de archivos se puede optar por incluir o excluir el código correspondiente a la captura de eventos. Se evitan así los inconvenientes existentes en WebVIP:

- Es difícil instrumentar un sitio web de tamaño considerable ya que se debe alterar el código de cada página web del sitio.
- La realización de una copia del sitio completo puede conducir al surgimiento de problemas con la navegación del sitio instrumentado, ya que se puede dar el caso de especificación de enlaces a rutas no válidas.

Un aspecto negativo que se debe mencionar es que durante el proceso de investigación sobre esta herramienta no se pudo encontrar ningún archivo adicional al documento que la describe. Esto implica que no hay documentación que explique el formato que siguen los archivos de registro generados por WET. No existe ningún programa ya sea en código fuente o ejecutable que pueda ser obtenido a fin de evaluar la funcionalidad descrita. Este inconveniente fue el factor persuasivo para la elección de usar WebVIP, pues se encuentran disponibles para descarga su código fuente, el archivo ejecutable y la documentación. Adicionalmente se tiene la certeza de contar con la capacidad de generar registros en formato FLUD.

### **2.3 Aspectos no cubiertos por enfoques anteriores**

Clasificando en dos géneros las herramientas de automatización de evaluación de interfaces de usuario, podremos resumir sus características y deficiencias.

#### **2.3.1 Herramientas de registro y/o reproducción de eventos de interacción del usuario con la interfaz**

Podemos indicar que las herramientas existentes dedicadas a la realización de evaluación automatizada de interfaces de usuario en el aspecto de registro y/o reproducción de eventos de interacción del usuario con la interfaz tienen las siguientes características:

- Registro de eventos del usuario a bajo nivel.
- Reproducción de eventos de usuario.

Dichas herramientas también presentan las siguientes deficiencias:

- Aquellas que tienen registro y reproducción de eventos de interacción del usuario con la interfaz solamente están disponibles para interfaces tipo WIMP (*windows, icons, pointer and mouse*; ventanas, íconos, apuntador y ratón). Se da por descartado su aplicabilidad inmediata a interfaces web.
- Las que realizan captura de eventos de usuario no utilizan un formato estandarizado.

- Los eventos de usuario capturados son de bajo nivel, por lo que solo tienen aplicación práctica en ejecuciones breves.
- Algunas de la que realizan captura de eventos de usuario están hechas para entornos de trabajo específicos (TCL/Tk, TAL, AWT), lo que dificulta en alto grado la portabilidad inmediata a otros ambientes.
- La propia herramienta VISVIP de la Suite WebMetrics puede tener similitud con el módulo de Visualización de Comparaciones de Secuencias de Eventos de usuario, pero su funcionamiento no es el mismo. VISVIP sólo grafica secuencias de eventos de un solo usuario en particular. Por otro lado, como se verá más adelante, el componente de visualización del presente proyecto está abocado a la representación de comparaciones entre dos usuarios. Otro punto de divergencia es que el módulo de visualización puede utilizar secuencias de eventos de un ingeniero de usabilidad que sigan el formato FLUD simplificado, con las ventajas que ya se han descrito en secciones anteriores.

### **2.3.2 Herramientas para verificación de ajuste a normatividad**

Estas herramientas tienen como única cualidad la de analizar el código HTML de interfaces web, e indicar si cumplen con los lineamientos de las normas reconocidas. Se enfocan principalmente en asegurar la accesibilidad de un determinado sitio o página. En ningún momento se puede realizar un análisis del comportamiento dinámico de la interfaz.

## **2.4 Conclusión y discusión**

En el presente capítulo se ha mostrado un panorama general de los avances que hasta ahora se han dado en el aspecto de automatización de evaluación de interfaces de usuarios.

La tabla II.1 muestra una matriz en donde se resumen los aspectos descritos en el presente capítulo. Se pueden observar las carencias de cada una de las herramientas presentadas. Es de hacer notar que ninguna de ellas cuenta con todas las características deseables. Ello significa que aún existe un campo abierto para la generación de herramientas innovadoras que incorporen los beneficios combinados de todas ellas.

Como se habrá podido observar, se trata de propuestas que se orientan a la solución de diferentes tipos de problemas, sobre diversos tipos de interfaces de usuario, y con sus particulares estrategias. Sin embargo, no todo está dicho al respecto, ya que como se verá en el capítulo siguiente, aún hay oportunidad de presentar nuevos enfoques que puedan aportar avances sustanciales.

Características	Interfaces WIMPS				Interfaces Web		
	TkReplay	UsAGE	QUIP	KALDI	Bobby	LIFT	DoctorHTML
Registro de eventos	✓	✓	✓	✓	x	x	x
Registro en Formato Estandarizado	x	x	x	x	x	x	x
Portabilidad entre plataformas	x	x	x	x	✓	✓	✓
Comparación de secuencias de eventos	✓	✓	✓	x	x	x	x
Disponibilidad	x	x	x	x	✓	✓	✓
Análisis de HTML estático	x	x	x	x	✓	✓	✓

Tabla II.1. Matriz comparativa de características proporcionadas por las herramientas existentes



# Capítulo III

## Proceso de desarrollo

A partir de los aspectos discutidos en el capítulo anterior se desarrolló un plan para la realización de un sistema de software que pueda satisfacer las necesidades que no son cubiertas por otras herramientas.

### 3.1 Especificación

Después de enumerar las características y carencias de los enfoques ya descritos. Se descubre la factibilidad de emitir una propuesta que logre subsanar las deficiencias descritas, contribuyendo así de manera eficaz al área de evaluación automatizada de interfaces gráficas de usuario. Se propone el desarrollo de herramienta que posea las siguientes características:

- 1) Obtención o generación de un registro que describa fielmente toda la interacción realizada por el usuario durante una o varias sesiones frente a la interfaz. Características esenciales de ese registro son:
  - Que los datos en él recolectados proporcionan elementos suficientes para describir aspectos de usabilidad.
  - Que el formato de los datos siga una convención o estándar de manera que pueda ser libremente analizado e intercambiado por múltiples herramientas de distinta naturaleza.
  - Que pueda ser usado para describir interacciones con páginas o sitios web.
- 2) Análisis de las acciones del usuario. Es decir, que a partir del análisis del registro indicar la correspondencia de esos registros con eventos que ya tengan cierto significado como son pulsaciones de botones de comando, movimiento de una ventana, etc.
- 3) Comparación de secuencias de usuario. Confrontar las acciones identificadas a partir del registro de la interacción del usuario con otras secuencias de acciones previamente alimentadas al sistema. Esto se realiza de manera que el sistema pueda inferir si lo que hace o desea hacer el usuario se asemeja de alguna manera a una de esas tareas. Para lograr lo anterior se buscará una representación que posibilite la comparación de tareas. También se propondrá una técnica que implemente esa comparación, para ser integrada al sistema.
- 4) Visualización de comparaciones de secuencias de eventos de usuario. La visualización de comparaciones de secuencias de eventos entre distintos usuarios. Proporcionará elementos informativos importantes para los expertos o partes interesadas en conocer y analizar con mayor profundidad los resultados arrojados por las comparaciones realizadas. Será un gran complemento visual para una toma de decisiones adecuada.

### 3.1.1 Contribución y relevancia

Se proporcionarán elementos de juicio para la búsqueda de mejoras en la calidad de la interacción para los expertos en usabilidad y para las interfaces mismas.

Se identifican dos vertientes en las que se harán los aportes más importantes:

- Interacción humano – máquina
- Interfaces adaptables al usuario

#### 3.1.1.1 Aportaciones en el área de interacción humano – máquina

En el área de estudio de la interacción humano – máquina se tendría un impacto considerable en diferentes aspectos:

Antes que todo, en el área de evaluación de interfaces de usuario. Se tendría una nueva herramienta para analizar de manera automática la interacción de un usuario con una interfaz. Se ahorraría mucho tiempo a los especialistas en la realización de estudios de usabilidad, con las ventajas adicionales de:

- Disminución dramática de costos por realización de estudios de usabilidad: compra o renta de equipo para llevar a cabo el estudio, acondicionamiento y mantenimiento de instalaciones adecuadas, etc.
- Posibilidad de mayor aceptación de un producto debido a la incorporación de una mejor usabilidad.
- Disminución de costos de capacitación de las personas que habrán de usar el sistema.
- Disminución de costos por concepto de soporte técnico
- Mejora de la imagen que tiene la organización ante los clientes, socios y competidores.
- Aplicabilidad a interfaces HTML o web, un aspecto dejado de lado hasta ahora por proyectos similares.

Por otra parte, las ventajas que proporciona el poder realizar comparaciones directas entre el comportamiento de un usuario con una especificación de referencia son varias, como se describen a continuación para algunos casos típicos:

- Usuario experto contra usuario novato. Dependiendo de los objetivos que se busquen, habrá sitios en los que el interés primordial será tener una productividad importante. Lo anterior se logra incluso a costa de un incremento en la complejidad de la navegación o de otros aspectos de la interfaz, con la consecuente agudización en la curva de aprendizaje.

Esto puede ser palpable para sitios como Intranets o terminales de punto de venta, donde un usuario experto puede obtener beneficios máximos en productividad una vez que ha logrado tener cierta experiencia.

Una comparación de este tipo podrá darnos indicadores para saber cuál podría ser la esperanza de productividad del sitio. También se dispondrá de elementos de decisión acerca del tipo de entrenamiento que deberán tener los usuarios que recién adopten el sistema. Así se logrará que logren un uso óptimo del mismo.

- Diseñador de la interfaz contra usuario novato. Indicará las posibles diferencias entre las expectativas del diseñador, y la ejecución realizada por los usuarios.

La utilidad de una comparación de estos datos cobraría gran importancia para sitios en los que se espera que los usuarios rara vez tengan experiencia. Un ejemplo de un sitio que se podría beneficiar de este tipo de pruebas sería uno del gobierno, en el que los usuarios llevarán a cabo trámites poco convencionales o que se realizan una vez en la vida.

- Diseñador de la interfaz contra usuario normal. Hay ocasiones en que el diseñador de la interfaz ha hecho un trabajo en el cual encuentra remotamente posible la existencia de algún problema de usabilidad. Nada hay mejor que la realización de pruebas con usuarios para conocer la veracidad o falsedad de su idea.

Esta comparación revelará a detalle en qué momento se alejan o se acercan los usuarios a lo que puede ser llamado ruta óptima propuesta por el diseñador. Si existe una alta incidencia de usuarios en un error, tal vez el usuario deberá tener en consideración las posibles causas de esa diferencia. Ello le permitirá elaborar conclusiones a partir de las causas descubiertas y emitir posibles correcciones o modificaciones para adecuar la interfaz a las expectativas del usuario.

- Diseñador de la interfaz contra usuario experto. Esta comparación podría señalar vicios que aparecieran en las interacciones de usuarios experimentados en el uso del sistema. Se harían evidentes conductas inesperadas por el diseñador de la interfaz.

Un ejemplo de esas desviaciones podría ser el descubrimiento de una forma de acceso a secciones del sitio en las que el usuario no tiene incumbencia, ya sea porque no está autorizado, o porque podrían desvirtuar el modelo mental que esperamos que el usuario tenga con respecto a la interfaz.

- Experto en usabilidad contra usuario novato. Las ventajas y aplicaciones serían semejantes a las descritas para una comparación entre el diseñador de la interfaz y el usuario novato.
- Experto en usabilidad contra usuario experto. Las ventajas y aplicaciones serían semejantes a las descritas para una comparación entre el diseñador de la interfaz y el usuario experto.
- Experto en usabilidad contra diseñador. Un experto en usabilidad podría emitir correcciones a un diseño usando el mismo formato que utilizara el diseñador para proponer rutas óptimas

### 3.1.1.2 Aportaciones en el área de interfaces adaptables al usuario

La herramienta que se propone puede servir como el núcleo o motor de inferencia para una interfaz que se adecue a las necesidades o limitaciones de un usuario en tiempo real. Esto debido a que la identificación de tareas de alto nivel (por ejemplo finalización de una búsqueda, realización completa de una transacción, necesidad de ayuda.) proporcionaría mayor información crítica para la toma de decisiones de la interfaz que un abundante flujo de eventos de bajo nivel. En sentido figurado se podría decir que la interfaz "dejará de ver árboles para concentrarse en la comprensión del bosque."

## 3.2 Diseño

Para la fase de diseño se propuso el modelo de un sistema que pudiera proporcionar una implementación de la especificación descrita en apartados anteriores.

### 3.2.1 Arquitectura general

Para cumplir con la propuesta emitida, se tendrá un sistema con la arquitectura básica que se muestra en la figura III.1.

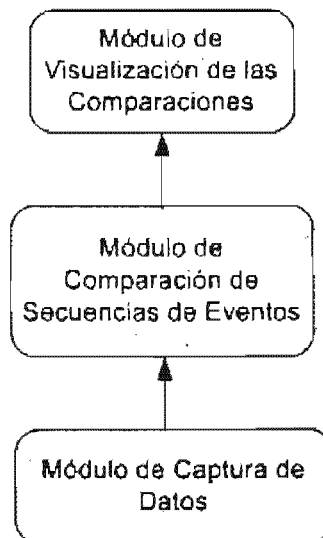


Figura III.1. Arquitectura Básica del Sistema

- a) Un módulo de captura de datos de acciones del usuario. Será un programa capaz de capturar los eventos generados por la interacción de un usuario con el sistema, de manera que puedan ser analizados posteriormente por alguna herramienta.
- b) Un módulo de comparación de secuencias de eventos. Este servirá para realizar comparaciones entre las secuencias de eventos generadas por distintos usuarios. En efecto, Las secuencias de eventos no necesariamente deberán proceder del módulo de captura de datos, sino que pueden ser también escritas manualmente por alguna persona (o generadas

auxiliándose de alguna herramienta adicional). Por ejemplo, el diseñador puede proponer secuencias que describan condiciones particulares de interacción.

- c) Un módulo de visualización de resultados. En esencia lo que se habrá de observar será una representación gráfica que muestre coincidencias y diferencias entre secuencias de usuarios. Así se podrán presentar los resultados a los expertos en usabilidad. Estos los podrán visualizar de una manera sencilla, sin tener que trabajar directamente con archivos de secuencias de eventos.

### **3.2.2 Recolección de datos de acciones del usuario**

Obtener un registro que describa fielmente toda la interacción realizada por el usuario durante una o varias sesiones frente a la interfaz. Puede provenir directamente de la interacción de un usuario, o ser escrito por un ingeniero de usabilidad. Características esenciales de ese registro son:

- Que los datos en él recolectados proporcionan elementos suficientes para describir aspectos de usabilidad como son: eventos generados durante su interacción, secuencia de páginas en que ha navegado, patrones de navegación, errores frecuentes, etc.

Esta meta pueden ser satisfecha usando el formato FLUD de NIST. Dicho formato está diseñado teniendo en mente precisamente la descripción de datos de la interacción de un usuario con un sistema. El formato FLUD tiene una fuerte orientación al uso de datos de usabilidad.

- Que el formato de los datos siga una convención o estándar de manera que pueda ser libremente analizado e intercambiado por múltiples herramientas de distinta naturaleza.

El formato FLUD es una opción viable, ya que la institución que ha emitido la especificación de este formato (NIST) tiene como una de sus actividades primarias la elaboración de estándares y normas generalmente relacionados con las Tecnologías de la Información. A pesar de no ser un estándar, FLUD se presenta como la opción más adecuada. Si bien existen otras propuestas de formatos de registro de interacción del usuario como WET [ETGEN 1999], estas carecen de soporte de una institución de la envergadura del NIST, de una amplia comunidad de investigadores, mucho menos de un estándar publicado.

- Que pueda ser usado para describir interacciones con páginas o sitios web.

De nueva cuenta el formato FLUD se muestra como una opción conveniente, pues sus diseñadores han tenido en cuenta la necesidad de aplicarlo en ambientes de interfaces web. Tales cualidades lo hacen el candidato ideal para cumplir con el presente punto de la propuesta. Otro punto persuasivo a favor del uso de FLUD en relación a este punto es el hecho de que el mismo NIST proporciona la herramienta WebVIP, ya descrita en un apartado previo. WebVIP en esencia sirve para poner en marcha un sistema de recolección

de datos de uso de interfaces web para un determinado sitio. Se obtiene como resultados archivos que describen en formato FLUD la interacción de un determinado usuario.

Una vez que se han recabado los datos del registro, se procede a analizarlos para obtener información a partir de ellos

### **3.2.3 Análisis de las acciones del usuario**

A partir del análisis del registro generado en la etapa anterior, el sistema debe indicar la correspondencia de los registros obtenidos por la interacción de cada usuario con eventos que ya tengan cierto significado: pulsaciones de botones de comando, movimiento de una ventana, etc.

La herramienta habrá de tener disponibles opciones de ejecución que precisamente realizan inferencias a partir de una secuencia de eventos en bruto, para identificar lo que podría llamarse eventos de mediano nivel. Estos son eventos que sin ser de bajo nivel, como los movimientos del ratón o presiones de una tecla, tampoco son de alto nivel, como por ejemplo la indicación de la realización de una compra, o la advertencia de que se ha llenado correctamente un formulario de registro. Los eventos de mediano nivel aquí descritos son por ejemplo la detección de pulsación de botones de envío de formulario, entrada o salida de una ventana, la apertura de un determinado enlace con su URL relacionada, etc.

### **3.2.4 Comparación de secuencias de eventos de interacción**

El sistema deberá confrontar las acciones identificadas a partir del registro de la interacción del usuario con otras secuencias de acciones previamente alimentadas al sistema. Así es como puede inferir si lo que hace o desea hacer el usuario se asemeja de alguna manera a una de esas tareas. Para lograr lo anterior se buscará una representación que posibilite la comparación de tareas. También se propondrá una técnica que implemente esa comparación, para ser integrada al sistema.

Para lograr el cometido de este punto se escribirá un módulo que sea capaz de indicar los puntos de coincidencias y discrepancias entre distintas secuencias de eventos. Por ejemplo, La figura III.2 muestra las diferencias entre los eventos llevados a cabo por un experto (trazos en azul), por un usuario de la interfaz (aristas rojas); y las coincidencias entre sus rutas (líneas moradas).

Al final se obtiene un conjunto de datos en formato simple que pueden ser leídos por una herramienta convencional para graficación de grafos(en el sentido matemático de la palabra).

### **3.2.5 Visualización de comparaciones de secuencias de eventos de interacción**

La generación de datos destinados específicamente para la visualización de las comparaciones de secuencias de eventos de usuarios se realizará de manera simultánea con el análisis descrito en el apartado anterior.

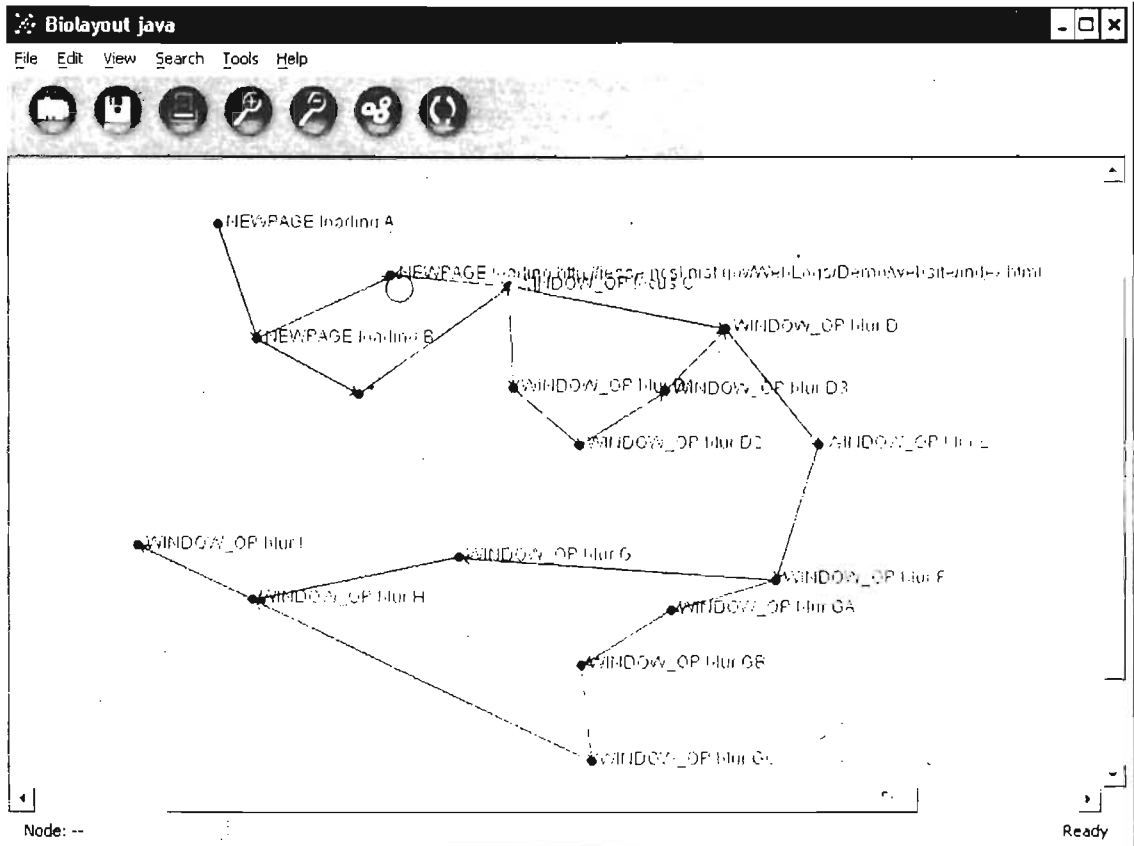


Figura III.2 Comparación de secuencias de eventos de interacción de dos usuarios distintos

Para esta fase del proceso se utilizará un programa llamado Biolayout Java [ENRIGHT 2001], enfocado a la construcción de grafos. Dicha pieza de software lee la especificación de los grafos desde un sencillo archivo de texto en que por cada renglón se mencionan en dos columnas los dos vértices de cada una de las aristas que forman parte del grafo. Existe una posible tercera columna opcional que indica el peso que ha de tener la arista en cuestión.

Se ha procurado mediante el uso de Biolayout Java, tal y como sucede con el caso de WebVIP, utilizar programas que ya se encuentran desarrollados. De esta manera se reduce el esfuerzo de desarrollo de la propuesta sustentada en esta tesis. Por otra parte nos ahorra la resolución de problemas o la implementación de soluciones para asuntos como la graficación automática de grafos [LOK 2000] mediante el uso de la herramienta Biolayout Java, o la implementación desde cero de un sistema de recolección de datos de un sitio web para el formato FLUD, funcionalidad proporcionada por WebVIP).

Resulta interesante mencionar que Biolayout Java está orientada a la visualización de grafos con un enfoque biológico. En esencia puede trabajar en aspectos como son similitud de subsecuencias, caminos metabólicos, cascadas regulatorias de genes e interacciones de proteínas, etc. Tiene un diseño tal que permite trabajar con redes de tipo, tamaño y complejidad variadas. El programa está escrito en el lenguaje Java por Leon Goldovsky, Ildefonso Cases y Anton Enright.

Para mayor información consultar los ejemplos de uso de la herramienta [ENRIGHT 2005].

Biolayout Java está liberado bajo una licencia de tipo académico avalada por el Laboratorio Europeo de Biología Molecular, situado en Alemania. Se garantiza el uso del programa únicamente para fines científicos sin fines de lucro [ENRIGHT 2001].

### **3.3 Implementación**

En este apartado se describirá cómo se llevo a cabo la implementación de los módulos del sistema.

#### **3.3.1 Módulo de captura de datos**

La propuesta de esta tesis se enfoca a interfaces del tipo web. Es por ello que habrá de utilizarse para este módulo será un programa que pueda incorporar algún mecanismo de captura de eventos a las páginas web usadas durante la interacción del usuario con el sitio.

Al momento de decidir sobre la implementación de este módulo, se tenían dos posibles opciones:

- a) Implementar un sistema o método capaz de lograr hacer que se obtenga un archivo, registro, o colección de objetos en memoria que contenga la secuencia de eventos generados por cada página del sitio sobre el que se apliquen las pruebas de usabilidad.
- b) Buscar alguna implementación ya existente para el problema descrito en el inciso anterior, y encontrar la forma de hacerla interoperar con los otros módulos que forman parte del sistema. Desde un punto de vista práctico esta opción se presenta como la más adecuada ya que ahorra mucho tiempo de desarrollo.

No es necesario buscar de manera tan exhaustiva para encontrarse con WebVIP, un programa que acompaña a la especificación del formato FLUD. Dicho programa, distribuido desde el sitio web dedicado a FLUD en el NIST, realiza las tareas primarias requeridas para poder llevar a cabo la recolección de eventos de la interacción del usuario con el sistema. La implementación de un esquema básico de registro de eventos se describe a continuación:

##### **3.3.1.1 Instalación de WebVIP**

Se procede a instalar o bien WebVIP exclusivamente, o toda la suite WebMetrics.

Como labor previa, se deben instalar los programas de los cuales depende WebVIP:



- Un servidor web, que puede ser Apache, o Microsoft IIS.
- Un intérprete del lenguaje Perl.

### 3.3.1.2 Instrumentación de las páginas de la prueba

Se debe hacer una elección del conjunto de páginas Web que habrán de usarse para la prueba. Por el momento el programa WebVIP solamente soporta la aplicación a páginas HTML estáticas, aunque, en secciones posteriores se discutirá la posible aplicación de este esquema a páginas dinámicas.

Usando la ventana de administración de WebVIP, indicar al programa la ubicación del sitio a probar, se elige qué eventos habrán de ser registrados (Figuras III.3, III.4 y III.5). En ellas se elige el sitio sobre el que se desea interactuar. Posteriormente se indica si ya se han copiado los archivos del sitio al servidor donde se habrán de realizar las pruebas. A continuación se eligen los eventos que habrán de ser capturados durante cada una de las pruebas, con lo que se concluye el proceso de instrumentación.

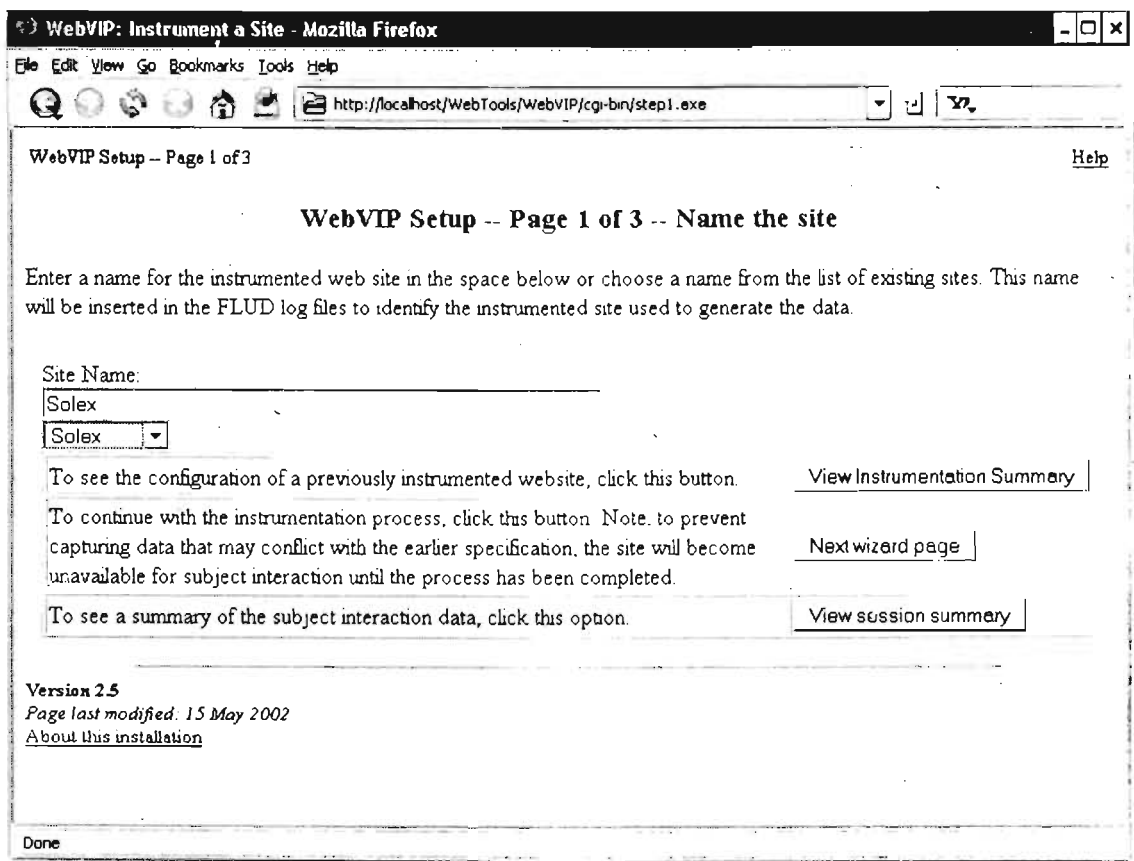


Figura III.3 Pantalla de selección del sitio a instrumentar

Para poder lograr que las páginas colecten y guarden la secuencia de eventos que se generan durante la interacción del usuario con la interfaz, a cada página HTML del conjunto de prueba se le agregan instrucciones especiales. Ellas están en lenguaje Javascript, y realizan la captura de eventos de los tipos que el

usuario ha indicado que desea sean procesados. Para una descripción a mayor detalle los eventos que pueden ser capturados por WebVIP, consulte la especificación del formato FLUD [CUGINI 2002].

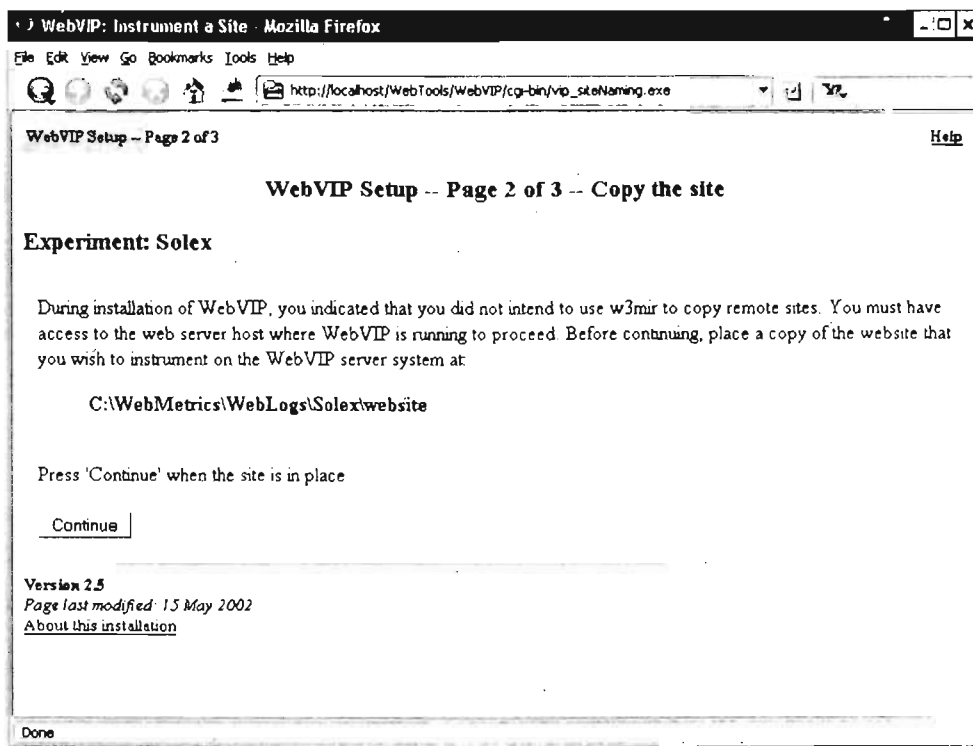


Figura III.4 Pantalla en que se confirma el origen de las páginas del sitio a instrumentar

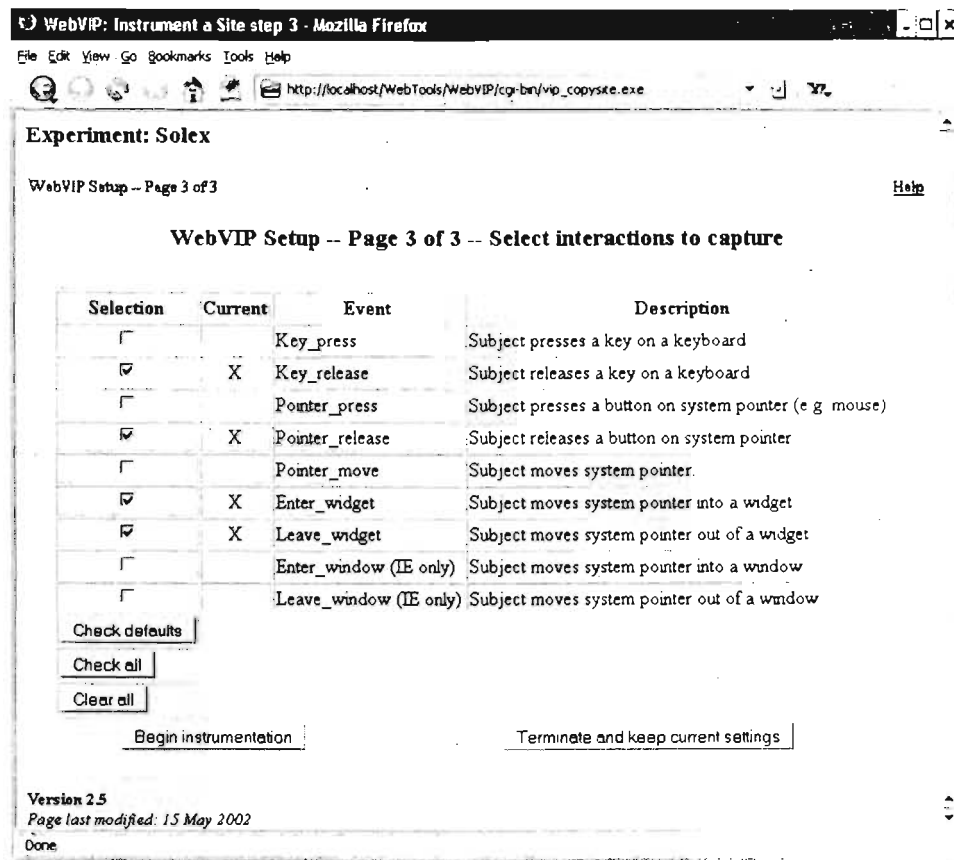


Figura III.5 Elección de los eventos que serán guardados a lo largo de la prueba de cada usuario

### 3.3.1.3 Realización de pruebas

Se lleva a cabo propiamente la interacción de los usuarios con el sitio web instrumentado. Al arrancar la prueba de cada usuario, este se identifica con un nombre, y se le asigna automáticamente un número de sesión. Los registros de actividades para cada usuario se guardarán en un subdirectorío expresamente creado para este fin. Tal subcarpeta se ubica bajo el directorio raíz donde WebVIP guarda todos los datos relacionados con la prueba. Los detalles de configuración de trabajo de WebVIP pueden encontrarse en el manual de instalación y configuración proporcionado por el NIST [NIST 2005c].

Una vez registrado en la prueba, el usuario empieza a realizar las tareas que se le indiquen. Para cada tarea especificada, el usuario llenará una ventana flotante como la que se muestra en las figuras III.6, III.7 y III.8; en esa ventana pondrá el identificador de la tarea, e indicará el inicio de ella presionando un botón. Una vez concluida esa tarea, dentro de la misma ventana flotante el usuario indicará que la ha dado por terminada, y podrá proseguir con la siguiente tarea, si es que se le ha asignado alguna otra. En todo momento el usuario puede agregar comentarios acerca de la experiencia que está teniendo en su contacto con la interfaz.

Al final de la interacción de un usuario, WebVIP generará un archivo de texto con extensión *.ulog* correspondiente a la sesión. Si así se indica, también generará un archivo con extensión *.parse*, que contiene la versión sujeta al formato

FLUD de la secuencia de eventos de la interacción de usuario con el sitio. También es posible obtener un archivo HTML de la sesión, que contiene una versión mucho más apta para visualizar, con coloreo de sintaxis.

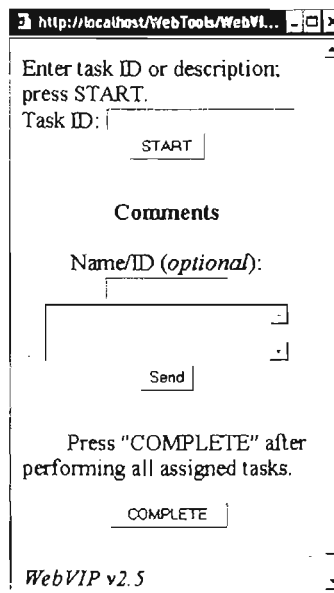


Figura III.6 Pantalla de control de pruebas con usuarios de WebVIP

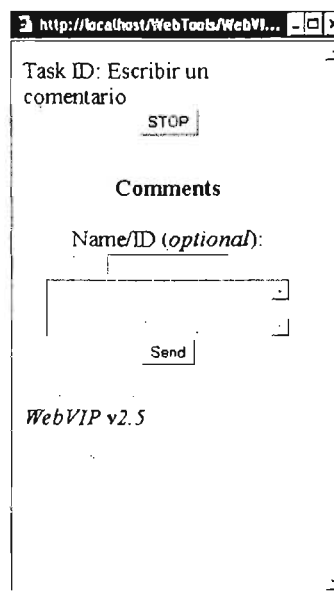


Figura III.7 Pantalla de control de pruebas con usuarios de WebVIP, visualizada durante la realización de una tarea

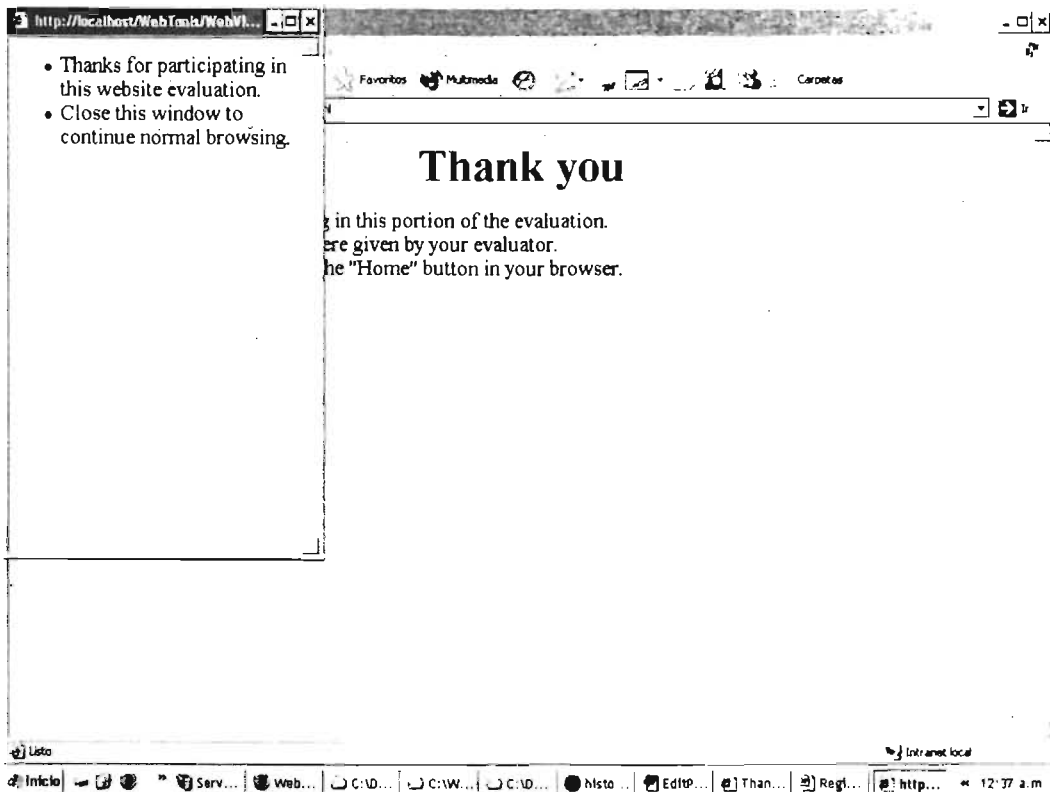


Figura III.8 Pantalla de control de pruebas con usuarios de WebVIP, al finalizar la prueba

#### 3.3.1.4. Generación de los datos en formato FLUD Simplificado usando CutFLUD

Debido a que los siguientes módulos consumen datos no en el Formato FLUD común sino en su versión simplificada, es necesario hacer una conversión.

Se ha desarrollado un programa llamado CutFLUD, cuya labor principal es realizar la conversión de archivos *.parse* (formato FLUD, secuencias de eventos generados por el usuario durante su interacción con el sistema) a archivos *.sflud* (formato FLUD simplificado), como se puede ver en la figura III.9. Se muestra la secuencia en que se realiza la transformación de datos desde su versión en bruto, recién capturados, hasta que se encuentran en formato FLUD Simplificado

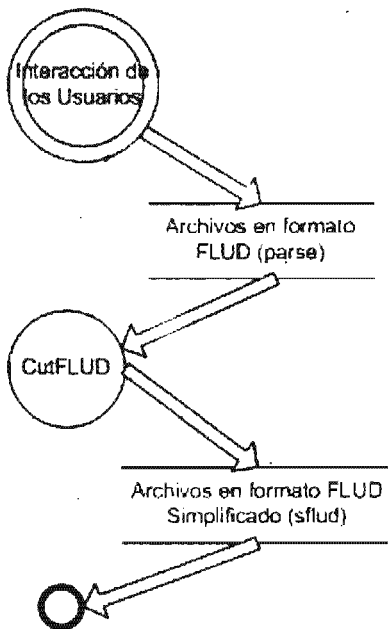


Figura III.9 Flujo de datos para la transformación de secuencias de eventos desde el formato FLUD a formato FLUD Simplificado

A pesar de que CutFLUD es un programa que está diseñado para ser invocado desde la línea de comandos, existe una versión sólo disponible para plataformas Microsoft Windows de 32 bits que cuenta con una interfaz gráfica. La versión recién mencionada fue desarrollada para facilitar el uso de la herramienta. La interfaz en modo gráfico de CutFLUD se caracteriza por su simplicidad.

La figura III.10 ilustra el proceso de elección de datos de entrada y obtención de archivos de salida, que también se describe a detalle a continuación:

Para llevar a cabo la conversión de archivos *.parse* a archivos *.sflud* basta situarse en el cuadro de diálogo principal (figura III.11). Se tienen dos paneles principales: el primero recaba los archivos de la interacción de los usuarios en formato *.parse*; y el segundo indica el directorio de salida para los archivos *.sflud*. Para poder llenar ambos espacios es necesario invocar los cuadros de diálogo correspondientes y realizar la selección de los siguientes elementos:

- Archivos *.parse* que contienen la secuencia de eventos de los usuarios en formato FLUD. Se eligen usando los cuadros de diálogo de las figuras III.12 y III.13. El proceso consiste básicamente de seleccionar el directorio donde se ubican los archivos *.parse*, para posteriormente elegir uno o más archivos.
- Directorio en que se grabarán los archivos de salida con extensión *.sflud*. Se elige a través del cuadro de diálogo mostrado en la figura III.14. Simplemente se elige el directorio de salida, y si este no existe, será creado.

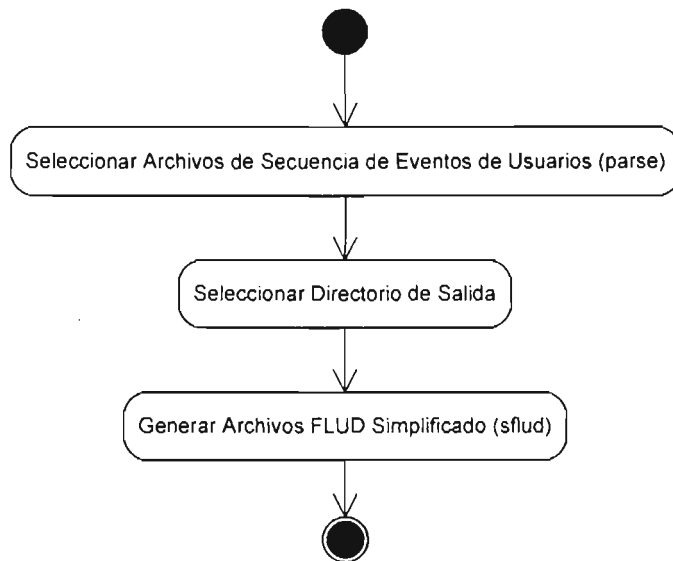


Figura III.10 Utilización de CutFLUD

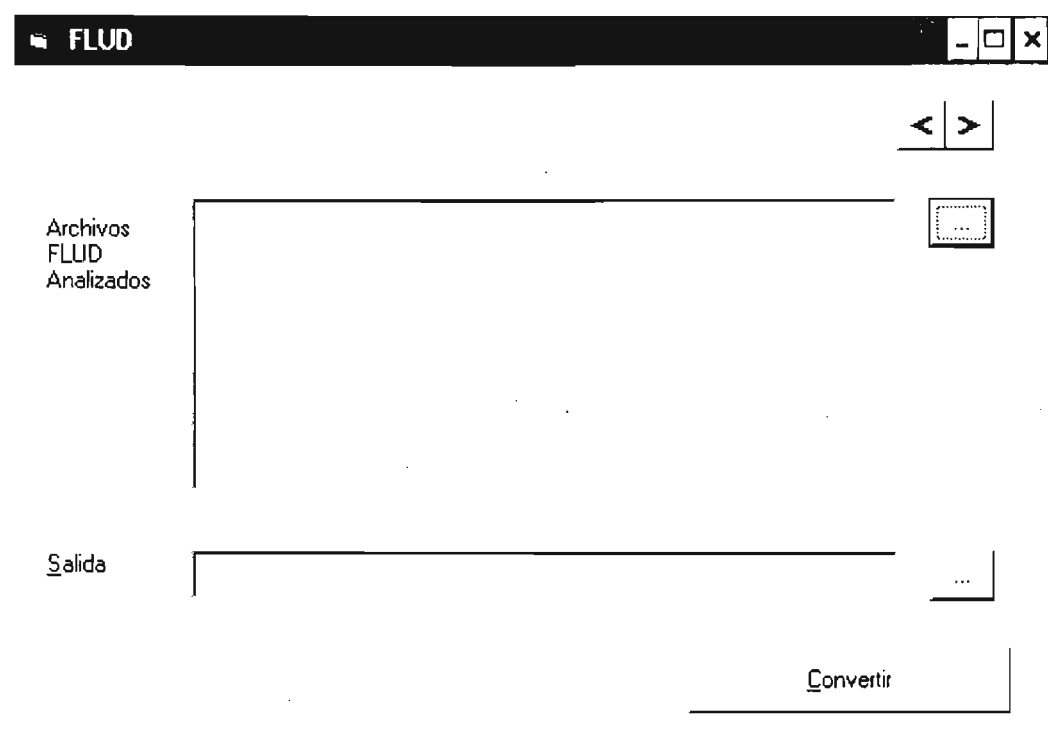


Figura III.11 Ventana principal de CutFLUD

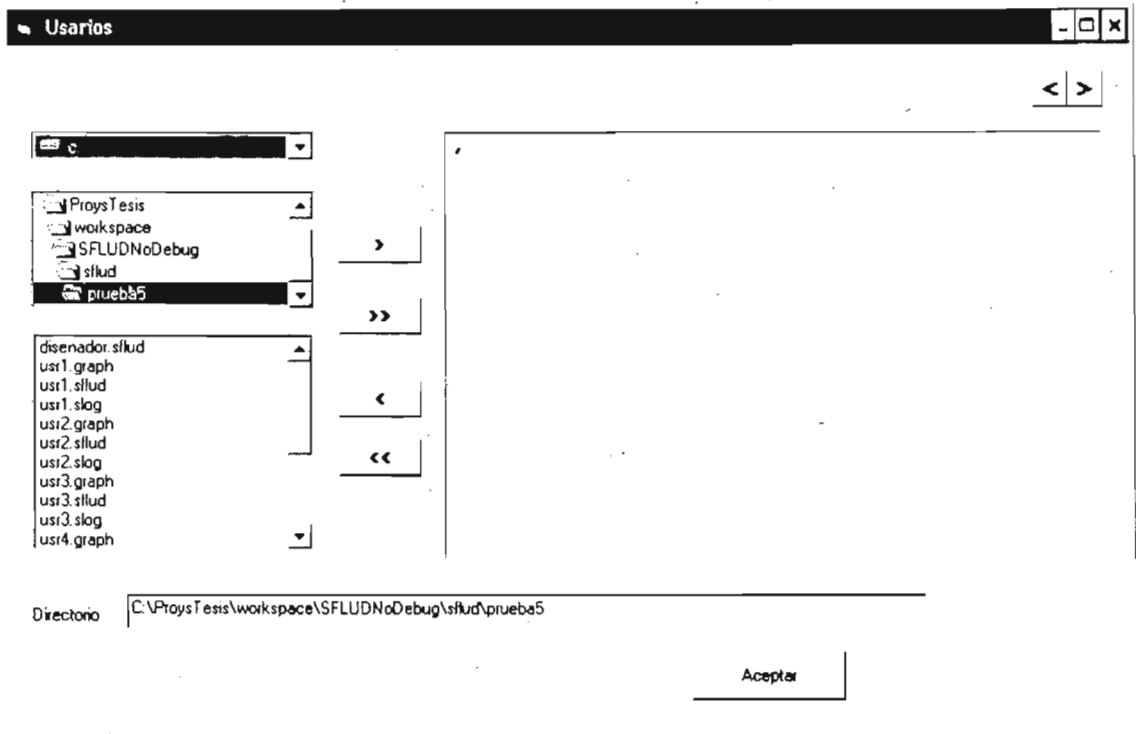


Figura III.12 Cuadro de diálogo para seleccionar los archivos *.parse* que serán analizados

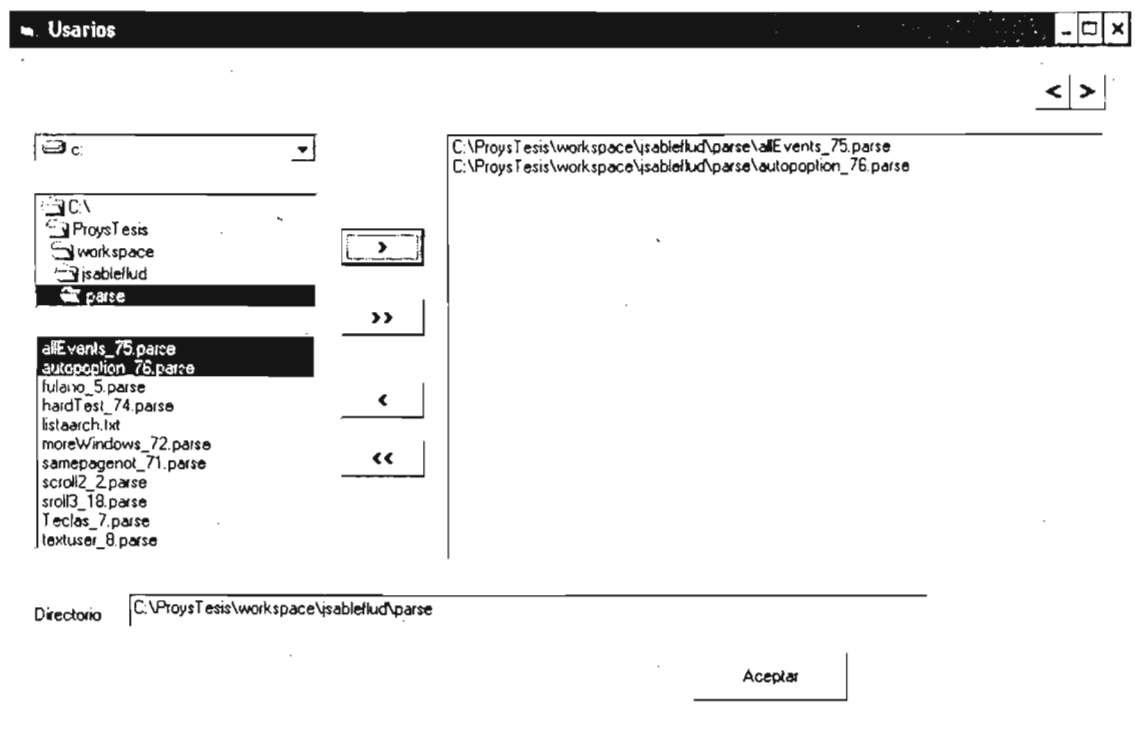


Figura III.13 Cuadro de diálogo para seleccionar los archivos *.parse* que serán analizados



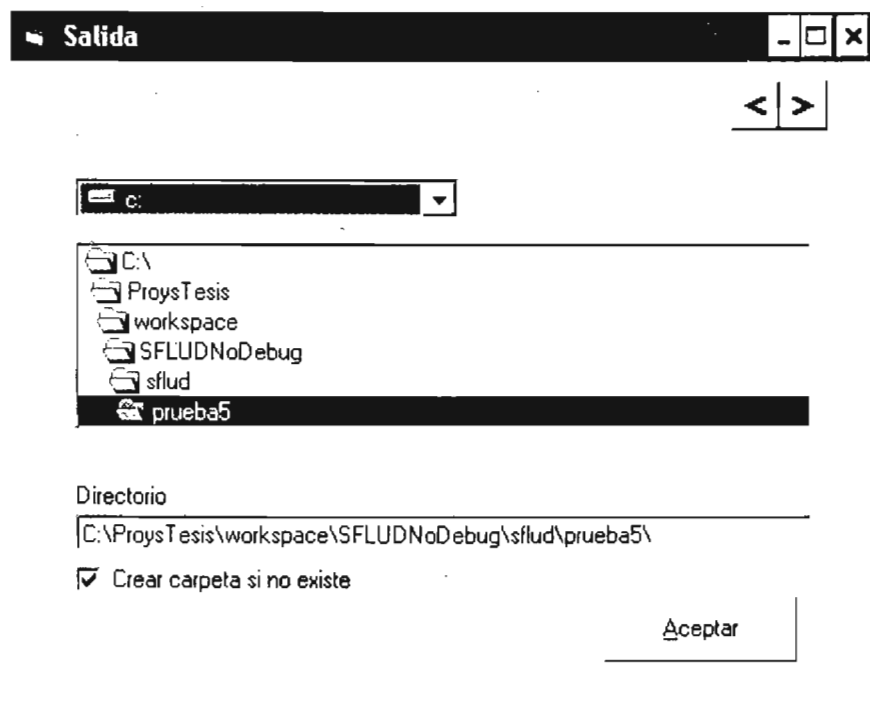


Figura III.14 Cuadro de diálogo para seleccionar el directorio en que se guardarán los archivos *.sflud*

### 3.3.2 Módulo de comparación de secuencias de eventos de interacción

Habiendo capturado los registros de interacción de los usuarios con la interfaz, el siguiente paso consiste en realizar las comparaciones.

Un aspecto a tener en cuenta es que no es indispensable que el diseñador de la interfaz o el experto en usabilidad, hagan una interacción directa con el sistema. Se puede prescindir de ello para describir rutas óptimas, o simplemente rutas que describan un comportamiento del usuario. En este módulo se propone una versión simplificada del formato FLUD.

#### 3.3.2.1 Propuesta de una versión simplificada de FLUD

La versión simplificada de FLUD tiene como objetivo facilitar las tareas de los profesionales de la usabilidad, ya sea el diseñador de la interfaz o el experto en usabilidad. Se les proporciona una forma de especificar secuencias de eventos o rutas de interacción de usuario (posiblemente rutas óptimas) que caractericen determinados comportamientos.

Una de las razones fundamentales por las que se crea un formato simplificado de FLUD es que el formato original recolecta toda la información posible acerca de la interacción. Se incluyen detalles y minucias que muchas veces no son de interés para el profesional de la usabilidad.

A modo de ejemplo baste mencionar que para cada evento registrado en formato FLUD se tienen datos como la hora, minuto, segundo y milésima de segundo en que ocurrió. Para una persona interesada en caracterizar un determinado comportamiento mediante una secuencia de eventos, sería ominoso tener que escribir una serie de amplísimos detalles. Estos tendrían que estar sujetos a la condición de cumplir al pie de la letra con el formato FLUD.

El cuadro de texto III.1 muestra cómo se vería la solicitud de una nueva página en formato FLUD. En contraste, el cuadro de texto III.2 muestra la misma solicitud en formato simplificado.

```
-----  
rectype => L / event  
time => T / 2002/8/14-11:57:19.505  
days => I / 52500  
secs => N / 43039.505  
window => L / main  
url => A / "http://jesse.ncsl.nist.gov/WebLogs/  
Demo/website/index.html"  
event_type#U => L / pointerrelease  
webpage_xy#U => XY / 440,145  
button#U => BU / 2  
widget_ID#W => A / "http://jesse.ncsl.nist.gov/  
WebLogs/Demo/website/registration.html"  
widget_type#W => L / link  
value#W => B / yes  
level#W => L / page  
web_op_type#OP[1] => L / newpage  
newpage_op#OP[1] => L / request  
window_spec#OP[1] => WN / *  
url#OP[1] => A / "http://jesse.ncsl.nist.gov/  
WebLogs/Demo/website/registration.html"  
-----
```

Cuadro de Texto III.1 Ejemplo de evento descrito en formato FLUD

```
NEWPAGE request "http://jesse.ncsl.nist.gov/WebLogs/Demo/website/registration.html"
```

Cuadro de Texto III.2 Ejemplo de evento descrito en formato FLUD Simplificado

La especificación del formato FLUD Simplificado se incluye en el Apéndice A.

### 3.3.2.2 Realización de las comparaciones con ShFLUD

La implementación de este módulo toma como datos de entrada los obtenidos a través del módulo de captura de datos. El flujo de control se muestra en el diagrama de estados de la figura III.15.

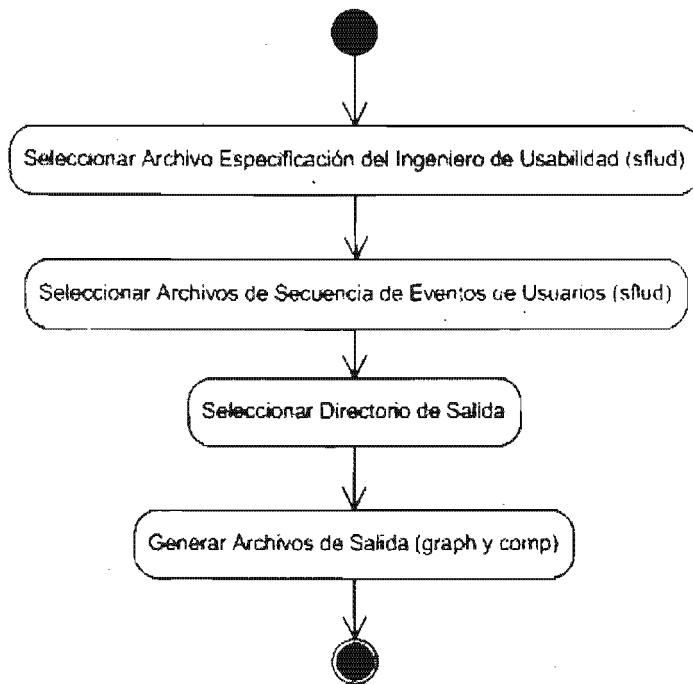


Figura III.15 Utilización de ShFLUD

Dentro de este módulo, el flujo de datos se muestra en la figura III.16.

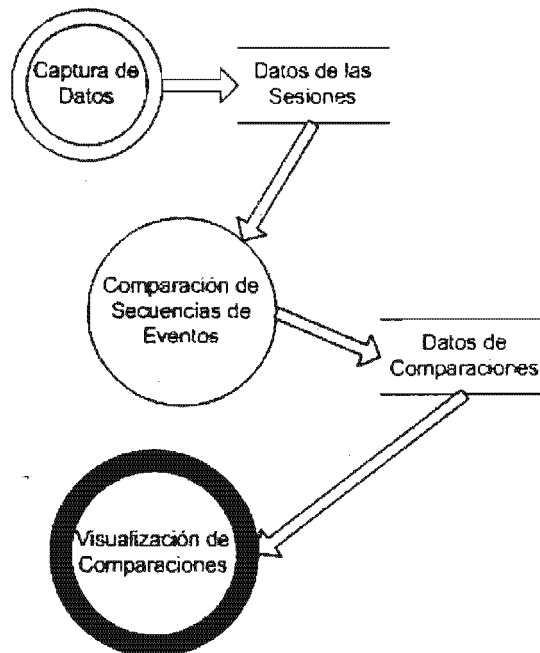


Figura III.16 Flujo de Datos en el Módulo de Captura de Datos

El proceso comienza con la elección de las secuencias de eventos a comparar. Se requieren dos elementos para realizar la comparación:

- a) Un archivo en formato FLUD que describe la interacción del usuario con el sitio instrumentado.

- b) Una especificación hecha por el diseñador o el experto en usabilidad, en que describe cierto comportamiento. El diseñador puede proponer una secuencia exacta de eventos que deben ocurrir, o hacer uso del operador de comodín (\*) que tiene un cometido similar al conocido operador aplicado comúnmente en las expresiones regulares [REGEXP 2005].

El cuadro de texto III.3 muestra la capacidad expresiva que brinda el formato FLUD Simplificado al profesional de usabilidad; se describe el proceso de registro de un usuario en un sitio web. Se ha agregado la numeración al principio de las líneas en cursivas para facilitar su posterior discusión.

---

```
1 *
2 NEWPAGE loading "http://jesse.ncsl.nist.gov/WebLogs/Demo/website/index.html"
3 *
4 POINTPRESS "http://jesse.ncsl.nist.gov/WebLogs/Demo/website/index.html" link
  "http://jesse.ncsl.nist.gov/WebLogs/Demo/website/registration.html" 2
5 *
6 KEYPRESS "http://jesse.ncsl.nist.gov/WebLogs/Demo/website/registration.html" textbox
  "name" e ""
7 KEYPRESS "http://jesse.ncsl.nist.gov/WebLogs/Demo/website/registration.html" textbox
  "name" m "e"
8 KEYPRESS "http://jesse.ncsl.nist.gov/WebLogs/Demo/website/registration.html" textbox
  "name" i "em"
9 KEYPRESS "http://jesse.ncsl.nist.gov/WebLogs/Demo/website/registration.html" textbox
  "name" l "eml"
10 KEYPRESS "http://jesse.ncsl.nist.gov/WebLogs/Demo/website/registration.html" textbox
  "name" e "emil"
11 *
12 POINTPRESS "http://jesse.ncsl.nist.gov/WebLogs/Demo/website/registration.html"
  button "Submit" 1
```

---

#### Cuadro de Texto III.3 Ejemplo de datos en formato FLUD Simplificado

Con las líneas 1 y 2 se especifica que se espera que el usuario, después de realizar cualquier secuencia de eventos, visualice en su navegador o agente la URL <http://jesse.ncsl.nist.gov/WebLogs/Demo/website/index.html>.

Las líneas 3 y 4 expresan que el usuario podrá realizar cualquier serie de eventos, hasta que presione con el botón principal (es decir, el botón 2, según la especificación de FLUD) del puntero del ratón el enlace a la página de registro de usuarios <http://jesse.ncsl.nist.gov/WebLogs/Demo/website/registration.html>.

Las líneas numeradas del 5 al 10 indican que el usuario proporcionará su nombre ("emil") presionando en secuencia las teclas e, m, i y l sobre el control de texto llamado name.

Por último, en las líneas 11 y 12 se describe cómo el usuario, habiendo proporcionado su nombre, y después de realizar cualquier secuencia de eventos finalmente presiona el botón de envío de datos.

#### 3.3.2.3 El proceso de comparación con ShFLUD

La comparación de las secuencias de datos se lleva a cabo de la siguiente manera:

1. Los archivos con extensión *.parse* que contienen la descripción en formato FLUD de la sesión del usuario son leídos desde la herramienta de conversión de formatos. Se genera como salida la versión simplificada de los archivos de entrada: archivos con extensión *.sflud*.
2. Se ejecuta el programa de realización de comparaciones, al que se le indica qué archivos habrá de comparar. La figura III.17 muestra el cuadro de diálogo principal de ShFLUD, en que se muestra que los datos a capturar son:
  - a) Ubicación del archivo del diseñador, mediante el cuadro de diálogo de la figura III.18.
  - b) Listado con la ubicación de cada uno de los archivos de usuario en formato *.sflud*, a través de la ventana de la figuras III.19 y III.20.
  - c) Directorio donde se colocarán los archivos de salida, elegido mediante el cuadro de diálogo de la figura III.21.

Al terminar su ejecución, el programa proporciona:

- Datos de salida enviados a la misma consola desde donde se ejecutó o a un archivo de texto con extensión *.slog* que contiene esos datos, dependiendo de la forma en que se ejecute la aplicación.
- Un archivo con extensión *.graph*, el cual se encuentra en un formato que puede ser leído por el módulo de visualización de comparaciones.

Otra de las herramientas elaboradas como parte de este trabajo de tesis es ShFLUD, un programa que realiza la comparación entre archivos *.sflud* (formato FLUD simplificado), generando como salida dos tipos de archivos: *.graph*, formato para ser leído por la herramienta de visualización; y *.slog*, con una descripción de similitudes y diferencias entre las secuencias comparadas.

Al igual que CutFLUD, la interfaz en modo gráfico de ShFLUD también es muy simple.

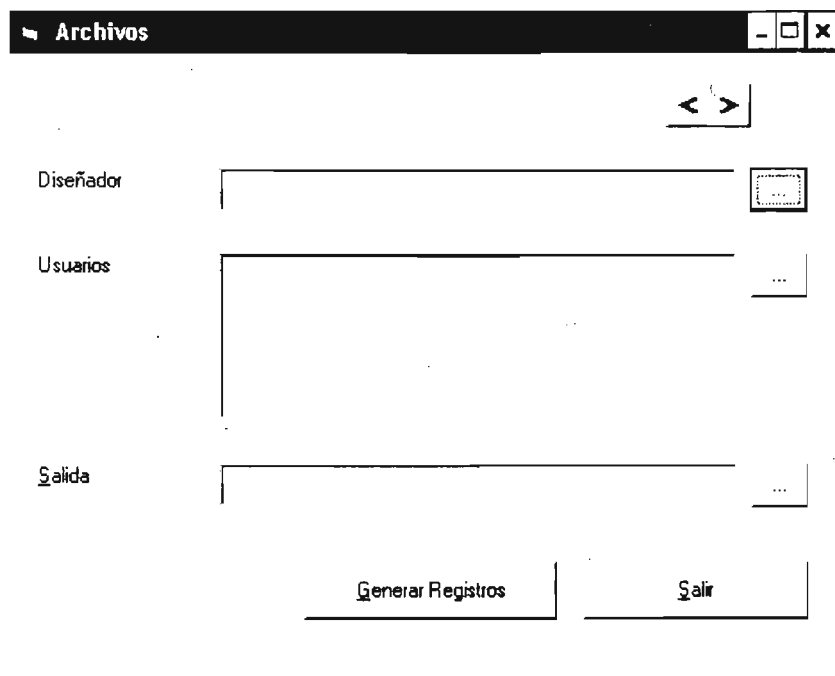


Figura III.17 Ventana principal de CutFLUD

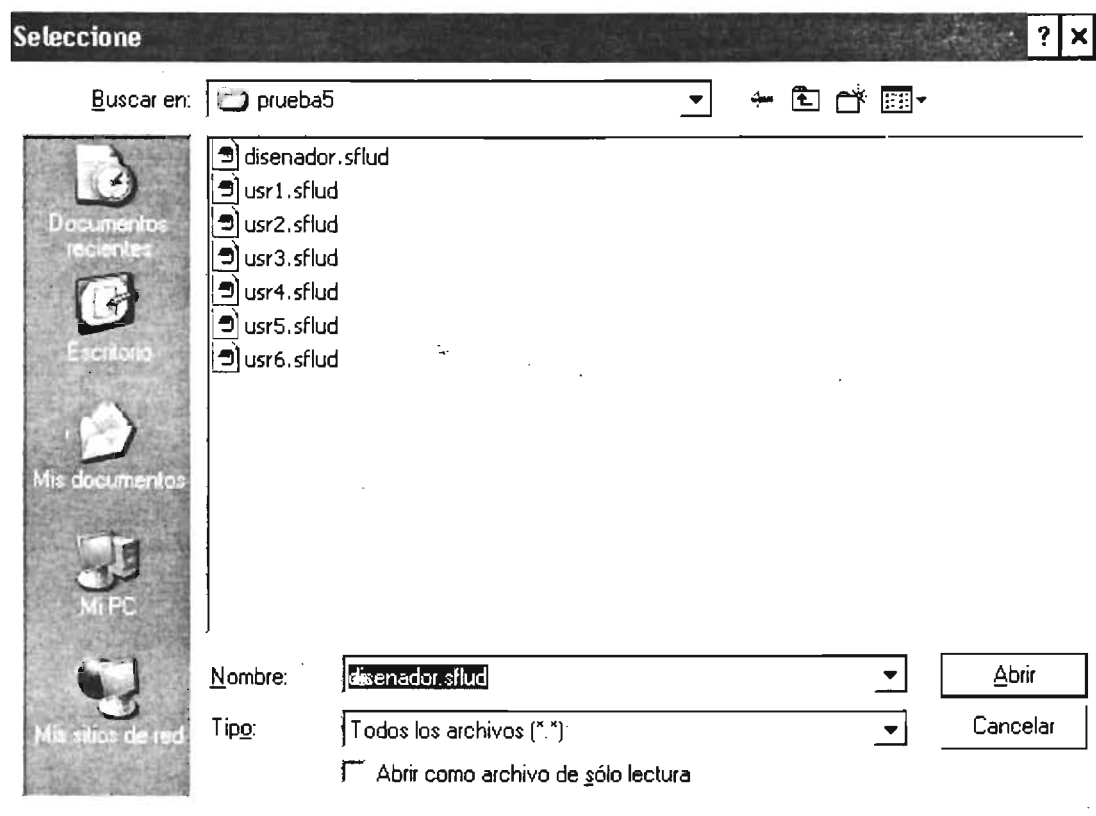


Figura III.18 Cuadro de diálogo para elección del archivo .sflud del diseñador o ingeniero de usabilidad.

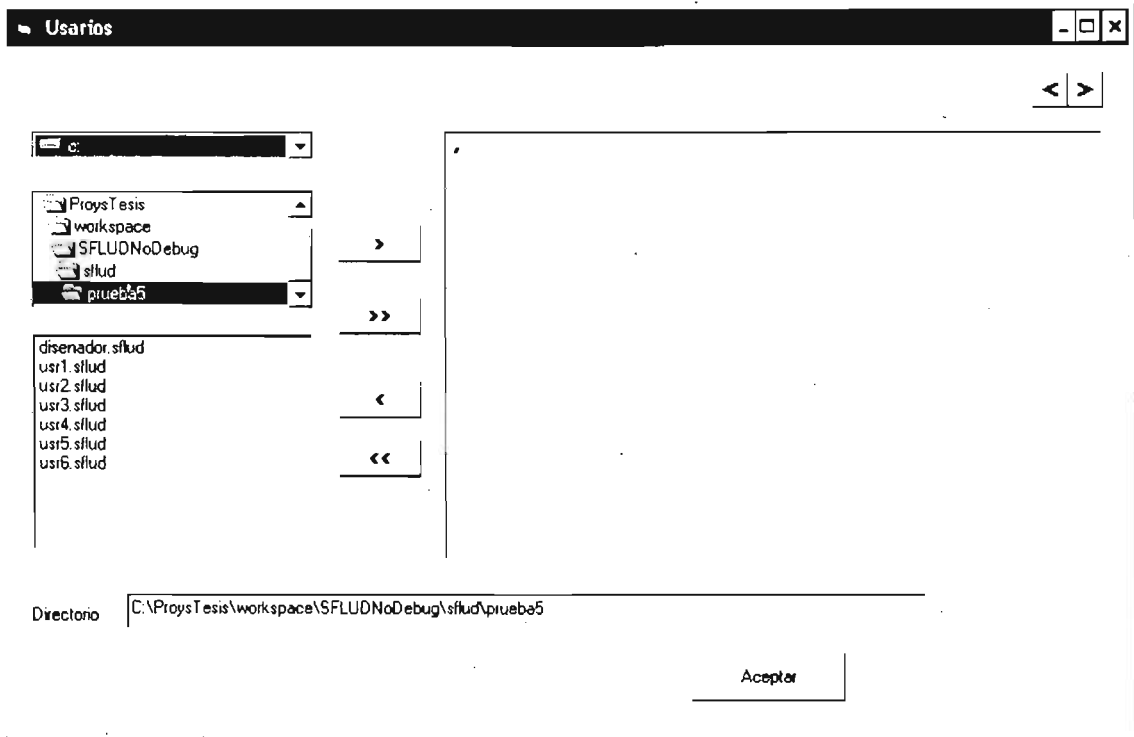


Figura III.19 Cuadro de diálogo para elegir los archivos *.sflud* que contienen la secuencia de eventos de cada usuario en formato FLUD Simplificado

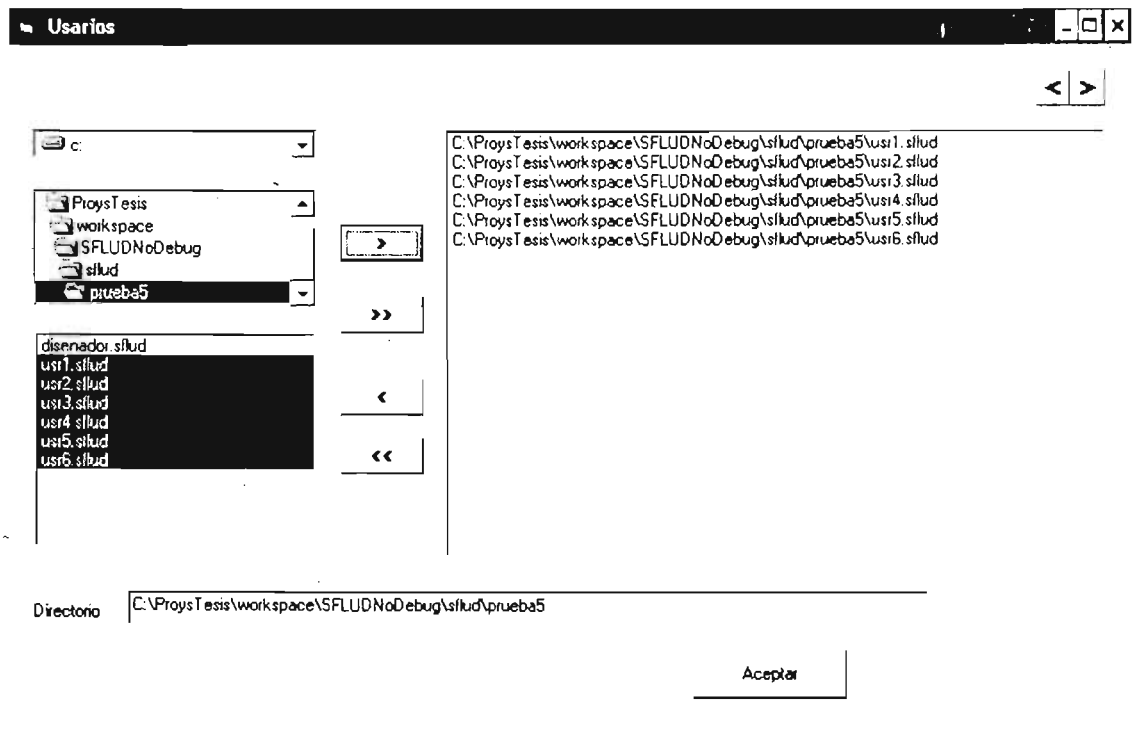


Figura III.20. Cuadro de diálogo para elegir los archivos *.sflud* que contienen la secuencia de eventos de cada usuario en formato FLUD Simplificado

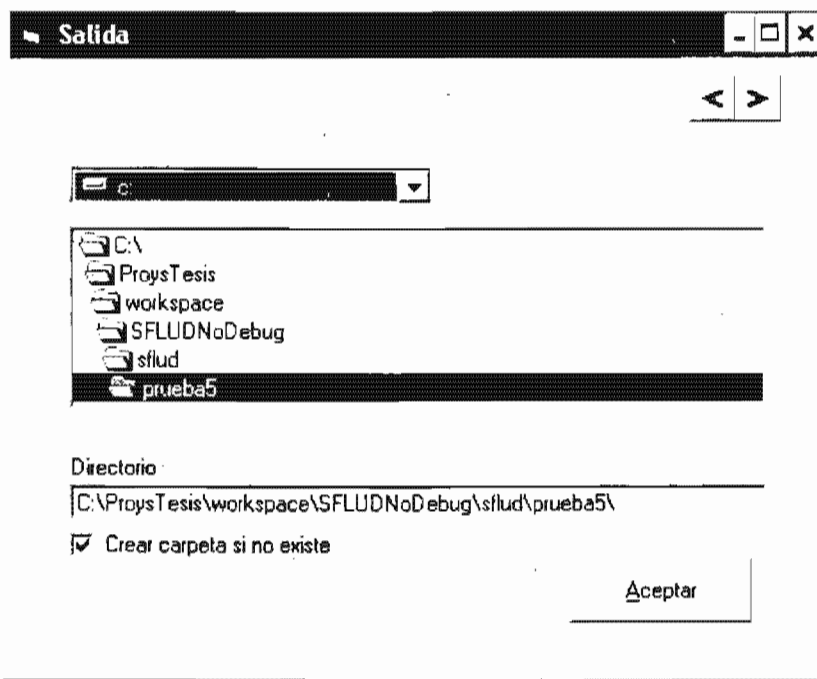


Figura III.21 Cuadro de diálogo para la elección del directorio en que se almacenarán los archivos de salida

### 3.3.3 Módulo de visualización de comparaciones de secuencias de eventos de interacción

El diagrama de la figura IH.22 muestra de manera sencilla el proceso de obtención y procesamiento de datos para la visualización de las comparaciones de secuencias de eventos de interacción de usuarios con la interfaz.

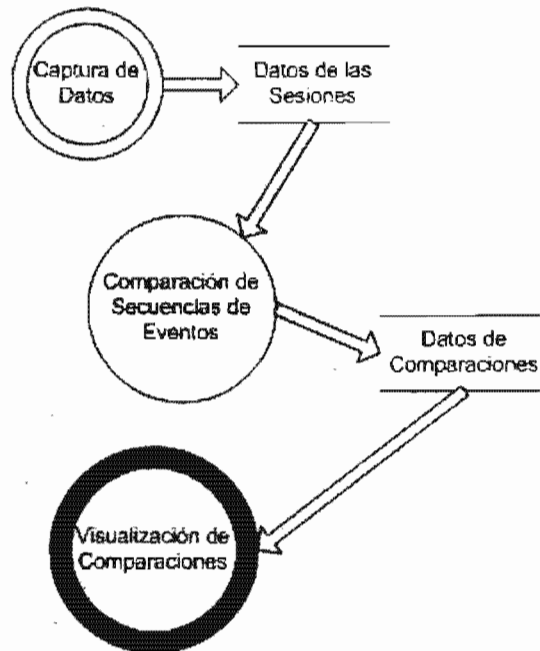


Figura III.22 Proceso de obtención y procesamiento de datos para la visualización



Este módulo toma como entrada un archivo generado por el módulo de comparación de secuencias de eventos. El archivo, que tiene extensión *.graph*, tiene un formato muy sencillo:

- Cada renglón del archivo describe una arista del grafo.
- Cada renglón posee de manera obligatoria dos columnas, que indican los dos vértices de la arista.
- Un renglón puede tener una tercera columna opcional. Representa el peso de la arista, y que sirve para que la herramienta de visualización muestre en colores distintos los tipos de arista que han sido preestablecidos, a saber:
  - Un peso de 0 para aristas de pasos realizados únicamente por el diseñador o experto. Se mostrarán en color azul.
  - Un peso de 100 para aristas de pasos realizados por ambos. Se mostrarán en color violeta.
  - Un peso de 200 para aristas de pasos realizados únicamente por el usuario. Se mostrarán en color rojo.

Para poner en funcionamiento lo detallado al respecto de este módulo se requeriría de un esfuerzo considerable de ingeniería de software, debido a que el programa tendría que resolver asuntos como los siguientes:

- Elección de un sistema de representación de grafos (en su sentido matemático).
- Elección o implementación de un sistema de visualización de grafos.
- Implementación de un algoritmo que visualizara las aristas y vértices de una forma adecuada. Esto es, conseguir una adecuada distribución de los elementos en el espacio de visualización, para evitar, por ejemplo, el poner un vértice exactamente encima de otro. Este problema es conocido como el de distribución o ubicación automática de grafos (Automated Graph Layout) [LOK, 2000].

El interés primordial de este trabajo de tesis no es escribir una solución a los problemas mencionados, que además ya han sido resueltos con anterioridad. Es por ello que se decidió aprovechar la existencia de le programa Biolayout Java, que encaja perfectamente dentro de la arquitectura general del proyecto que nos ocupa.

Las imágenes que se muestran a continuación son visualizaciones presentadas por Biolayout Java proporcionándole como entrada los archivos de comparaciones de secuencias de eventos (archivos con extensión *.graph*).

La figura III.23 muestra el grafo tal cual, sin ningún tipo de leyenda. La utilidad que tiene es el poder ilustrar el flujo de eventos, incluyendo similitudes y diferencias entre las sesiones de dos usuarios. Las flechas indican la secuencia temporal en que ocurren los eventos, iniciando desde el primer evento registrado hasta el último.

En la figura III.24 se presenta la visualización de la misma comparación de secuencias de eventos. La única diferencia consiste en que se ha habilitado la visualización de etiquetas, de manera que se puede ver exactamente a qué evento se refiere cada uno de los vértices.

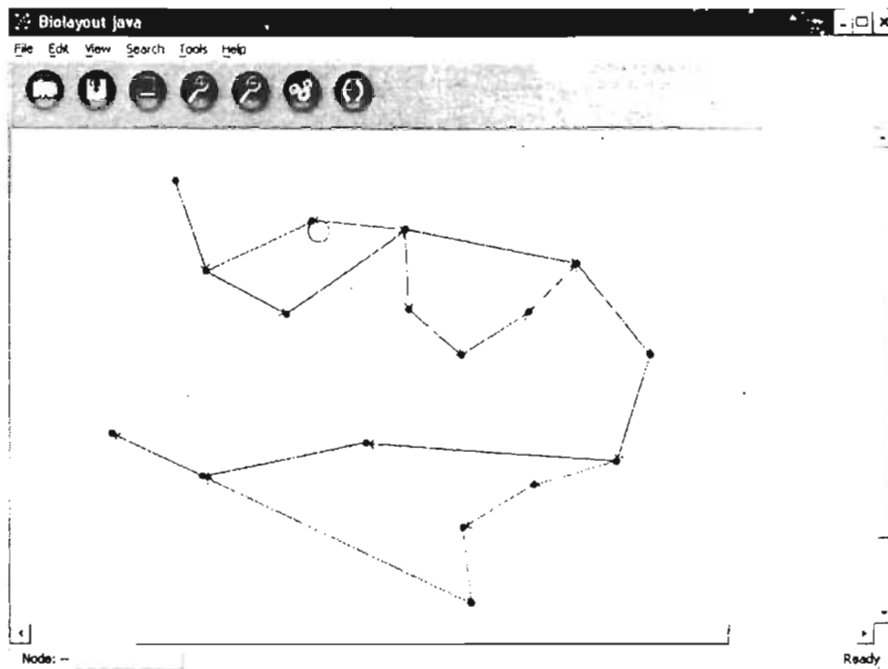


Figura III.23 Grafo que muestra únicamente similitudes y diferencias, sin anotaciones

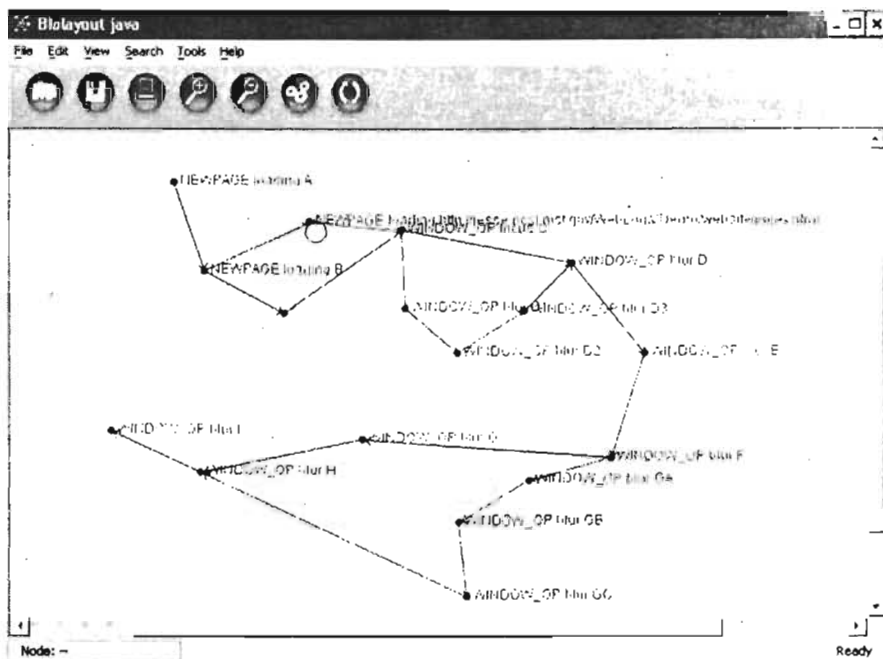


Figura III.24 Grafo de similitudes y diferencias, anotando los datos de cada evento

La figura III.25 muestra cómo la herramienta Biolayout Java es capaz de discriminar las aristas de algún peso específico. Esta característica nos permite de momento eliminar las aristas correspondientes exclusivamente a la ruta del diseñador o experto en usabilidad. Se recomienda al lector interesado en conocer a mayor detalle el funcionamiento de Biolayout Java consultar el manual proporcionado por sus creadores. Dicho mañuela ha sido empaquetado y distribuido de manera conjunta con el software [ENRIGHT 2001].

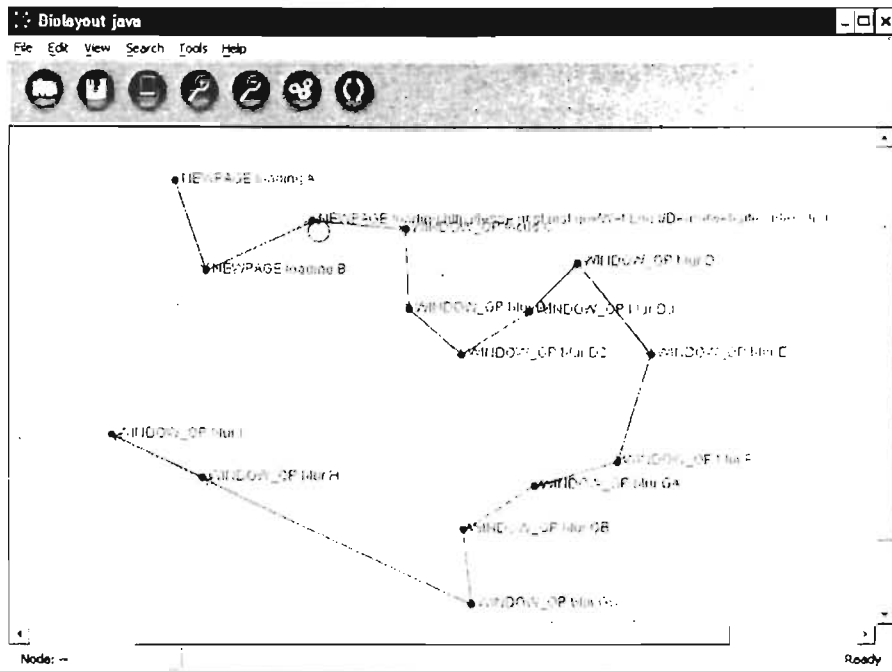


Figura III.25 Grafo que muestra únicamente la secuencia de eventos del diseñador

Las anteriores capturas de pantalla han mostrado ejemplos en que las rutas del usuario y del diseñador confluyen. Las coincidencias indican que en cierta forma el usuario ha logrado cumplir la tarea de principio a fin, y de que al menos tiene un modelo mental de la interfaz semejante al del diseñador. Esto significa esencialmente que el usuario ha identificado el momento en que concluye la tarea que se le ha asignado.

Sin embargo, en las figuras III.26 y III.27 se muestra un ejemplo del caso opuesto, cuando el usuario jamás llega a terminar la tarea de la forma en que lo esperaba el diseñador. Esto sin duda también es un aporte importante para la evaluación de un sitio: saber si los usuarios logran terminar exitosamente sus cometidos.

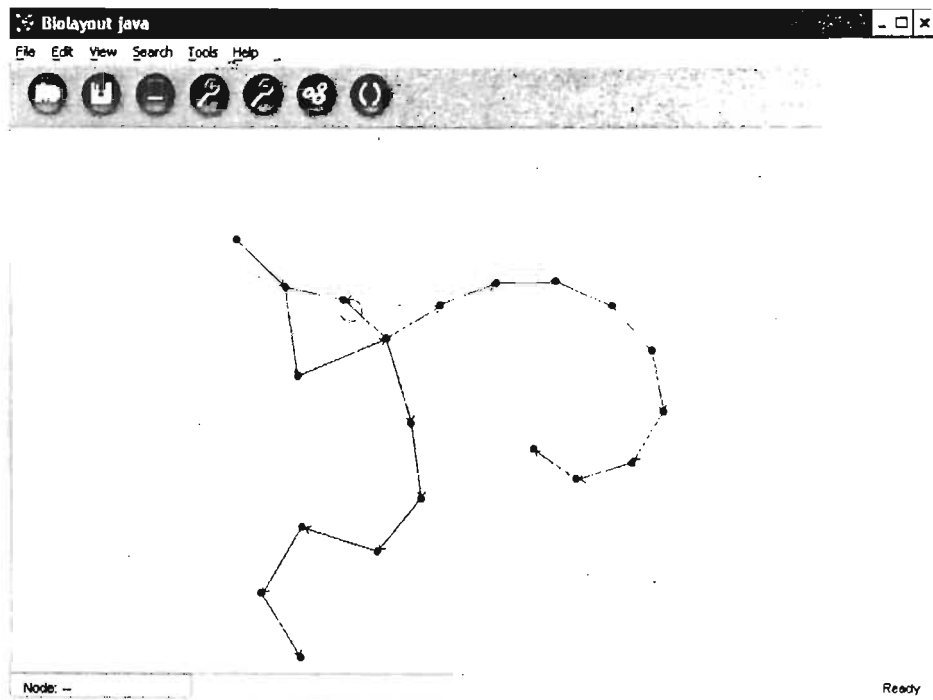


Figura III.26 Grafo de un usuario que nunca terminó la tarea especificada

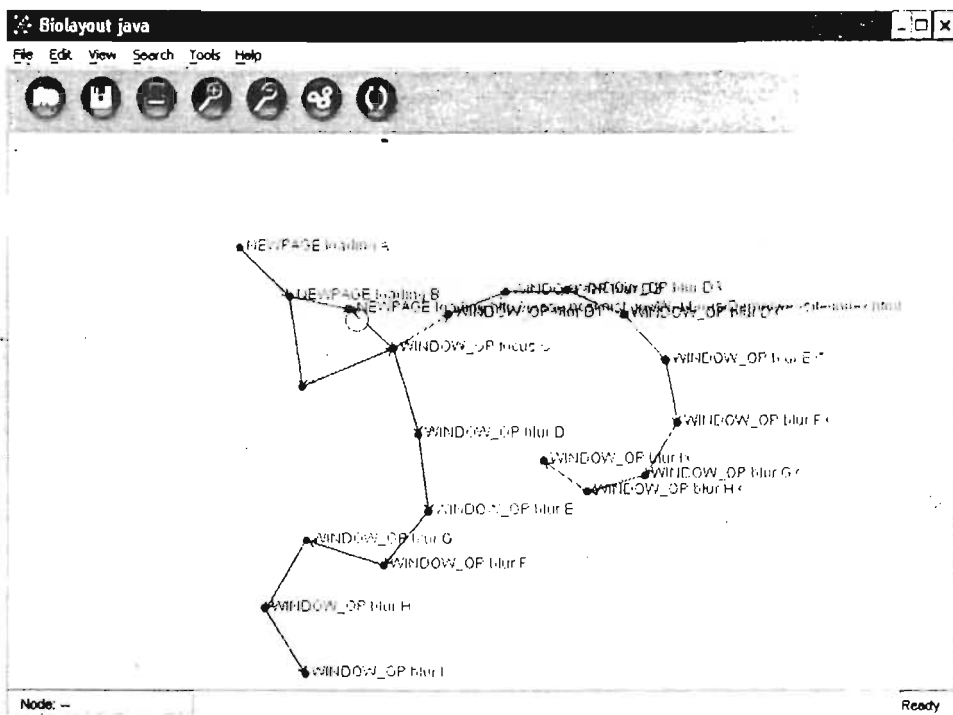


Figura III.27 Grafo de un usuario que nunca terminó la tarea especificada, con anotaciones

A partir de la observación de los grafos comparativos puede obtenerse información valiosa acerca de la interacción del usuario, como se describe en los siguientes puntos:

- Si ambas secuencias de eventos tienen un amplio grado de coincidencias (es decir, gran parte de las aristas se visualizan en color violeta) , se deduce que el usuario ha logrado un alto grado de comprensión acerca de cómo el diseñador de la interfaz esperaba que interactuara.
- La existencia de amplias zonas de divergencia entre ambas secuencias de eventos puede indicar, entre otras cosas:
  - Que el usuario ha elegido una opción inesperada por el diseñador, lo que implica que una de las opciones de navegación no es totalmente clara
  - Que el usuario ha descubierto una forma alterna de llevar a cabo una tarea, y que no se descarta la posibilidad de que esta sea más eficiente,
- Las secuencias no terminan en un punto en común. Esto puede deberse a que para el usuario no es claro el momento en que ha terminado su tarea.
- La secuencia de eventos del usuario tiene ciclos. Esta sería la expresión visible de un usuario que definitivamente no encuentra la opción que busca, lo que nos revela que se requeriría replantear seriamente el esquema de navegación.

### **3.3.4 Detalles de la implementación**

A continuación se presentarán los aspectos detallados de la implementación.

#### **3.3.4.1 Lenguajes de Programación**

Para la implementación general del sistema se eligió el lenguaje de programación Java [SUN JAVA, 2005], debido a sus ya muy conocidas cualidades [WIKIMEDIA, 2005]:

- Soporte de múltiples plataformas de desarrollo y ejecución.
- Soporte del paradigma de programación orientada a objetos.
- Robustez y escalabilidad existente en la mayoría de la implementaciones de la máquina virtual de Java existentes.
- Existencia de una cantidad enorme de desarrollos listos para usarse: estándares, bibliotecas de clases, herramientas, especificaciones, etc.

la versión de máquina virtual de Java utilizada para el desarrollo del presente proyecto fue la incluida Con el kit de desarrollo 1.4.2\_04, proporcionada por la empresa Sun Microsystems: JSDK 1.4.2\_04 [SUN JAVA 1.4.2, 2005].

### 3.3.4.2 Generadores de compiladores

Existen dos módulos que requirieron para su implementación un compilador de compiladores, estos son: el módulo de comparación de secuencias de eventos y el de visualización de comparación de secuencias de eventos.

Se utilizó en programa Sablecc, un programa elaborado por Etienne M. Gagnon bajo la licencia GNU Lesser General Public License [GANNON, 2005].

Sablecc es un generador de Analizadores escrito en Java que tiene las siguientes características

- Generación de analizadores léxicos basados en autómatas finitos determinísticos con soporte total de Unicode y estados léxicos.
- Soporte de la sintaxis extendida de Backus-Naur para gramáticas. (Incluyendo los operadores \*, ? y +)
- Generación automática de árboles de análisis sintáctico con revisión estricta de tipos.
- Generación automática de clases para recorrido de árboles

Sablecc lee la especificación de una gramática en un formato similar al de la forma normal de Backus Naur (BNF). Una vez escrita la gramática deseada, el compilador de compiladores genera el código fuente para el analizador léxico y el analizador sintáctico, en el lenguaje de programación Java.

Teniendo ambos analizadores, se escribe el código adicional para adecuar la funcionalidad proporcionada por Sablecc. Se le agrega el código necesario para realizar las acciones correspondientes a cada uno de los sintácticas que se van detectando al analizar los archivos de secuencias de eventos.

El primer analizador que se construyó usando Sablecc fue el que reconoce la gramática del formato FLUD. A este analizador se le proporciona como entrada un archivo que contiene la captura de todos los eventos de la sesión de un usuario. Como salida arroja el archivo que contiene una versión simplificada de la secuencia introducida, en el formato FLUD Simplificado. De esta manera se cumple con un aspecto primario de la especificación para el módulo de comparación de secuencias de eventos, preparando los datos de entrada para adecuarlos a la realización de la comparación.

El segundo analizador es el correspondiente al módulo de visualización de comparación de secuencias de eventos. Este analizador recibe como datos de entrada los generados por el primero. Brinda como salida la descripción de las similitudes y diferencias entre las dos secuencias proporcionadas, y un archivo de texto que contiene la especificación del grafo que se proporcionará a la herramienta de visualización.

### 3.3.4.3 Herramientas en general

Una de las partes esenciales para la implementación es la suite WebMetrics, de la cual se usó la versión 2.5, disponible gratuitamente para ser descargada a través de la red.

Un aspecto interesante y digno de hacer notar es que todas las partes de esta suite de trabajo han sido puestas en la Internet para ser descargadas, sin haber especificado una licencia específica bajo la cual se liberan. Lo anterior complica determinar los usos aceptables que los creadores le han impuesto. Simplemente se indica que se proporciona el software tal cual, regalándolo a la persona u organización que lo quiera utilizar, modificar y adaptar a voluntad, como una contribución para el desarrollo de la ingeniería de usabilidad.

### 3.4 Conclusión y discusión

En el presente capítulo se describe a detalle todo el proceso de desarrollo llevado a cabo para la obtención de un entorno de trabajo que implemente las ideas fundamentales de esta propuesta de tesis: captura de datos del usuario, análisis de sus acciones, y visualización de comparaciones de secuencias de eventos.

La figura III.28 muestra la correspondencia entre los módulos de la propuesta y los programas que se han usado o implementado para proporcionar la funcionalidad expresada en los requerimientos.

El módulo de captura de datos es implementado en parte por WebVIP, que realiza la captura de datos, y también por CutFLUD, que los adecua y los convierte al formato FLUD Simplificado.

El módulo de comparación de secuencias de eventos tiene toda su funcionalidad en el programa ShFLUD, que realiza la comparación entre diversas secuencias de eventos especificadas en formato FLUD Simplificado.

El módulo de visualización de las comparaciones tiene correspondencia con la herramienta Biolayout Java, que lee la especificación de un grafo y lo presenta en pantalla con diferentes opciones de coloreo y filtrado de datos.

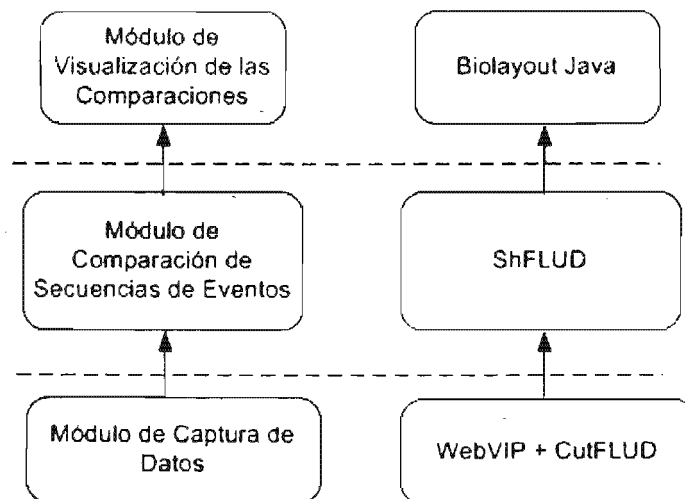


Figura III.28. Correspondencia entre los módulos de la arquitectura general y los programas que los implementan

Se han descrito a suficiente nivel de detalle los aspectos más importantes tomados en cuenta para la obtención de un sistema que será de gran ayuda para los ingenieros de usabilidad que deseen incorporarlo a sus evaluaciones de interfaces con usuarios.



# Capítulo IV

## Resultados

Con el fin de comprobar el correcto funcionamiento de la propuesta de esta tesis se llevó a cabo un caso práctico en el que se tomó el sitio web de una empresa para realizarle un estudio de usabilidad. Se han utilizado en dicho estudio las herramientas aquí presentadas. A lo largo del presente capítulo se describirá a detalle la planeación, desarrollo y análisis de resultados de dicho estudio. Se ha seguido la metodología ya descrita previamente en el capítulo I, en el apartado relativo a las pruebas empíricas con usuarios.

Se realiza la aplicación de pruebas de usabilidad teniendo en mente el poder demostrar de manera práctica que es posible integrar las ideas expuestas en los capítulos anteriores a las técnicas ya conocidas y utilizadas por la comunidad de investigación en interacción humano - computadora.

### 4.1 Plan de pruebas

Para iniciar con la prueba, se solicitó aprobación a la empresa Grupo ideSolex S.A. de C.V. para poder aplicar las pruebas de usabilidad sobre su sitio web. Una vez obtenida su anuencia, se llevaron a cabo los siguientes pasos:

1. Solicitud de autorización por parte de la empresa Grupo ideSolex S.A. de C.V. para poder reproducir los archivos que conforman su sitio web y ubicarlos en un servidor de pruebas. Sobre él se realizará el estudio de usabilidad.
2. Instalación del software requerido en el servidor de pruebas: servidor web, herramienta WebVIP, Analizador de FLUD, así como los programas de que dependen.
3. Instrumentación del sitio web sobre el que se habrán de realizar las pruebas. La instrumentación del sitio web consiste en adecuar los archivos que lo componen, de manera que faciliten la aplicación de las técnicas y herramientas descritas en este trabajo.
4. Elaboración de la documentación necesario para la realización de las pruebas con usuarios:
  - Cuestionario de perfil de usuario (Apéndice B)
  - Protocolo de bienvenida (Apéndice C)
  - Cuestionario de usabilidad (Apéndice D)
  - Especificación de las tareas del usuario (Apéndice E)
  - Descripción de los instrumentos (Apéndice F)

5. Elaboración de la especificación del diseñador. Es un archivo en formato FLUD Simplificado en que el ingeniero de usabilidad plasma la secuencia de eventos correspondiente a las tareas que espera serán realizadas por los usuarios durante su interacción con el sistema.
6. Selección de cinco usuarios para llevar a cabo la prueba. Se invita a cinco personas, ya que a decir de los expertos esta es la cantidad óptima para este tipo de pruebas [NIELSEN, 2000].
7. Desarrollo de una sesión de prueba en la que se comprueba la correcta instalación del ambiente sobre el que trabajarán los usuarios que realizarán las pruebas.
8. Desarrollo de las pruebas con los usuarios seleccionados.
9. Obtención de los archivos que contengan la secuencia de eventos realizados por cada usuario (en formato FLUD Simplificado). Se crean a partir de los archivos en formato FLUD generados durante la interacción de cada usuario con el sistema. Esta tarea se puede realizar a través de la herramienta CutFLUD, en su versión de línea de comando o en su versión de gráfica, lo que facilita este proceso.
10. Realización del análisis de la secuencia de eventos realizados por cada usuario (en formato FLUD Simplificado), comparándolos contra la especificación del diseñador. Para esta operación se puede usar la herramienta ShFLUD, ya sea en su versión gráfica o de línea de comandos. A través de ShFLUD se obtendrá por cada archivo de usuario analizado:
  - Un archivo en el que se registran las similitudes y diferencias entre las secuencias del usuario y del diseñador. La extensión de este archivo es *.slog*.

Un archivo con extensión *.graph* que puede ser leído por la aplicación Biolayout Java, que muestra la representación gráfica de las similitudes y diferencias.

## 4.2 Desarrollo de las pruebas

Las pruebas fueron llevadas de acuerdo al plan previamente elaborado. Se dará una descripción a continuación, respetando la secuencia de la realización de cada una de las etapas, obviando los pasos irrelevantes:

1. La especificación del diseñador de la prueba (Listado IV.1) fue creada por el mismo autor de esta tesis. En ella se vierten los eventos que representan la conclusión exitosa de cada una de las tareas enunciadas en el plan de pruebas, como se muestra en la tabla IV.1. Los archivos de secuencias de eventos de cada uno de los usuarios fueron confrontados con la especificación mencionada. El objetivo era obtener una comparación que indique similitudes y diferencias entre ambas.

```

NEWPAGE loading "http://localhost/WebLogs/prueba/website/index.html"
NEWPAGE loading "http://localhost/WebLogs/prueba/website/historia.htm"
NEWPAGE loading "http://localhost/WebLogs/prueba/website/filosofia.htm"
NEWPAGE loading "http://localhost/WebLogs/prueba/website/tecnologias/"
NEWPAGE loading "http://localhost/WebLogs/prueba/website/productos/pph.htm"
NEWPAGE loading "http://localhost/WebLogs/prueba/website/tut/css/"
WINDOW_OP close "http://localhost/WebLogs/prueba/website/tut/asp/cookies.htm"
NEWPAGE loading "http://localhost/WebLogs/prueba/website/contacto/"

```

#### Listado IV.1 Especificación del Diseñador

En la tabla IV.1 puede observarse como se mapea cada tarea del plan de pruebas con uno o más eventos secuencia correspondiente a la especificación del diseñador.

Tarea del plan de pruebas	Evento de la especificación del diseñador
Abra la página principal del sitio	NEWPAGE loading "http://localhost/WebLogs/prueba/website/index.html"
Lea la historia de la empresa "Grupo iSolex"	NEWPAGE loading "http://localhost/WebLogs/prueba/website/historia.htm"
Encuentre objetivo, misión y visión del grupo iSolex	NEWPAGE loading "http://localhost/WebLogs/prueba/website/filosofia.htm"
Ingrese a la página principal de tecnologías del sitio	NEWPAGE loading "http://localhost/WebLogs/prueba/website/tecnologias/"
Describa las características de PPH	NEWPAGE loading "http://localhost/WebLogs/prueba/website/productos/pph.htm"
Ingrese al tutorial de CSS	NEWPAGE loading "http://localhost/WebLogs/prueba/website/tut/css/"
Ingrese al capítulo llamado "Cookies" del tutorial de ASP	WINDOW_OP close "http://localhost/WebLogs/prueba/website/tut/asp/cookies.htm"
Escriba y envíe su opinión acerca del sitio	NEWPAGE loading "http://localhost/WebLogs/prueba/website/contacto/"
Encuentre el domicilio postal de la empresa "Grupo iSolex"	NEWPAGE loading "http://localhost/WebLogs/prueba/website/contacto/"

Tabla IV.1 Correspondencia entre las tareas del plan de pruebas y los eventos de la Especificación del Diseñador

- Se verificó que para cada prueba de usuario se hubiera generado el archivo `.parse` (datos en formato FLUD) correspondiente al registro de todas sus acciones.
- Se obtuvieron las versiones simplificadas de los registros de acciones para cada prueba (archivos `.sflud`, formato FLUD Simplificado) utilizando la herramienta CutFLUD. Un ejemplo de un archivo en formato simplificado es

el mostrado en el listado IV.2, que es la versión corta de la interacción del primer usuario que presentó la prueba.

```
NEWPAGE loading "http://localhost/WebLogs/prueba/website/index.html"
NEWPAGE request "http://localhost/WebLogs/prueba/website/index.htm"
WINDOW_OP close "http://localhost/WebLogs/prueba/website/index.html"
NEWPAGE loading "http://localhost/WebLogs/prueba/website/index.htm"
WINDOW_OP close "http://localhost/WebLogs/prueba/website/index.htm"
NEWPAGE loading "http://localhost/WebLogs/prueba/website/tecnologias/"
WINDOW_OP close "http://localhost/WebLogs/prueba/website/tecnologias/"
NEWPAGE loading "http://localhost/WebLogs/prueba/website/tut/"
NEWPAGE request "http://localhost/WebLogs/prueba/website/tut/css/"
WINDOW_OP close "http://localhost/WebLogs/prueba/website/tut/"
NEWPAGE loading "http://localhost/WebLogs/prueba/website/tut/css/"
WINDOW_OP close "http://localhost/WebLogs/prueba/website/tut/css/"
NEWPAGE loading "http://localhost/WebLogs/prueba/website/tut/asp/"
NEWPAGE request "http://localhost/WebLogs/prueba/website/tut/asp/cookies.htm"
WINDOW_OP close "http://localhost/WebLogs/prueba/website/tut/asp/"
NEWPAGE loading "http://localhost/WebLogs/prueba/website/tut/asp/cookies.htm"
WINDOW_OP close "http://localhost/WebLogs/prueba/website/tut/asp/cookies.htm"
NEWPAGE loading "http://localhost/WebLogs/prueba/website/contacto/"
WINDOW_OP close "http://localhost/WebLogs/prueba/website/contacto/"
NEWPAGE loading "http://localhost/WebLogs/prueba/website/contacto/"
```

Listado IV.2 Secuencia de Eventos en Formato FLUD Simplificado, usuario 1

4. Se llevaron a cabo las comparaciones entre las secuencias de usuarios y la especificación del diseñador, usando la herramienta ShFLUD.
5. Se realizó la visualización de los archivos *.graph* (obtenidos durante el proceso de comparación) a través de la herramienta Biolayout Java. Este punto se trata a mayor detalle en un apartado posterior.

### 4.3 Interpretación de los resultados

Al realizar y visualizar las comparaciones de secuencias, se obtiene un grafo para cada prueba. En las imágenes posteriores, las aristas en color azul representan el recorrido del diseñador o ingeniero de usabilidad. Aquellas en color rojo son las recorridas por el usuario únicamente, en tanto que las de color violeta indican las rutas en que el usuario y el diseñador han coincidido.

#### 4.3.1 Caso del Usuario 1

A continuación se trata a detalle el caso específico para el primer usuario, y posteriormente se describe de manera general los cuatro restantes. Los datos de visualización para poder obtener el grafo de las figuras IV.1 y IV.2 provienen de la comparación entre la especificación del diseñador (listado IV.1) y la secuencia de eventos de interacción del usuario 1 (listado IV.2).

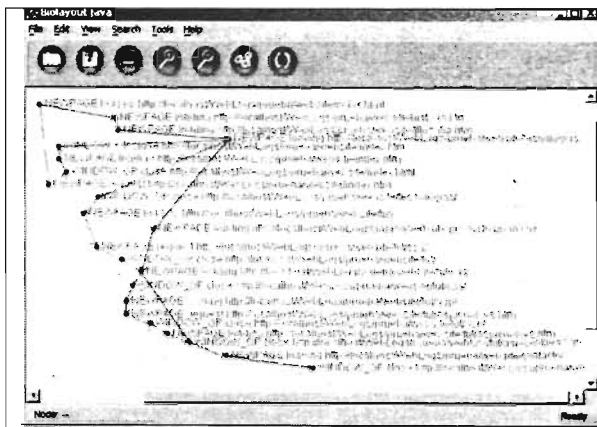


Figura IV.1 Visualización de comparación de secuencias, usuario 1 contra diseñador, con anotaciones

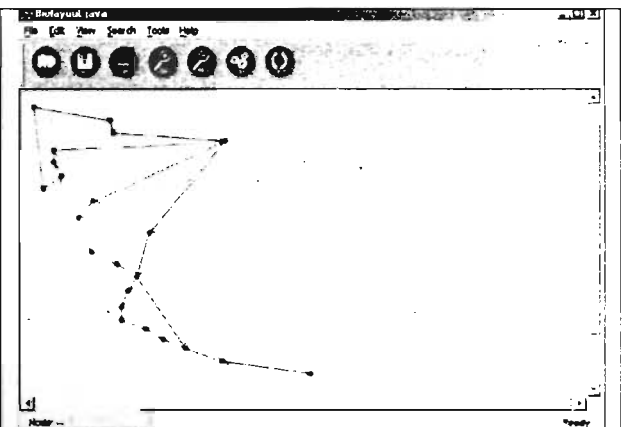


Figura IV.2 Visualización de comparación de secuencias, usuario 1 contra diseñador

A partir de analizar la figura IV.1 se pueden emitir las siguientes aseveraciones:

- El flujo de interacción inicia en un punto en común, la página de inicio; con ello se observa que el usuario cumplió con la primera tarea asignada: "Abra la página principal del sitio".
- No se cumple con la tarea 2: "Lea la Historia de la empresa «Grupo iSolex»". Esto se infiere a partir del hecho de que la ruta del usuario no toca el nodo correspondiente a la página historia.htm, que es la que debería haber sido accedida para poder cumplir con la tarea.
- No se cumple con la tarea 3: "Encuentre el objetivo, misión y visión de «Grupo iSolex»". Se deducido del hecho de que la ruta del usuario no toca el vértice correspondiente a la página filosofia.htm, que debería haber sido accedida para poder cumplir con la tarea.
- Las rutas se reúnen en el nodo que representa a la página tecnologías.htm, lo que indica que se cumplió con la tarea 4, que reza: "Ingrese a la página principal de tecnologías del sitio".
- La ruta del usuario no pasa nunca por la página pph.htm, condición para considerar cumplida la tarea 5, por lo que podemos decir que al no poder encontrar dicho nodo, no se ha cumplido con la tarea.
- En la ruta del usuario indica que este solicita la página correspondiente al curso de CSS, por lo que sí se cumple con la tarea 6.
- Para cumplir con la tarea 7, "ingresar a capítulo titulado Cookies del curso de ASP", ha visualizado la página cookies.htm, por lo que se ha cumplido con la tarea 7. Es de hacer notar que el usuario ha solicitado varias páginas antes de poder acceder a la que debía. Esto puede significar que en primera instancia no es evidente para él cuál de las opciones le lleva a la página adecuada. El origen del problema de usabilidad podría ser una falla en el

esquema de navegación, en la visibilidad de las opciones, o en el nombrado de las mismas.

- El usuario ha ingresado a la sección correspondiente a la página de contacto, cumpliendo con la última de las tareas.
- Las rutas del diseñador y del usuario se unen en un nodo final común, por lo que se puede decir que el usuario sabe que ha terminado la prueba exitosamente.

### 4.3.2 Usuarios restantes

Realizando un análisis semejante al anterior para cada uno de los usuarios, se pueden elaborar las siguientes conclusiones acerca de su interacción:

- Dos usuarios no realizaron la tarea 2: "Lea la historia de la empresa Grupo iSolex", pues creyeron erróneamente haber cumplido con dicha tarea al abrir la página principal del sitio.
- Dos usuarios no realizaron la tarea 3: "Encuentre objetivo, misión y visión de Grupo iSolex", pues creyeron erróneamente haber cumplido con dicha tarea al abrir la página principal del sitio.
- Un usuario no cumplió con la tarea 5, debido a que no logró información acerca del producto PPH.

## 4.4 Reporte general de usabilidad

En este apartado se proporciona el reporte final que acompaña a todo estudio de usabilidad.

### 4.4.1 Descripción de la prueba

La prueba consistió de nueve tareas que debían ser llevadas a cabo por cada uno de los usuarios, en un intervalo máximo de tiempo. En la tabla IV.2 se enumeran las tareas de que se componía la prueba, y una justificación para su inclusión.

#	Título de la Prueba	Justificación	Tiempo max.
1	Abra la pagina principal del sitio	Se busca que el usuario identifique la forma de acceder al sitio web.	30 segs.
2	Lea la historia de la empresa "Grupo iSolex"	Se desea saber si la sección de historia del sitio puede ser identificada y accedida por los usuarios fácilmente	30 segs.
3	Encuentre objetivo, misión y visión del grupo iSolex	Se desea saber si la categorización del objetivos, misión y visión ha sido correctamente clasificados dentro de la sección de filosofía, y si estos puntos pueden ser identificados y accedidos por los usuarios fácilmente	30 segs.

4	Ingrese a la página principal de tecnologías del sitio	Se busca saber si la sección de historia del sitio puede ser identificada y accedida por los usuarios fácilmente.	30 segs.
5	Describa las características de PPH	Se desea saber si los usuarios logran identificar la ubicación y características de los productos de la empresa, y de manera específica PPH. Este punto puede cobrar relevancia si dichos productos comienzan a ser comercializados mediante este mismo sitio.	30 segs.
6	Ingrese al Tutorial de CSS	El interés radica en conocer si es claro a los usuarios la forma de acceder a los manuales, y específicamente al de Hojas de Estilo en Cascada (CSS).	30 segs.
7	Ingrese al capítulo llamado "Cookies" del tutorial de ASP	Se busca conocer si es claro a los usuarios la forma de acceder a los manuales, y verificar que el esquema de navegación dentro de cada uno de los manuales es adecuado.	30 segs.
8	Escriba y envíe su opinión acerca del sitio	Se desea saber si la sección de contacto del sitio puede ser identificada y accedida por los usuarios fácilmente, y si existe algún problema de usabilidad dentro del proceso de envío de información del cliente al momento de llenar el formulario correspondiente.	3 min.
9	Encuentre el domicilio postal de la empresa "Grupo iSolex"	Se requiere saber si la ubicación del texto que contiene el domicilio postal de la empresa es intuitiva a los usuarios.	30 segs.

Tabla IV.2 Tareas que conforman la prueba

#### 4.4.2. Validación de la prueba

La prueba fue exitosa, debido a que a pesar de que no todos los usuarios pudieron realizar las tareas propuestas en el tiempo establecido, sí cumplieron con la mayoría. Adicionalmente, se obtuvieron datos valiosos que ayudarán a implementar los cambios que corrijan los defectos encontrados en la interfaz.

##### 4.4.2.1. El sistema

El sistema en cuestión es el sitio web de la compañía Grupo idaSolex S.A. de C.V., que también se da a conocer como Grupo iSolex. La versión bajo prueba es la 1.0, que puede ser accedida desde la ubicación siguiente: <http://www.solex.com.mx/> [ISOLEX 2005].

El sitio es la imagen de la empresa en Internet; en él se enuncian los servicios y productos ofrecidos por la empresa, y se exponen los valores que rigen a esta.

Para poder realizar la prueba se copiaron todos los archivos que componen al sitio, de manera que pudieran sufrir ligeras modificaciones y así hacerlos aptos

para poder aplicarles las prueba. La instalación del sistema en que se realizaron las pruebas se hizo sobre una laptop IBM T21, con microprocesador Intel Pentium III y 256 MB en memoria RAM. El sistema operativo de la computadora es Windows XP, y tiene instalado la versión más reciente de WebMetrics y todas sus dependencias: un intérprete de lenguaje Perl, y el servidor Microsoft Internet Information Services.

El equipo mencionado en el párrafo anterior sirvió a la vez como cliente y servidor de la aplicación. Se descarta que el uso de una computadora portátil para la realización de las pruebas haya tenido un impacto negativo sobre el desempeño de los usuarios, ya que a decir de ellos, cuentan con experiencia en el manejo de este tipo de hardware.

#### 4.4.2.2. Los participantes

Para llevar a cabo las pruebas fue necesario elegir usuarios. Se buscaba para ello a personas que tuvieran cierta experiencia con el manejo de computadoras, especialmente en el uso de Internet, debido a la naturaleza del sistema bajo prueba. Otra característica que se buscaba de los usuarios es que pudieran tener algún interés o relación con la temática del sitio. Por ello se eligió a un grupo de estudiantes de Ingeniería en Computación, que ciertamente pueden estar interesados en las propuestas y cursos de una empresa de tecnologías de la información como es Grupo iSolex. La tabla IV.3 muestra un breve resumen de las características de los cinco usuarios que participaron dentro de la prueba; los datos fueron recabados a través del cuestionario de perfil de usuario.

Usuario	Nivel de Estudios	Experiencia usando computadoras	Experiencia usando Internet
1	Estudiante Ingeniería en Computación	6 años	2 años
2	Estudiante Ingeniería en Computación	3 años	3 años
3	Estudiante Ingeniería en Computación	5 años	2 años
4	Estudiante Ingeniería en Computación	6 años	3 años
5	Estudiante Ingeniería en Computación	3 años	1 año

Tabla IV.3 Características de los participantes de la prueba

#### 4.4.3 Resultados

Los resultados de la prueba fueron satisfactorios, aunque no necesariamente halagadores para el desarrollador del sistema, debido a que cumplieron su cometido: sacaron a la luz algunos errores que tenía la interfaz y que es necesario corregir:

- El esquema de navegación no es intuitivo para algunos usuarios.
- La clasificación de los contenidos del sistema debe ser revisada, pues algunos errores de los usuarios se deben a su búsqueda de información en categorías equivocadas..



#### 4.4.3.1. Análisis de las pruebas

De acuerdo a la especificación de los instrumentos de la prueba vemos que algunas de las pruebas a que fue sometido el sistema no fueron aprobadas. El problema que se presentó es que los fueron confundidos por un esquema de navegación que no tiene claros algunos mensajes y categorías.

#### 4.4.3.2. Calificación del sistema

Tomando como base el cuestionario de usabilidad aplicado a los usuarios, podemos obtener el promedio de las calificaciones emitidas por estos, que se muestran a continuación, discutiendo aquellos puntos que resulten de interés:

1. Creo que me gustará usar el sistema frecuentemente  
Calificación Promedio = 4.2
2. Encuentro el sistema innecesariamente complejo  
Calificación Promedio = 2.8  
El alto promedio probablemente se debe a que las categorías del sistema no fueron claras para los participantes
3. Pienso que el sistema era fácil de usar  
Calificación Promedio = 4.2
4. Creo que necesitaré la ayuda de un técnico para poder usar el sistema  
Calificación Promedio = 1.4
5. Encontré que las distintas funciones del sistema están bien integradas  
Calificación Promedio = 4.2
6. Pienso que había mucha inconsistencia en el sistema  
Calificación Promedio = 3.2  
El resultado probablemente se debe a que algunas categorías tienen subcategorías y otras no, de manera que no se distribuyen adecuadamente
7. Me imagino que la gente aprenderá a usar el sistema bastante rápido  
Calificación Promedio = 4.4
8. Encuentro al sistema engorroso de usar  
Calificación Promedio = 1.6
9. Me sentí muy seguro usando el sistema  
Calificación Promedio = 4.4
10. Necesito aprender muchas cosas antes de poder utilizar el sistema  
Calificación Promedio = 1.6

#### 4.4.4 Conclusiones

Se encontraron tres problemas, que son descritos a mayor detalle en la tabla IV.4, que además discute las causas probables, y enuncia propuestas de solución.

Problema	Causas Probables	Solución
El usuario no concluyó la tarea 2: "Lea la historia de la empresa Grupo iSolex", pues creyó erróneamente haber cumplido con dicha tarea al abrir la página principal del sitio.	<ul style="list-style-type: none"> <li>- La opción de "Historia" del menú principal no es visible de manera inmediata a los usuarios que se encuentran en la página principal del sitio. La causa de la problema consiste en que el texto de la página principal da una descripción breve de la empresa, y dicho contenido puede ser confundido con la historia la organización.</li> <li>- El esquema de navegación no es intuitivo, se encuentra mal organizado, debido a que la sección de historia no se encuentra en donde el usuario espera que esté.</li> </ul>	<ul style="list-style-type: none"> <li>- Realizar una revisión del esquema de navegación, asignando nombres más adecuados a las opciones. Para ello se recomienda el uso de la técnica de de "ordenación de cartas" con usuarios, para asegurar que la categorización sea corregida [PEARROW 2000]</li> <li>- Incluir dentro del texto de la página principal un enlace a la sección de Historia, de manera que la diferencia entre ambas se haga evidente</li> </ul>
El usuario no realizó la tarea 3: "Encuentre objetivo, misión y visión de Grupo iSolex", pues creyó erróneamente haber cumplido con dicha tarea al abrir la página principal del sitio.	<ul style="list-style-type: none"> <li>- La opción de "Filosofía" del menú principal no es visible de manera inmediata a los usuarios que se encuentran en la página principal del sitio. La causa de la problema consiste en que el texto de la página principal da una descripción breve de la empresa, y dicho contenido puede ser confundido con la filosofía organización.</li> <li>- El esquema de navegación no es intuitivo, se encuentra mal organizado, debido a que los usuarios no esperan que el objetivo, misión y visión se encuentren dentro de la categoría de filosofía.</li> </ul>	<ul style="list-style-type: none"> <li>- Realizar una revisión del esquema de navegación, asignando nombres más adecuados a las opciones. Para ello se recomienda el uso de la técnica de de "ordenación de cartas" con usuarios, para asegurar que la categorización sea corregida [PEARROW 2000]</li> <li>- Incluir dentro del texto de la página principal un enlace a la sección de Filosofía, mencionando que el objetivo, misión y visión de la empresa se encuentra dentro de esa sección. De esta manera la diferencia entre ambas se hará evidente</li> </ul>
Un usuario no cumplió con la tarea 5, debido a que no logró encontrar información	<ul style="list-style-type: none"> <li>- La información para encontrar los productos ofrecidos por iSolex no se encuentra en un lugar relevante, no está en el primer plano que le corresponde</li> </ul>	<ul style="list-style-type: none"> <li>Los productos que vende una empresa deben tener un lugar privilegiado dentro del sitio web. - Se debe resaltar la opción del esquema de navegación relativa</li> </ul>

acerca del producto PPH.		a los productos - Se debe colocar en un lugar privilegiado de la pantalla un acceso adicional a la sección de productos, mencionando inclusive algunos de ellos si el espacio lo permite
-----------------------------	--	---

Tabla IV.4 Problemas encontrados durante las pruebas, y propuestas de solución

#### 4.4 Conclusión y discusión

A lo largo de este capítulo se ofreció una disertación en que se cubrió a detalle la forma en que se aplicaron las técnicas y herramientas propuestas en esta tesis a una evaluación de interfaces mediante la realización de pruebas con usuarios.

La importancia de los resultados aquí expuestos radica en que se ha demostrado la posibilidad de integrar las ideas aquí expuestas a las técnicas ya conocidas y utilizadas por la comunidad de investigación en interacción humano - computadora.

# Capítulo V

## Conclusiones y perspectivas

A lo largo del presente trabajo de tesis se expuso y sustentó una propuesta que, tal como se describe en la introducción, consiste básicamente de lo siguiente:

*“La automatización de los procesos de recolección de datos y análisis de las acciones del usuario, en el ámbito de las pruebas con usuarios”*

Teniendo en mente lo anterior se han elaborado técnicas y herramientas que permiten al ingeniero de usabilidad llevar a cabo de manera automatizada el proceso de recolección de datos acerca de la interacción de un usuario con una interfaz. Más importante aún es el análisis que se hace de tales datos a fin de obtener información relevante orientada al descubrimiento de fallas y patrones de usabilidad en una interfaz, y por ende, a su mejora.

El trabajo comienza con un breve recuento de las características, ventajas y desventajas de la evaluación de interfaces de usuario: el tipo de pruebas que existen, clasificación, aplicabilidad en determinadas circunstancias, problemas que solucionan y retos que presentan

Posteriormente se realiza una revisión de trabajos relacionados, elaborados por investigadores del área de interacción humano – computadora. El objetivo de lo anterior es conocer las oportunidades de aportar ideas nuevas que no choquen con las existentes, sino que se adapten a ellas de forma tal que puedan tener una acogida apacible entre los profesionales que realizan pruebas de usabilidad. Se descubrió que en otras ocasiones ya había sido explorada la automatización de evaluación de interfaces de usuario. Sin embargo no todo está dicho en la materia, especialmente hablando de interfaces web, a las que se orienta esta tesis.

### **5.1 Automatización del proceso de evaluación de interfaces de usuario**

Este trabajo ha aprovechado efectivamente las oportunidades de automatización del proceso de realización de pruebas con usuarios en tres etapas importantes, a saber:

#### a) Obtención de datos del usuario

A través de la interacción de un usuario con una interfaz se genera una gran cantidad de datos relacionados con cada una de las acciones llevadas a cabo por él. La captura de tales datos es un elemento imprescindible durante la evaluación de una interfaz, ya que una vez adquiridos pueden ser analizados según las metas que se tengan en mente.

El presente trabajo si bien no se enfocó a la automatización de captura de datos, si utilizó para ello la herramienta WebVIP, elaborada por el NIST(National Institute of Standards and Technology). WebVIP funciona agregando instrucciones de recolección de datos al código fuente de interfaces web. Cuando un usuario se encuentra trabajando con la interfaz, esta va acumulando un registro de todos los eventos realizados. Al finalizar la sesión el registro se grabará en un archivo. El formato con el que se almacenan los datos se llama FLUD. Tal formato tiene la ventaja de ser promovido por el propio NIST, que es una organización promotora de los estándares. Se da el caso de que a pesar de que FLUD no es una norma o estándar en sí, bien puede ser adoptado como un estándar *de facto* por la comunidad de ingenieros de usabilidad. Esto redundo en grandes beneficios como son la capacidad de intercambio de datos de prueba entre colegas, o la construcción de herramientas que analicen esos datos organizados en un formato universalmente reconocido.

El formato FLUD puede expresar fielmente muchos detalles de la interacción del usuario, sin embargo en sesiones de prueba largas se pueden tener registros enormes que para los humanos resultan difíciles de analizar.

El aporte de esta tesis respecto a esta etapa consiste en proponer una versión simplificada del formato FLUD que puede ser usada para describir la interacción de un usuario en unas cuantas líneas. Se elaboró la especificación del formato FLUD Simplificado, y una herramienta llamada CutFLUD que convierte archivos de formato FLUD común a formato FLUD Simplificado.

#### b) Análisis de las acciones del usuario

El punto central de esta tesis consistió en realizar un análisis de las acciones de usuario, llevado a cabo de la siguiente manera:

1. El ingeniero de usabilidad a cargo de la prueba elabora una lista de las tareas que harán los usuarios.
2. En base a esa lista de tareas el ingeniero de usabilidad crea una especificación de ellas en formato FLUD simplificado, ahorrándose con ello la especificación de minuciosos detalles en el formato FLUD común.
3. Se obtienen los datos de interacción de los usuario en formato FLUD.
4. Se utiliza la herramienta ShFLUD para llevar a cabo un análisis que consiste en identificar las coincidencias y diferencias existentes entre la especificación del ingeniero de usabilidad y la interacción de cada usuario.
5. ShFLUD genera como salida dos archivos por cada comparación. El primer archivo describe profusamente todas y cada una de las coircidencias y diferencias entre ambas secuencias de eventos. El segundo archivo contiene instrucciones que serán seguidas por un programa adicional para poder mostrar gráficamente el resultado del análisis.

El proceso previamente enumerado evidencia que efectivamente se ha automatizado una parte del proceso de análisis de los datos del usuario. A partir de la información obtenida por la comparación un ingeniero de usabilidad podrá determinar si los usuarios han logrado cumplir total o parcialmente con el modelo de ejecución de la tarea esbozado por él.

### c) Visualización de los resultados del análisis.

La visualización de los resultados de la comparación de secuencias de eventos se lleva a cabo mediante la herramienta Biolayout Java, creada por investigadores del EMBL-European Bioinformatics Institute.

Lo que se puede observar en las gráficas generadas en esta etapa es un grafo que muestra puntos de unión y divergencia entre la ruta de interacción propuesta por el ingeniero de usabilidad, y la ruta seguida por cada usuario.

Con la finalidad de demostrar la aplicabilidad de los conceptos aquí vertidos se llevó a cabo un estudio de usabilidad sobre la interfaz del sitio web de una empresa. Los resultados fueron satisfactorios, y las técnicas se pudieron incorporar al proceso de evaluación sin requerir de grandes esfuerzos.

Se puede decir entonces que la propuesta realizada en esta tesis pueden ser trasladada ya aplicados en cualquier proyecto de ingeniería del software, y que de su uso se puede beneficiar cualquier organización que deseé

## 5.2 Ventajas y problemáticas del enfoque

Las ventajas esenciales sobre enfoques anteriores se pueden resumir en los siguientes puntos:

- a) Análisis aplicable a interfaces web. En investigaciones anteriores ya se ha implementado el análisis comparativo de las secuencias de eventos realizados por un experto y por usuarios de prueba. Sin embargo hasta ahora nunca se había aplicado a interfaces web. Los enfoque se han orientado a interfaces WIMP (*windows, icons, pointer and mouse*; ventanas, íconos, apuntador y ratón).
- b) En las investigaciones en que se ha logrado implementar el análisis comparativo de las secuencias de eventos realizados por un experto y por usuarios de prueba no se ha tenido en cuenta la posibilidad de almacenar los datos de interacción en un formato estandarizado. Cada herramienta ha tenido su propio formato diseñado a la medida. De esa forma se impide entre otras cosas el intercambio de información entre colegas y la proliferación de herramientas de análisis de uso generalizado.
- c) Por un lado las escasas herramientas de automatización de procesos de evaluación de usabilidad en interfaces web hasta ahora se han orientado únicamente a la etapa de recolección de datos. Se arrojan datos que quedan para la libre interpretación del ingeniero de usabilidad. Por otra parte, otras herramientas simplemente se abocan a la confrontación del código fuente de la página web con reglas bien definidas de diseño y accesibilidad, en un entorno estático. No brindan la posibilidad de aplicar el mismo concepto a sesiones reales con usuarios.

Como desventajas se pueden mencionar las siguientes:

- a) En el aspecto puramente práctico, se espera tener una implementación más adecuada de la herramienta WebVIP. Aún se deben implementar breves secciones de la especificación original y corregir errores de programación menores.
- b) El formato FLUD aún no ha cobrado el auge que se esperaría entre la comunidad investigadora de interacción humano – computadora. Se pone así en entredicho la ventaja de intercambio de información de pruebas entre colegas.

### **5.3 Futuras direcciones de investigación**

A continuación se mencionan algunas vertientes que podrían ser de interés para investigaciones ulteriores:

- Aplicación a Interfaces web creadas dinámicamente. Cada vez es más común encontrar sitios en que las páginas que lo componen no se corresponden directamente con archivos en el servidor. El contenido de tales páginas es generado bajo demanda, creando una página distinta para cada petición. Una primera ampliación de la investigación podría orientarse a una aplicación directa de los conceptos a esa clase de sitios.
- Interfaces adaptables. Un proyecto aún más ambicioso podría ser el de olvidarse por un momento del contexto bajo el que se desarrolló esta tesis: la evaluación de interfaces de usuario. Se podría tener dentro de un servidor web un programa motor de inferencia. Dicho motor sería alimentado por especificaciones de tareas en formato FLUD Simplificado. Cada vez que recibiera una petición compararía la secuencia de eventos generados hasta ahora por el usuario actual con la especificación que guarda. De esa manera podría determinar la tarea que está realizando en esos momentos el usuario. Inclusive se podrían agregar funciones predictivas al servidor. Si se reconociera una secuencia de pasos como el inicio de una tarea en particular, se sabrían con antelación las intenciones del usuario. Se podría entonces tomar la acción más adecuada en ese momento: brindarle ayuda, proponerle algún atajo, etc.
- Construcción de Herramientas. Se podría explotar mucho más la simplicidad y capacidad expresiva que tiene el formato FLUD Simplificado, generando nuevas herramientas basadas en él. Por ejemplo, se podría construir un programa que tomara la especificación en formato FLUD de una tarea como instrucciones para llevar a cabo una interacción automatizada con un sitio web determinado.

Existen obviamente más enfoques posibles. Aquí solamente se han tomado las que consideramos más importantes e inmediatas.

# Apéndice A

## Especificación del Formato FLUD Simplificado

```
cr = 13;
lf = 10;
tab = 9;
digito = ['0'..'9'];
punto = '.';
nodigito = ['_' + [{'a' .. 'z'} + {'A' .. 'Z'}]];
comillas = 34;
comilla = 39;
espacio = ' '+';
l_par = '(';
r_par = ')';
tecla = [1..255];
menorque = '<';
mayorque = '>';
```

```
deftareaini = 'deftarea=';
deftareafin = 'deftarea.';
endtareaini = 'endtarea=';
endtareafin = 'endtarea.';
```

```
request = 'request';
loading = 'loading';
complete = 'complete';
label = 'label';
```

```
newpage = 'NEWPAGE';
pointerpress = 'POINTERPRESS';
keypress = 'KEYPRESS';
window_op = 'WINDOW_OP';
```

```
button = 'button';
radio = 'radio';
checkbox = 'checkbox';
textbox = 'textbox';
menu = 'menu';
slider = 'slider';
scrollbarh = 'scrollbarh';
scrollbarv = 'scrollbarv';
handle = 'handle';
link = 'link';
unknown = 'unknown';
```

```
open = 'open';
close = 'close';
icon = 'icon';
de_icon = 'de_icon';
move = 'move';
resize = 'resize';
raise = 'raise';
lower = 'lower';
focus = 'focus';
blur = 'blur';
```



```

comodin = '*';
comillas = '"';

new_line = cr | lf | cr lf;
tab = tab;
espacio = ' '+';

cadenacomillas = comillas
(nodigito | digito | '/' | '?' | '?' | '?' | '[' |
']' | '\' | punto | ';' | '@' | ':' | '#' | '~' | '_' |
'(' | ')') | comilla | espacio)*
comillas ;

widget_id = menorque
(nodigito | digito | '/' | '?' | '?' | '?' | '[' | ']' |
'\ ' | punto | ';' | '@' | ':' | '#' | '~' | '_' | '(' | ')') |
comilla | espacio)*
mayorque;

widget_value = comilla
(nodigito | digito | '/' | '?' | '?' | '?' | '[' | ']' |
'\ ' | punto | ';' | '@' | ':' | '#' | '~' | '_' | '(' |
')') | comilla | espacio)*
comilla ;

tecla = tecla ;
id_tecla = '$' tecla '$' ;

log = shorthevent * ;

shorthevent =
{newpage} evento_newpage |
{pointerpress} evento_pointerpress |
{keypress} evento_keypress |
{window_op} evento_window_op |
{comodin} comodin ;

evento_newpage =
{request} newpage request url |
{loading} newpage loading url |
{complete} newpage complete url |
{label} newpage label url ;

evento_pointerpress =
{button}pointerpress button widget_id widget_value url |
{radio} pointerpress radio widget_id widget_value url |
{checkbox} pointerpress checkbox widget_id widget_value url |
{textbox} pointerpress textboxwidget_id widget_value url |
{menu} pointerpress menu widget_id widget_value url |
{slider}pointerpress slider widget_id widget_value url |
{scrollbarh} pointerpress scrollbarh widget_id widget_value url |
{scrollbarv} pointerpress scrollbarv widget_id widget_value url |
{handle}pointerpress handle widget_id widget_value url |
{link} pointerpress link widget_id widget_value url |
{unknown} pointerpress unknownwidget_id widget_value url ;

```

```

evento_keypress =
    {button}keypress id_tecla button widget_id widget_value url |
    {radio}keypress id_tecla radio widget_id widget_value url |
    {checkbox} keypress id_tecla checkbox widget_id widget_value url |
    {textbox} keypress id_tecla textboxwidget_id widget_value url |
    {menu} keypress id_tecla menu widget_id widget_value url |
    {slider}keypress id_tecla slider widget_id widget_value url |
    {scrollbarh} keypress id_tecla scrollbarh widget_id widget_value url |
    {scrollbarv} keypress id_tecla scrollbarv widget_id widget_value url |
    {handle}keypress id_tecla handle widget_id widget_value url |
    {link} keypress id_tecla link widget_id widget_value url ;

evento_window_op =
    {open} window_op open url |
    {close} window_op close url |
    {icon} window_op icon url |
    {de_icon} window_op de_icon url |
    {move} window_op move url |
    {resize} window_op resize url |
    {raise} window_op raise url |
    {lower} window_op lower url |
    {focus} window_op focus url |
    {blur} window_op blur url ;

url = cadenacomillas;

```

## Apéndice B

### Questionario de perfil de usuario

Prueba: \_\_\_\_\_

Fecha: \_\_\_\_\_

Nota: \_\_\_\_\_

---

### Questionario de perfil de usuario

1. ¿Cuál es su nivel académico?

\_\_\_\_\_

2. ¿Desde cuándo usa computadoras?

\_\_\_\_\_

3. ¿Con qué frecuencia las utiliza?

\_\_\_\_\_

4. ¿Hace cuánto utiliza Internet?

\_\_\_\_\_

5. ¿Con qué frecuencia?

\_\_\_\_\_

6. ¿Para qué lo utiliza?

\_\_\_\_\_

7. ¿Qué entiende por Internet?

\_\_\_\_\_

8. ¿Está familiarizado con capturas de información?

---

9. ¿Ha capturado información a través de Internet?

---

# Apéndice C

## Protocolo de bienvenida

Prueba: \_\_\_\_\_

Fecha: \_\_\_\_\_

Nota:

---

### **Protocolo de bienvenida para prueba de usabilidad del "Sitio Web del Grupo ideSolex S.A. de C.V."**

Buenos días, mi nombre es <Nombre del monitor> y estaré con usted en esta sesión. Permítame explicarle por qué está usted aquí.

Estaremos probando el sitio web de la empresa Grupo ideSolex S.A. de C.V.

Mediante el estudio de sus acciones trataremos de determinar los defectos o virtudes de este sitio en términos de la facilidad con que usted interactúe con el mismo. Es importante enfatizar que el que será evaluado es el prototipo y no usted, por ello le pedimos que actúe con naturalidad, pero sobre todo, que responda con honestidad.

Se le presentará una ventana con el sitio web y se le pedirá que realice algunas tareas típicas para las cuales está diseñado el sistema, la sesión consistirá en que usted las efectúe y describa en voz alta sus acciones así como cualquier opinión que tenga ya que esto será de mucha ayuda para nosotros.

Una vez comenzada la sesión siéntase en total libertad de hacer cualquier pregunta, aunque no podré contestar algunas de ellas, ya que el objetivo es simular la situación real en la cual operará el sistema de manera autónoma.

¿Tiene alguna duda?

## Apéndice D. Cuestionario de usabilidad

Prueba: \_\_\_\_\_

Fecha: \_\_\_\_\_

Nota: \_\_\_\_\_

### Cuestionario de usabilidad

**Por favor lea cuidadosamente las aseveraciones y marque con una X qué tan de acuerdo o qué tan en desacuerdo está con cada una de ellas. Si no está seguro(a) de qué contestar, marque 3. Por favor registre su respuesta inmediata a cada una.**

- |   | Fuertemente<br>en desacuerdo |                          |                          |                          | Fuertemente<br>de acuerdo |
|---|------------------------------|--------------------------|--------------------------|--------------------------|---------------------------|
| 1. Creo que me gustará usar el sistema frecuentemente                     | <input type="checkbox"/>     | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/>  |
|   | 1                            | 2                        | 3                        | 4                        | 5                         |
| 2. Encuentro el sistema innecesariamente complejo                         | <input type="checkbox"/>     | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/>  |
|   | 1                            | 2                        | 3                        | 4                        | 5                         |
| 3. Pienso que el sistema era fácil de usar                                | <input type="checkbox"/>     | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/>  |
|   | 1                            | 2                        | 3                        | 4                        | 5                         |
| 4. Creo que necesitaré la ayuda de un técnico para poder usar el sistema  | <input type="checkbox"/>     | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/>  |
|   | 1                            | 2                        | 3                        | 4                        | 5                         |
| 5. Encontré que las distintas funciones del sistema están bien integradas | <input type="checkbox"/>     | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/>  |
|   | 1                            | 2                        | 3                        | 4                        | 5                         |
| 6. Pienso que había mucha inconsistencia en el sistema                    | <input type="checkbox"/>     | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/>  |
|   | 1                            | 2                        | 3                        | 4                        | 5                         |
| 7. Me imagino que la gente aprenderá a usar el sistema bastante rápido    | <input type="checkbox"/>     | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/>  |
|   | 1                            | 2                        | 3                        | 4                        | 5                         |
| 8. Encuentro al sistema engorroso de usar                                 | <input type="checkbox"/>     | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/>  |
|   | 1                            | 2                        | 3                        | 4                        | 5                         |
| 9. Me sentí muy seguro usando el sistema                                  | <input type="checkbox"/>     | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/>  |
|   | 1                            | 2                        | 3                        | 4                        | 5                         |
| 10. Necesito aprender muchas cosas antes e poder utilizar el sistema      | <input type="checkbox"/>     | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/>  |
|   | 1                            | 2                        | 3                        | 4                        | 5                         |

## Apéndice E

### Especificación de las tareas del usuario

Prueba: \_\_\_\_\_

Fecha: \_\_\_\_\_

Nota: \_\_\_\_\_

---

#### Tareas de usuario

**Ir tachando las actividades realizadas con el fin de no perderse durante la prueba. NO se realizará ninguna actividad si antes no se ha concluido la anterior (a menos que se instruya lo contrario).**

1.  Abra la pagina principal del Sitio
2.  Lea la historia de la empresa "Grupo iSolex"
3.  Encuentre objetivo, misión y visión del grupo iSolex.
4.  Ingrese a la página principal de Tecnologías del sitio
5.  Describa las características de PPH
6.  Ingrese al Tutorial de CSS
7.  Ingrese al capítulo llamado "Cookies" del tutorial de ASP
8.  Escriba y envíe su opinión acerca del sitio
9.  Encuentre el domicilio postal de la empresa "Grupo iSolex"

## Apéndice F

### Descripción de los instrumentos

Prueba: \_\_\_\_\_

Fecha: \_\_\_\_\_

Nota: \_\_\_\_\_

### Instrumentos

<b>Actividad del Usuario</b>	<b>Tipo de Resultados</b>
1	Que el usuario sepa acceder al sistema e identificar cuando se encuentra dentro
2	El usuario encontrará la página de la historia de la empresa en no más de 30 segundos
3	El usuario encontrará objetivo, misión y visión de la empresa en no más de 30 segundos
4	El usuario ingresará a la página principal de Tecnologías del sitio en no más de 30 segundos
5	El usuario encontrará la página del producto PPH en no más de 30 segundos
6	El usuario encontrará el tutorial de CSS en no más de 30 segundos
7	El usuario ingresará al capítulo "Cookies" del tutorial de ASP en no más de 30 segundos
8	El usuario escribirá y enviará sus opiniones acerca del sitio en no más de 3 minutos
9	El usuario encontrará el domicilio postal en no más de 30 segundos



## Apéndice G

### Secuencias de eventos en formato FLUD Simplificado generadas durante las pruebas

```
NEWPAGE loading "http://localhost/WebLogs/prueba/website/index.html"  
NEWPAGE loading "http://localhost/WebLogs/prueba/website/historia.htm"  
NEWPAGE loading "http://localhost/WebLogs/prueba/website/filosofia.htm"  
NEWPAGE loading "http://localhost/WebLogs/prueba/website/tecnologias/"  
NEWPAGE loading "http://localhost/WebLogs/prueba/website/productos/pph.htm"  
NEWPAGE loading "http://localhost/WebLogs/prueba/website/tut/css/"  
WINDOW_OP close "http://localhost/WebLogs/prueba/website/tut/asp/cookies.htm"NEWPAGE loading  
"http://localhost/WebLogs/prueba/website/contacto/"
```

#### Listado G.1 Especificación del Diseñador

```
NEWPAGE loading "http://localhost/WebLogs/prueba/website/index.html"  
NEWPAGE request "http://localhost/WebLogs/prueba/website/index.htm"  
WINDOW_OP close "http://localhost/WebLogs/prueba/website/index.html"  
NEWPAGE loading "http://localhost/WebLogs/prueba/website/index.htm"  
WINDOW_OP close "http://localhost/WebLogs/prueba/website/index.htm"  
NEWPAGE loading "http://localhost/WebLogs/prueba/website/tecnologias/"  
WINDOW_OP close "http://localhost/WebLogs/prueba/website/tecnologias/"  
NEWPAGE loading "http://localhost/WebLogs/prueba/website/tut/"  
NEWPAGE request "http://localhost/WebLogs/prueba/website/tut/css/"  
WINDOW_OP close "http://localhost/WebLogs/prueba/website/tut/"  
NEWPAGE loading "http://localhost/WebLogs/prueba/website/tut/css/"  
WINDOW_OP close "http://localhost/WebLogs/prueba/website/tut/css/"  
NEWPAGE loading "http://localhost/WebLogs/prueba/website/tut/asp/"  
NEWPAGE request "http://localhost/WebLogs/prueba/website/tut/asp/cookies.htm"  
WINDOW_OP close "http://localhost/WebLogs/prueba/website/tut/asp/"  
NEWPAGE loading "http://localhost/WebLogs/prueba/website/tut/asp/cookies.htm"  
WINDOW_OP close "http://localhost/WebLogs/prueba/website/tut/asp/cookies.htm"  
NEWPAGE loading "http://localhost/WebLogs/prueba/website/contacto/"  
WINDOW_OP close "http://localhost/WebLogs/prueba/website/contacto/" NEWPAGE loading  
"http://localhost/WebLogs/prueba/website/contacto/"
```

#### Listado G.2 Secuencia de Eventos en Formato FLUD Simplificado, usuario 1

```
NEWPAGE loading "http://localhost/WebLogs/prueba/website/index.html"  
NEWPAGE request "http://localhost/WebLogs/prueba/website/index.htm"  
WINDOW_OP close "http://localhost/WebLogs/prueba/website/index.html"  
NEWPAGE loading "http://localhost/WebLogs/prueba/website/index.htm"  
WINDOW_OP close "http://localhost/WebLogs/prueba/website/index.htm"  
NEWPAGE loading "http://localhost/WebLogs/prueba/website/historia.htm"  
WINDOW_OP close "http://localhost/WebLogs/prueba/website/historia.htm"  
NEWPAGE loading "http://localhost/WebLogs/prueba/website/filosofia.htm"  
WINDOW_OP close "http://localhost/WebLogs/prueba/website/filosofia.htm"  
NEWPAGE loading "http://localhost/WebLogs/prueba/website/tecnologias/"  
WINDOW_OP close "http://localhost/WebLogs/prueba/website/tecnologias/"  
NEWPAGE loading "http://localhost/WebLogs/prueba/website/productos/pph.htm"  
WINDOW_OP close "http://localhost/WebLogs/prueba/website/productos/pph.htm"  
NEWPAGE loading "http://localhost/WebLogs/prueba/website/tut/css/"  
WINDOW_OP close "http://localhost/WebLogs/prueba/website/tut/css/"  
NEWPAGE loading "http://localhost/WebLogs/prueba/website/tut/asp/"  
NEWPAGE request "http://localhost/WebLogs/prueba/website/tut/asp/cookies.htm"  
WINDOW_OP close "http://localhost/WebLogs/prueba/website/tut/asp/"  
NEWPAGE loading "http://localhost/WebLogs/prueba/website/tut/asp/cookies.htm"  
WINDOW_OP close "http://localhost/WebLogs/prueba/website/tut/asp/cookies.htm"  
NEWPAGE loading "http://localhost/WebLogs/prueba/website/contacto/"  
WINDOW_OP close "http://localhost/WebLogs/prueba/website/contacto/" NEWPAGE loading  
"http://localhost/WebLogs/prueba/website/contacto/"
```

#### Listado G.3 Secuencia de Eventos en Formato FLUD Simplificado, usuario 2

```

NEWPAGE loading "http://localhost/WebLogs/prueba/website/index.html"
NEWPAGE request "http://localhost/WebLogs/prueba/website/index.htm"
WINDOW_OP close "http://localhost/WebLogs/prueba/website/index.html"
NEWPAGE loading "http://localhost/WebLogs/prueba/website/index.htm"
WINDOW_OP close "http://localhost/WebLogs/prueba/website/index.htm"
NEWPAGE loading "http://localhost/WebLogs/prueba/website/tecnologias/"
WINDOW_OP close "http://localhost/WebLogs/prueba/website/tecnologias/"
NEWPAGE loading "http://localhost/WebLogs/prueba/website/index.htm"
WINDOW_OP close "http://localhost/WebLogs/prueba/website/index.htm"
NEWPAGE loading "http://localhost/WebLogs/prueba/website/productos/pph.htm"
WINDOW_OP close "http://localhost/WebLogs/prueba/website/productos/pph.htm"
NEWPAGE loading "http://localhost/WebLogs/prueba/website/tut/css/"
WINDOW_OP close "http://localhost/WebLogs/prueba/website/tut/css/"
NEWPAGE loading "http://localhost/WebLogs/prueba/website/tut/asp/"
NEWPAGE request "http://localhost/WebLogs/prueba/website/tut/asp/cookies.htm"
WINDOW_OP close "http://localhost/WebLogs/prueba/website/tut/asp/"
NEWPAGE loading "http://localhost/WebLogs/prueba/website/tut/asp/cookies.htm"
NEWPAGE request "http://localhost/WebLogs/prueba/website/tut/asp/enviaremail.htm"
WINDOW_OP close "http://localhost/WebLogs/prueba/website/tut/asp/cookies.htm"
NEWPAGE loading "http://localhost/WebLogs/prueba/website/tut/asp/enviaremail.htm"
WINDOW_OP close "http://localhost/WebLogs/prueba/website/tut/asp/enviaremail.htm"
NEWPAGE loading "http://localhost/WebLogs/prueba/website/contacto/"
WINDOW_OP close "http://localhost/WebLogs/prueba/website/contacto/" NEWPAGE loading
"http://localhost/WebLogs/prueba/website/contacto/"

```

**Listado G.4 Secuencia de Eventos en Formato FLUD Simplificado, usuario 3**

```

NEWPAGE loading "http://localhost/WebLogs/prueba/website/index.html"
NEWPAGE request "http://localhost/WebLogs/prueba/website/index.htm"
WINDOW_OP close "http://localhost/WebLogs/prueba/website/index.html"
NEWPAGE loading "http://localhost/WebLogs/prueba/website/index.htm"
WINDOW_OP close "http://localhost/WebLogs/prueba/website/index.htm"
NEWPAGE loading "http://localhost/WebLogs/prueba/website/historia.htm"
WINDOW_OP close "http://localhost/WebLogs/prueba/website/historia.htm"
NEWPAGE loading "http://localhost/WebLogs/prueba/website/filosofia.htm"
WINDOW_OP close "http://localhost/WebLogs/prueba/website/filosofia.htm"
NEWPAGE loading "http://localhost/WebLogs/prueba/website/tecnologias/"
WINDOW_OP close "http://localhost/WebLogs/prueba/website/tecnologias/"
NEWPAGE loading "http://localhost/WebLogs/prueba/website/productos/pph.htm"
WINDOW_OP close "http://localhost/WebLogs/prueba/website/productos/pph.htm"
NEWPAGE loading "http://localhost/WebLogs/prueba/website/tut/css/"
WINDOW_OP close "http://localhost/WebLogs/prueba/website/tut/css/"
NEWPAGE loading "http://localhost/WebLogs/prueba/website/tut/asp/"
NEWPAGE request "http://localhost/WebLogs/prueba/website/tut/asp/cookies.htm"
WINDOW_OP close "http://localhost/WebLogs/prueba/website/tut/asp/"
NEWPAGE loading "http://localhost/WebLogs/prueba/website/tut/asp/cookies.htm"
WINDOW_OP close "http://localhost/WebLogs/prueba/website/tut/asp/cookies.htm"
NEWPAGE loading "http://localhost/WebLogs/prueba/website/contacto/"
WINDOW_OP close "http://localhost/WebLogs/prueba/website/contacto/" NEWPAGE loading
"http://localhost/WebLogs/prueba/website/contacto/"

```

**Listado G.5 Secuencia de Eventos en Formato FLUD Simplificado, usuario 4**

```
NEWPAGE loading "http://localhost/WebLogs/prueba/website/index.html"
NEWPAGE request "http://localhost/WebLogs/prueba/website/index.htm"
WINDOW_OP close "http://localhost/WebLogs/prueba/website/index.html"
NEWPAGE loading "http://localhost/WebLogs/prueba/website/index.htm"
WINDOW_OP close "http://localhost/WebLogs/prueba/website/index.htm"
NEWPAGE loading "http://localhost/WebLogs/prueba/website/historia.htm"
WINDOW_OP close "http://localhost/WebLogs/prueba/website/historia.htm"
NEWPAGE loading "http://localhost/WebLogs/prueba/website/filosofia.htm"
WINDOW_OP close "http://localhost/WebLogs/prueba/website/filosofia.htm"
NEWPAGE loading "http://localhost/WebLogs/prueba/website/tecnologias/"
WINDOW_OP close "http://localhost/WebLogs/prueba/website/tecnologias/"
NEWPAGE loading "http://localhost/WebLogs/prueba/website/productos/pph.htm"
WINDOW_OP close "http://localhost/WebLogs/prueba/website/productos/pph.htm"
NEWPAGE loading "http://localhost/WebLogs/prueba/website/tut/css/"
WINDOW_OP close "http://localhost/WebLogs/prueba/website/tut/css/"
NEWPAGE loading "http://localhost/WebLogs/prueba/website/tut/asp/"
NEWPAGE request "http://localhost/WebLogs/prueba/website/tut/asp/cookies.htm"
WINDOW_OP close "http://localhost/WebLogs/prueba/website/tut/asp/"
NEWPAGE loading "http://localhost/WebLogs/prueba/website/tut/asp/cookies.htm"
WINDOW_OP close "http://localhost/WebLogs/prueba/website/tut/asp/cookies.htm"
NEWPAGE loading "http://localhost/WebLogs/prueba/website/contacto/"
WINDOW_OP close "http://localhost/WebLogs/prueba/website/contacto/" NEWPAGE loading
"http://localhost/WebLogs/prueba/website/contacto/"
```

Listado G.6 Secuencia de Eventos en Formato FLUD Simplificado, usuario 5

## Referencias

- [1] M. T. P. CARD S.K., NEWELL A., "The Psychology of Human-Computer Interaction", Erlbaum, New Jersey 1983.
- [2] C. f. A. S. T. CAST, Página Web de la Herramienta Bobby, <http://bobby.watchfire.com/bobby/html/en/index.jsp>, 2005
- [3] C. CROWLEY, "TkReplay: Record and Replay in Tk", University of New Mexico.
- [4] J. CUGINI, The FLUD format: Logging Usability Data from Web-based Applications, <http://www.itl.nist.gov/iaui/vvrg/cugini/webmet/flud/specification.html>, 2002
- [5] L. S. CUGINI John, Design of a File Format for Logging Website Interaction, <http://www.itl.nist.gov/iaui/vvrg/cugini/webmet/flud/specification.html/design-paper.html>, 2001
- [6] B. D. DAVIDSON, "Web Traffic Logs: An Imperfect Resource for Evaluation", The State University of New Jersey al Rutgers 1999.
- [7] L. S. DE MARSICO Maria , "Evaluating Web Sites: exploiting user's expectations", University of Rome "La Sapienza" 2003.
- [8] F. J. DIX A., ABOWD G., BEALE R., Human Computer Interaction, 2 ed: Prentice Hall, 1998.
- [9] C. I. ENRIGHT Anton, GOLDOVSKY Leon, "Biolayout java v1.21b",.: Computational Genomics Group, EMBL-European Bioinformatics Institute, 2001.
- [10] C. I. ENRIGHT Anton, GOLDOVSKY Leon, Ejemplos de Uso de la Herramienta Biolayout Java, <http://maine.ebi.ac.uk:8000/services/biolayout/examples>, 2005
- [11] C. J. ETGEN Michael, "What does getting WET (Web Event-logging Tool) Mean for Web Usability?", User Experience Engineering Division, AT&T Labs 1999.
- [12] D. G. J. J. FULLER Rodney, "Measuring User Motivation from Server Log Files", 1996.
- [13] E. M. GAGNON, "Sablecc Java Parser Generator", 2005.
- [14] B. D. S. GILLAN J. Douglas, "Designer's Models of the Human-Computer Interface", presented at SIGCHI, 1990.
- [15] S. P. GUZDIAL Mark, BADRE Albert, HUDSON Scott, GRAY Mark, "Analysing and Visualizing Log Files: A Computational Science of Usability", Georgia Institute of Technology 1994.

- [16] L. J. A. HELFRICH Brian, "QUIP: Quantitative User Interface Profiling", University of California at Berkeley 1999.
- [17] D. M. HILBERT, "A Survey of Computer-Aided Techniques for Extracting Usability Information from User Interface Events", University of California at Irvine 1998.
- [18] R. D. F. HILBERT David M., "Extracting Usability Information from User Interface Events", University of California at Irvine 1999.
- [19] I. o. E. a. E. E. IEEE, "Recommended Practice for Internet Practices - Web Page Engineering - Intranet / Extranet Applications", 1999.
- [20] IMAGIWARE, Página Web de la Herramienta DoctorHTML, <http://www.doctor-html.com/>, 2005
- [21] ISO, "Guidance on usability specification and measures", EUA, 1992.
- [22] G. ISolex, Página Web de Grupo iSolex, <http://www.solex.com.mx/>, 2005
- [23] H. M. A. IVORY Melody Y., "The State of the Art in Automating Usability Evaluation of User Interfaces", University of California, Berkeley 2001.
- [24] C. A. IVORY Melody Y., "A Study of Automated Site Evaluation Tools", 2002.
- [25] D. KIERAS, "A Guide to GOMS Model Usability Evaluation using GOMSL and GLEAN3", University of Michigan 1999.
- [26] F. S. LOK Simon, "A Survey of Automated Layout Techniques for Information Presentations", Columbia University 2000.
- [27] D. J. MCROSTIE, "KALDI: A CAUSE Tool for Supporting Testing and Analysis of Human-Computer Interaction", RMIT University, 1998.
- [28] L. M. NEWMAN W., Interactive System Design: Addison Wesley, 1995.
- [29] J. NIELSEN, "Finding Usability Problems Through Heuristic Evaluation", presented at ACM CHI'92 Conference on Human Factors in Computing Systems, 1992.
- [30] J. NIELSEN, Usability Engineering: Academic Press, 1993.
- [31] J. NIELSEN, "Iterative User-Interface Design", IEEE Computer, 1993.
- [32] J. NIELSEN, Designing Web Usability: The Practice of Simplicity: New Riders Publishing, 2000.
- [33] J. NIELSEN, Jakob Nielsen's Alertbox, August 25, 2003: Usability 101: Introduction to Usability, <http://www.useit.com/alertbox/20030825.html>, 2003
- [34] N. I. o. S. a. T. NIST, Página Web de la Suite WebMetrics, <http://www.nist.gov/webmetrics/>, 2005

- [35] N. I. o. S. a. T. NIST, Página Web de Documentación Técnica de la Suite WebMetrics, <http://zing.ncsl.nist.gov/WebTools/tech.html>, 2005
- [36] N. I. o. S. a. T. NIST, Manual de Instalación de WebMetrics, <http://zing.ncsl.nist.gov/WebTools/WebVIP/installation.html>, 2005
- [37] D. R. OLSER, "MIKE: The Menu Interaction Kontrol Environment",.: Brigham Young University, 1997.
- [38] M. PEARROW, Web Site Usability Handbook: Charles Rover Media Inc.. 2000.
- [39] G. PERLMAN, Web-Based User Interface Evaluation with Questionnaires, <http://www.acm.org/~perlman/question.html>, 2001
- [40] REGEXP, Sitio de Información acerca de las Expresiones Regulares, <http://www.regular-expressions.info/>, 2005
- [41] F. M. RIEMAN John , REDMILES David, "Usability Evaluation with the Cognitive Walkthrough", presented at CHI, 1995.
- [42] S. M. SUN, "Sitio Web del Java Development Kit version 1.4.2", 2005.
- [43] S. M. SUN, Sitio Web del Lenguaje de Programación Java, <http://java.sun.com/>, 2005
- [44] S. M. SUN, Página Web del Kit de Desarrollo de Interfaces Gráficas de Usuario Java Swing, <http://java.sun.com/products/jfc/>, 2005
- [45] S. M. SUN, Página Web del Kit de Desarrollo de Interfaces Gráficas de Usuario Java AWT, <http://java.sun.com/products/jdk/awt/>, 2005
- [46] TCL/TK, Página Web del Kit de Desarrollo de Interfaces Gráficas de Usuario Tcl/Tk, <http://www.tcl.tk/>, 2005
- [47] TEC-ED, "Assessing Web Site Usability from Server Log Files", 1999.
- [48] TROLLTECH, Trolltech, Sitio Web del Marco de Trabajo Qt, <http://www.trolltech.com/products/qt/>, 2005
- [49] T. S. TULLIS, "Is User Interface Design Just Common Sense?", presented at Fifth International Conference on Human-Computer Interaction, 1993.
- [50] W. K. UEHLING Dana L., "User Action Graphing Effort (UsAGE)", 1995.
- [51] U. P. A. UPA, Sitio web de la Asociación de Profesionales de Usabilidad, Usability Professionals Association, [http://www.upassoc.org/usability\\_resources/guidelines\\_and\\_methods/standards.html](http://www.upassoc.org/usability_resources/guidelines_and_methods/standards.html), 2005

- [52] USABLENET, Pagina Web de la Herramienta LIFT, <http://www.usablenet.com/>, 2005
- [53] W. W. W. C. W3C, Web Accessibility Initiative, <http://www.w3.org/WAI/>, 2005
- [54] W. W. W. C. W3C, "Web Content Accesibility Guidelines 1.0", 2005.
- [55] WIKIPEDIA, Java programming language, [http://en.wikipedia.org/wiki/Java\\_programming\\_language](http://en.wikipedia.org/wiki/Java_programming_language), 2005
- [56] WXWINDOWS, Sitio Web del Marco de Trabajo wxWindows, <http://www.wxwindows.org/>, 2005
- [57] XWINDOW, Sitio Web del Sistema XWindow, <http://www.x.org/>, 2005