

# Evaluación Heurística

González María Paula  
Pascual Afra  
Lorés Jesús

Universitat de Lleida

# Evaluación Heurística

María Paula González, Jesús Lorés, Afra Pascual

<b>Objetivos</b>	<b>2</b>
<b>1. Introducción</b>	<b>2</b>
<b>2. Definición de Evaluación Heurística</b>	<b>3</b>
<b>3. El concepto de Principio Heurístico</b>	<b>4</b>
<b>4. Pasos de una Evaluación Heurística</b>	<b>26</b>
<b>5. Conclusiones</b>	<b>35</b>
<b>Referencias</b>	<b>35</b>

## Objetivos

Este capítulo tiene como objetivo:

- Introducir a la metodología de la Evaluación Heurística
- Presentar el concepto de Principio Heurístico desde una amplia base bibliográfica
- Describir los pasos de una Evaluación Heurística
- Brindar un ejemplo práctico de una Evaluación Heurística

## 1. Introducción

Este Capítulo presenta un método de evaluación de la usabilidad por inspección denominado *Evaluación Heurística*. La Evaluación Heurística consiste en verificar la calidad de una serie de principios llamados Principios Heurísticos o simplemente "heurísticas" previamente establecidos. Es llevada a cabo por evaluadores expertos en usabilidad que actúan imitando las reacciones que tendría un usuario promedio al interactuar con el sistema que se está evaluando. A continuación, la Sección 2 presenta a la Evaluación Heurística definiéndose el concepto de heurística y describiéndose las principales características, el objetivo y la potencialidad de esta metodología. Se presentan las principales ventajas y desventajas de esta metodología y se muestran contradicciones encontradas en diversos autores en cuanto a su eficiencia y su efectividad. Seguidamente, la Sección 3 detalla el concepto de Principio Heurístico y presenta distintas corrientes metodológicas que plantean fundamentos teóricos alternativos y complementarios para su definición. A continuación, la Sección 4 describe los diferentes pasos incluidos en una Evaluación Heurística: su Planificación, su Puesta en Marcha y por último, el Análisis de los resultados obtenidos tras la inspección.

En relación a la Planificación de una Evaluación Heurística, en la Sección 4.1 se discute el tipo de perfil y la cantidad de evaluadores óptimos según distintas corrientes bibliográficas estudiadas. También se describe el proceso de traducción que permite transformar a los Principios Heurísticos generales presentados en la Sección 2 en preguntas heurísticas adecuadas para la evaluación de la interfaz de un sistema interactivo particular. Se plantea la importancia de establecer también la escala de valores que se utilizará para la posterior ponderación de cada pregunta heurística considerando a esta selección desde el punto de vista de la llamada Usabilidad Transcultural. Al final de la Sección 4.1 se detalla que tipo de

documentación debería obtenerse al concluir esta etapa de la Evaluación Heurística enfatizándose las ventajas de contar con plataformas de software que actúen como gestores de todo el proceso posterior. A modo de ejemplo, el software UsabAIPO-GestorHeurística es presentado y descrito. Con respecto a la Puesta en Marcha de una Evaluación Heurística, la Sección 4.2 analiza los pasos sugeridos por los creadores de la metodología. Se enfatiza la importancia de minimizar la influencia de factores externos en el proceso de inspección y se detalla que tipo de actividades serán seguidas por cada uno de los evaluadores involucrados. En cuanto al Análisis de resultados de la Evaluación Heurística, la Sección 4.3 describe el procesamiento cuantitativo y cualitativo al que pueden someterse los datos obtenidos. En este sentido se entiende a ambos tipos de procesamiento como complementarios y no excluyentes. Finalmente, la Sección 5 presenta las conclusiones de este Capítulo.

## 2. Definición de Evaluación Heurística

La palabra *heurística* procede etimológicamente de la palabra griega "euriskein" que procede de "eureka", un vocablo que significa hallar o encontrar. Este vocablo fue exclamado por Arquímedes en un famoso episodio sin bases históricas. El diccionario de la Real Academia Española define a la palabra "heurística" como:

- Técnica de la indagación y del descubrimiento.
- Busca o investigación de documentos o fuentes históricas.
- En algunas ciencias, manera de buscar la solución de un problema mediante métodos no rigurosos, como por tanteo, reglas empíricas, etc.

Partiendo de la definición anterior, la Interacción Persona Ordenador (IPO) presenta a la Evaluación Heurística (EH) como un método de evaluación de la usabilidad por inspección que debe ser llevado a cabo por evaluadores expertos a partir de unos principios (denominados "heurísticos") previamente establecidos. Por ser una técnica de evaluación de la usabilidad, la EH tiene como objetivo el medir la calidad de la interfaz de cualquier sistema interactivo en relación a su facilidad para ser aprendido y usado por un determinado grupo de usuarios en un determinado contexto de uso [ISO98, UNET06].

La metodología de la EH fue presentada inicialmente por Nielsen y Molich [MOL90]. Esta metodología consiste en analizar la conformidad de la interfaz con unos principios reconocidos de usabilidad (la "heurística") mediante la inspección de varios evaluadores. Los autores [MOL90] solo plantearon que los evaluadores deberían poseer algún conocimiento sobre los principios de la usabilidad. Más tarde, autores como [JEF91] postularon que la EH sería más efectiva cuando los evaluadores fueran expertos en usabilidad. Cabe destacar que estos evaluadores intentan ponderar a la interfaz que esta siendo evaluada en función de lo que creen que será la percepción de usuarios de distintos perfiles. Es por ello que frente a otros métodos que enfatizan la realización de tareas (como es el caso del método del Recorrido Cognitivo [WRLP94, GPL04]), la EH inspecciona problemas potenciales, ya que el evaluador predice los errores que el usuario real podrá tener cuando interaccione con la interfaz del sistema que se esta evaluando.

Una característica fundamental de la EH es su bajo coste, aunque el mismo puede variar dependiendo del número de evaluadores que se incluya en el proceso. Varios

autores sostienen que la EH es barata, intuitiva y que esto motiva a la gente a utilizarla [MOL90, MUL98, LAI04]. De hecho, la EH no requiere una larga planificación para realizarse y puede usarse en las etapas iniciales del proceso de desarrollo de sistemas, siempre que se disponga de un prototipo a evaluar. Según [NIE94], el método de la EH detecta aproximadamente un 42 % de los problemas graves de diseño y un 32 % de los problemas menores, dependiendo del número de evaluadores que revisen el sitio. Sin embargo, según nuestra opinión, este tipo de afirmaciones no pueden sostenerse en la actualidad, ya que la Ingeniería de la Usabilidad ha avanzado considerablemente en la última década, por lo que el contexto actual es mucho más favorable que el que podía observarse en el año 1994 (más conocimiento sobre la metodología de la EH, más madurez y experiencia en la evaluación de la usabilidad, nuevos modelos de desarrollo de sistemas propios de la Ingeniería de la Usabilidad, evaluadores más capacitados, más fuentes bibliográficas, mayor madurez científica y metodológica en la disciplina de la IPO).

En realidad, la potencialidad de la EH para detectar problemas de usabilidad en un determinado sistema interactivo depende en gran medida de las heurísticas que se hayan definido para efectuar la inspección de la interfaz de dicho sistema interactivo [DR99]. A continuación se presenta en más detalle el concepto de heurística detallándose brevemente distintas corrientes metodológicas que plantean fundamentos teóricos alternativos y complementarios para su definición.

## **2.1. Ventajas y Desventajas de la Evaluación Heurística**

La EH ha sido presentada por J. Nielsen como un método de la denominada Ingeniería de la Usabilidad de Descuento (Discount Usability Engineering) en [Nie99]. En este trabajo Nielsen defiende que la EH es menos costosa que otros métodos para la evaluación de la usabilidad ya que requiere, relativamente, pocos recursos y disminuye el costo asociado a la producción de software. En general, las principales ventajas y desventajas de la EH son las siguientes (información basada en material encontrado en la web "Usability Body of Knowledge", consultada en julio del 2006 en <http://www.usabilitybok.org/methods/p275>):

### **Ventajas de la Evaluación Heurística:**

- Es económica en comparación con otros métodos de evaluación de la usabilidad [NIE90]
- Es intuitiva y es fácil motivar a los evaluadores potenciales a que la utilicen [NIE90]
- No requiere planificación por adelantado [NIE90].
- Los evaluadores no necesitan ser expertos en usabilidad. Por ejemplo, en [NIE90] y [NIE92] se presentan EH realizadas por profesionales de las Ciencias de la Computación como programadores y estudiantes de Informática.
- Puede ser utilizada en etapas tempranas del proceso de desarrollo de sistemas, siempre que se cuente con un prototipo a evaluar [NIE90].
- Tiempo de procesamiento menor que los teste de laboratorio [KR97]. Cabe destacar que esta última ventaja puede disminuir su importancia

debido a los avances y mejoras en las metodologías de evaluación de la usabilidad con testeos de laboratorio que se han producido desde el año 1997 a la fecha.

### **Desventajas de la Evaluación Heurística:**

- Según la descripción original de la metodología propuesta en [NIE90], los evaluadores deben conocer los principios de la usabilidad. Pero más tarde, J. Nielsen probó que los evaluadores expertos en usabilidad detectaba mayor cantidad de problemas de usabilidad que los no expertos, y que los evaluadores "doblemente expertos" (expertos en usabilidad que además poseen experticia en el contexto de uso al cual pertenece el sistema que se desea evaluar) detectan aún mayor cantidad de problemas de usabilidad [NIE92]. Esto supone una desventaja por la alta capacitación requerida para los evaluadores si se desea optimizar la detección de problemas de usabilidad con la metodología de la EH.
- Es recomendable que la EH sea realizada por más de un evaluador porque una sola persona no detectará la totalidad de los problemas de usabilidad del sistema que se está evaluando [NIE90]
- La EH puede no identificar tantos atributos relacionados con la usabilidad como otras metodologías, por ejemplo las metodologías por testeo [NIE89]. Sin embargo, en nuestra opinión esta desventaja también está presente en otras metodologías y puede minimizarse si se estructura a los pasos de la EH tal como se muestra en la Sección 4.
- La EH puede identificar mayor cantidad de problemas de usabilidad menores y menor cantidad de problemas de usabilidad mayores que la metodología del Thinking-Aloud [JD92].
- La EH puede resultar difícil de ejecutar si la interfaz que se evalúa es muy compleja. En este tipo de interfaces un grupo pequeño de evaluadores no puede detectar la mayoría de los problemas de usabilidad de alta importancia [SC99]
- La EH no siempre sugiere de manera fácil o clara soluciones para las características que son identificadas. Además, debe tenerse en cuenta el sesgo asociado a la subjetividad de los diferentes evaluadores [NIE90]
- En una EH tradicional los evaluadores solo emulan el comportamiento de los usuarios son usuarios reales [KR97]
- La EH puede tender a reportar falsas alarmas (problemas que son detectados en la aplicación pero que no corresponden en realidad a problemas de usabilidad) [JEF94]

A pesar de que las principales ventajas y desventajas de la EH han sido ampliamente discutidas por autores, se observa una cierta falta de consenso en varios de los trabajos consultados. A continuación presentaremos cuatro ejemplos que se muestran en la web "Usability Body of Knowledge" (consultada en julio del 2006 en <http://www.usabilitybok.org/methods/p275>). Un primer ejemplo que ilustra las contradicciones relacionadas con las ventajas y desventajas de la EH se presenta en [JMWU91]. En este trabajo los autores comparan la efectividad de diferentes métodos de evaluación incluyendo a la EH. Primeramente, estos autores postulan que la EH identifica más problemas de usabilidad que otras metodologías incluidas en su estudio (como los Test de Usabilidad, las Guías de Revisión o los Recorridos Cognitivos) si los resultados de cada uno de los evaluadores (a los cuales considera solo como personas con experiencia en usabilidad en contraposición con la metodología original presentada en [NIE90]) son comparados y compilados en un solo informe de usabilidad (tal como se propone en la Sección

4.3 del presente Capítulo). Sin embargo, más adelante los autores sostienen que aunque la EH detecte más problemas de usabilidad que los Test de Usabilidad, la severidad de los problemas detectados con la EH es menor que la severidad de aquellos detectados con los Test de Usabilidad.

Un segundo ejemplo puede observarse en [DKA92]. En este caso, los autores muestran que los evaluadores expertos identifican más problemas de usabilidad que los evaluadores no expertos, y que los evaluadores "doblemente expertos" [NIE92] aún detectan más problemas de usabilidad que los expertos. Sin embargo, más adelante explican que incluir evaluadores "doblemente expertos" es demasiado costoso por lo que recomiendan la inclusión de evaluadores expertos sin especificar el decremento en la efectividad de la metodología. Un tercer ejemplo más actual puede verse en [CW02]. En este tercer caso, los autores revisan las metodologías de la Ingeniería de la Usabilidad de Descuento desde el punto de vista del costo-beneficio. Por un lado, estos autores señalan que los problemas reales de usabilidad de un determinado sistema se detectan a partir de una compleja interacción entre un usuario y ese sistema. En este sentido, el trabajo de [CW02] sostiene que los métodos de descuento, como la EH, son demasiado simples para evaluar de manera adecuada esa compleja interacción. Contradictoriamente, en el mismo trabajo se concluye que los métodos de descuento son tan buenos para detectar errores de usabilidad que sus beneficios sobrepasan de sobremanera a sus costos.

Finalmente, un cuarto ejemplo que ilustra las contradicciones relacionadas con las ventajas y desventajas de la EH se presenta en [MDJ05]. En este trabajo se comparan diferentes metodologías de evaluación de la usabilidad (entre ellas la EH) aplicadas al sitio web del Hotel Pennsylvania en Nueva York ([www.hotelpenn.com](http://www.hotelpenn.com)) durante el año 2003 por 17 equipos de evaluación de la usabilidad diferentes. Uno de los 17 equipos utilizó la técnica tradicional de la EH tal como se presenta en [NIE90]. Dentro de las conclusiones de [MDJ05] se sostiene que: a) no se produjeron falsas alarmas en relación a los problemas de usabilidad detectados por los evaluadores, conclusión que se contradice con la afirmación dentro del mismo trabajo que advierte sobre las falsas alarmas provocadas por la EH y; b) los evaluadores expertos identificaron la misma proporción de problemas de usabilidad muy severos y poco severos que los Testeo de Usabilidad, conclusión que se contradice con la afirmación anteriormente indicada en el mismo trabajo acerca de la mayor eficiencia de las evaluaciones realizadas por expertos sobre la menor eficiencia de las evaluaciones realizadas por medio de un testeo.

### **3. El concepto de Principio Heurístico**

La Heurística, también conocida como denominada Principio Heurístico (PH) o criterio heurístico, trata de aplicar normas conversacionales a la interacción entre una persona a un sistema: su objetivo es intentar crear un "puente comunicacional" en el que tanto la persona como el sistema se entiendan y trabajen juntos en pos de un objetivo a alcanzar. Estas reglas empíricas generales son utilizadas dentro de la planificación de una HE como punto de partida para la creación de una lista de comprobación de ítems que posteriormente utilizará el evaluador experto dentro de la puesta en marcha de la evaluación. De esta manera, esas reglas generales son adecuadas a cada caso concreto de evaluación para reflejar en los ítems a evaluar la naturaleza y tipo de interfaz a evaluar y el contexto de uso de la misma. Los PHs pueden servir para varios propósitos:

- Guían a diseñadores durante el proceso de diseño.

- Ayudan a los evaluadores a identificar problemas en las interfaces de usuario, comprobando que las reglas de usabilidad se respeten.
- Explican problemas de usabilidad observados,
- Dan pautas sobre porqué los usuarios cometen determinados errores.

Muchos autores han publicado PH para interfaces de usuario. En el año 1986 Schneiderman publicó las "ocho reglas de oro" del diseño de interfaces y posteriormente diversos autores desarrollaron listados similares de PHs basados en problemas observados de usabilidad. A continuación se detalla en orden cronológico una selección de los más relevantes:

### 3.1. Las 8 reglas de oro de Ben Schneiderman

En el año 1986, Ben Schneiderman presenta sus "Ocho reglas de oro para el diseño de Interfaces" (Por más información consultar en <http://www.cs.utexas.edu/users/almstrum/cs370/elvisino/rules.html>). En la actualidad, esas ocho reglas de oro son entendidas como PHs generales que pueden guiar tanto el diseño como la evaluación de la usabilidad de un sistema interactivo. Es evidente que, tal como se explica anteriormente, para llevar a cabo una EH concreta deben traducirse estas ocho reglas de oro a una lista de ítems plausibles de ser evaluados en relación a una determinada interfaz concreta. Las ocho reglas de oro de Schneiderman son:

**Esforzarse por la consistencia:** Las secuencias constantes de acciones se deben repetir en situaciones similares; la misma terminología se debe utilizar en avisos, menús, y pantallas de la ayuda; y los comandos constantes se deben emplear en todas partes del mismo modo.

**Crear atajos para los usuarios frecuentes:** cuando la frecuencia de uso aumenta, los usuarios agradecen reducir el número de interacciones. Las abreviaturas, los comandos ocultos, y las macros son muy útiles para un usuario experto.

**Ofrecer feedback:** Para cada acción del usuario, debe haber una cierta regeneración del sistema. Para las acciones frecuentes y de menor importancia, la respuesta puede ser modesta, mientras que para las acciones infrecuentes e importantes, la respuesta debe ser más substancial.

**Diseñar el diálogo para mostrar trabajo pendiente:** Las secuencias de acciones se deben organizar en grupos con un inicio, un medio y un final. La regeneración informativa en la terminación de un grupo de acciones da a usuarios la satisfacción de la realización, y una indicación clara para prepararse para el grupo siguiente de acciones.

**Ofrecer una gestión sencilla de los errores:** Tanto como sea posible, el sistema ha de estar diseñado para que el usuario no puede causar un error grave. En el caso que ocurra un error, el sistema debe poder detectar el error, y ofrecer mecanismos para poder recuperarse o manejar el error.

**Permitir una fácil recuperación de acciones:** Esta característica ofrece tranquilidad al usuario, puesto que sabe que los errores pueden ser deshechos; anima así la exploración de opciones desconocedoras. La recuperación puede ser una sola acción, una entrada de datos, o un grupo completo de acciones.

**Soportar el control por el usuario:** Los usuarios experimentados desean tener el control total del sistema y que el sistema responda a sus acciones. El diseño del sistema debe responder a las acciones del usuario, y que estos

sean los iniciadores de las acciones, no solo los que respondan a acciones del sistema.

**Reducir la carga de memoria reciente en el usuario:** La limitación humana del tratamiento de la información en memoria a corto plazo requiere que lo que se muestra por pantalla sea simple.

## 3.2. Principios heurísticos de Molich y Nielsen

En el año 1990 R. Molich y J. Nielsen acuñaron una serie de PHs a los que denominaron "heurísticas" [NIE90b, MOL90]. Cuatro años más tarde, J. Nielsen presenta un conjunto minimal de heurísticas de máximo poder expresivo, el cual se basa en los trabajos originales realizados con R. Molich y en la observación de de 249 problemas de usabilidad pertenecientes a proyectos de Diseño Centrado en el Usuario que J. Nielsen presenta en [NIE94a]. Ese conjunto minimal de PHs generales fueron publicados en [NIE94] y sus elementos se presentan a continuación:

1. **Visibilidad del Estado del Sistema.** El sistema debe siempre mantener a los usuarios informados del estado del sistema, con una realimentación apropiada y en un tiempo razonable.
2. **Lenguaje de los Usuarios.** El sistema debe hablar el lenguaje de los usuarios, con las palabras, las frases y los conceptos familiares, en lugar de que los términos estén orientados al sistema. Utilizar convenciones del mundo real, haciendo que la información aparezca en un orden natural y lógico.
3. **Control y libertad para el Usuario.** Los usuarios eligen a veces funciones del sistema por error y necesitan a menudo una salida de emergencia claramente marcada, esto es, salir del estado indeseado sin tener que pasar por un diálogo extendido. Es importante disponer de deshacer y rehacer.
4. **Consistencia y Estándares.** Los usuarios no deben tener que preguntarse si las diversas palabras, situaciones, o acciones significan la misma cosa. En general siga las normas y convenciones de la plataforma sobre la que se está implementando el sistema.
5. **Ayuda a los Usuarios para Reconocimiento, Diagnóstico y Recuperación de errores.** Que los mensajes de error se deben expresar en un lenguaje claro (no haya códigos extraños), se debe indicar exactamente el problema, y deben ser constructivos.
6. **Prevención de Errores.** Es importante prevenir la aparición de errores que mejor que generar buenos mensajes de error.
7. **Reconocimiento antes que Cancelación.** El usuario no debería tener que recordar la información de una parte de diálogo a la otra. Es mejor mantener objetos, acciones, y las opciones visibles que memorizar.
8. **Flexibilidad y eficiencia de uso.** Las instrucciones para el uso del sistema deben ser visibles o fácilmente accesibles siempre que se necesiten. Los aceleradores no vistos por el usuario principiante, mejoran la interacción para el usuario experto de tal manera que el sistema puede servir para usuarios inexpertos y experimentados. Es importante que el sistema permita personalizar acciones frecuentes.
9. **Estética de diálogos y Diseño minimalista.** No deben contener la información que sea inaplicable o se necesite raramente. Cada unidad

adicional de la información en un diálogo compite con las unidades relevantes de la información y disminuye su visibilidad relativa.

10. **Ayuda General y Documentación.** Aunque es mejor si el sistema se pueda usar sin documentación, puede ser necesario disponer de ayuda y documentación. Ésta ha de ser fácil de buscar, centrada en las tareas del usuario, tener información de las etapas a realizar y que no sea muy extensa.

### 3.2.1. Aplicación de los Principios heurísticos de Molich y Nielsen: la lista de comprobación de ítems de D. Pierotti

El conjunto de PHs presentado en la lista anterior ha sido el punto de partida para la creación de numerosas listas de comprobación de ítems (sub-heurísticas o sub-PHs) que luego de ser definidas durante la Etapa de Planificación de una EH particular (ver Etapa de Planificación de una EH en Sección 4.1) son utilizadas por los evaluadores expertos dentro de la Puesta en Marcha de la evaluación (ver Etapa de Puesta en Marcha de una EH en Sección 4.2). A modo de ejemplo, presentamos a continuación una lista de ítems desarrollada por Deniese Pierotti y utilizada por la empresa Xerox Corporation para evaluar la usabilidad de interfaces norteamericanas orientadas a la web (ver ítems 2.20 y 5.14) mediante la metodología de la EH [Pie04]. Cabe mencionar que esta lista agrega a los 10 principios de Nielsen tres PHs más:

11. **Habilidades:** el sistema debería tener en cuenta, extender, suplementar e incentivar las habilidades del usuario, sus conocimientos y su experticia.
12. **Interacción con el Usuario Placentera y Respetuosa:** las interacciones de los usuarios con el sistema deben favorecer la calidad de su vida. El usuario debe ser tratado con respeto. El diseño debe ser estético y placentero, en donde los valores artísticos se igualen a los valores funcionales.
13. **Privacidad:** el sistema debe ayudar al usuario a proteger la información personal o privada, tanto la que pertenece al propio usuario como la que pertenece a los clientes del usuario.

A continuación se detallan las sub-heurísticas (o sub-PHs) definidos por D. Pierotti tanto para los 10 PHs de Nieslen (puntos 1 al 10 de la siguiente lista) como para los tres PHs agregados por ella misma (puntos 11 a 13 de la siguiente lista) [Pie04]:

#### 1. Ítems definidos para PH "Visibilidad del Estado del Sistema":

- 1.1. ¿Cada parte de la interfaz comienza con un título o encabezamiento que describa el contenido de la pantalla?
- 1.2. ¿El esquema de diseño de los íconos y su estética es consistente en todo el sistema?
- 1.3. Cuando se selecciona un icono particular rodeado por otros íconos, ¿se distingue el icono seleccionado claramente?
- 1.4. Los menús de instrucciones, puntos de entrada de datos y mensajes de error ¿aparecen en el mismo lugar de la pantalla o en el mismo menú?

- 1.5. En pantallas múltiples para entrada de datos ¿cada página esta etiquetada para mostrar su relación con las otras?
- 1.6. Si el sistema provee de los modos de sobre-escritura y de inserción, ¿hay información visible de cuál de los dos modos está activado?
- 1.7. Si se utilizan ventanas emergentes (pop-up) para mostrar mensajes de error, ¿permiten estas ventanas que el usuario visualice el error en la interfaz cuando se despliegan?
- 1.8. ¿Hay algún tipo de "feedback" para cada acción u operación?
- 1.9. Luego de que usuario completa una acción o un grupo de acciones, ¿el "feedback" del sistema indica que el siguiente grupo de acciones puede comenzarse?
- 1.10. ¿El sistema provee algún tipo de "feedback" visual en menús o cajas de diálogo que indiquen las opciones que pueden seleccionarse?
- 1.11. ¿El sistema provee algún tipo de "feedback" visual en menús o cajas de diálogo que indiquen en cuál de las posibles opciones se halla posicionado el cursor o el puntero del ratón?
- 1.12. Si hay menús o cajas de diálogo en donde pueden seleccionarse múltiples opciones, ¿el sistema provee algún tipo de "feedback" visual que indique cuáles son las opciones que ya han sido seleccionadas?
- 1.13. ¿Hay algún tipo de "feedback" visual cuando los objetos de la interfaz son seleccionados o movidos?
- 1.14. Es estado actual de cada icono, ¿es claramente indicado?
- 1.15. ¿Existe "feedback" cuando una tecla de función es presionada?
- 1.16. Si existen demoras mayores a 15 segundos en las respuestas del sistema, ¿el usuario es informado del progreso en la concreción de la respuesta?
- 1.17. ¿Los tiempos de respuestas son apropiados para cada tarea?
- 1.18. Tiempo de escritura, movimiento del cursor o selección con el ratón: entre 0,5 y 1,5 milisegundos.
- 1.19. Tiempo de respuesta de preguntas frecuentes: menos de 1 segundo
- 1.20. Tareas más comunes: 2 a 4 segundos
- 1.21. Tareas complejas: 8 a 12 segundos
- 1.22. ¿Los tiempos de respuesta del sistema son adecuados al proceso cognitivo del usuario?
- 1.23. Necesidad de continuar un mismo proceso de pensamiento donde cierta información debe ser retenida por el usuario: menos de 2 segundos.
- 1.24. No son necesarios altos niveles de concentración y no es requerido retener información: 2 a 15 segundos
- 1.25. La terminología utilizada en los menús, ¿es consistente con el dominio de conocimiento del usuario en relación a la tarea a realizar?
- 1.26. ¿El sistema provee visibilidad? Es decir, ¿el usuario puede expresar verbalmente cuál es el estado del sistema y que alternativas de acción posee en un determinado momento?
- 1.27. Los menús gráficos (GUI) ¿muestran de manera obvia cuál es el ítem que ha sido seleccionado?
- 1.28. Los menús gráficos (GUI), ¿muestran de manera clara las opciones que pueden ser deseleccionada?

- 1.29. Si los usuarios navegan entre diferentes pantallas del sistema, ¿el sistema utiliza etiquetas conceptuales, mapa de menús o marcas de navegación a modo de ayudas para esa navegación?

## **2. Ítems definidos para PH "Lenguaje de los Usuarios":**

- 2.1. ¿Los íconos son concretos y familiares para el usuario?
- 2.2. Dados un determinado usuario, una determinada lista de nombres de ítems y variables para realizar tareas. ¿las opciones en los menús (nombres de los ítems) están ordenadas en la manera más lógica para el usuario?
- 2.3. Si existe una secuencia natural para la selección de elementos en un menú, ¿esta implementada esa secuencia?
- 2.4. Los campos relacionados e interdependientes, ¿aparecen en la misma pantalla?
- 2.5. Si las formas de los objetos de la interfaz son utilizados como pistas visuales, ¿conducen con las convenciones culturales de los usuarios?
- 2.6. Los colores seleccionados, ¿corresponden a valores esperados según un los códigos de los usuarios?
- 2.7. Cuando una tecla o botón virtual para presionar en la pantalla (prompt) implica una acción necesaria, ¿incluye un mensaje con palabras consistentes con esa acción?
- 2.8. Las referencias indicadas en las teclas o botones virtuales de la interfaz para presionar en la pantalla (prompts), ¿son consistentes con nombres de teclas reales?
- 2.9. Cuando se ingresan datos en la pantalla, ¿la terminología utilizada para describir la tarea es familiar para los usuarios?
- 2.10. ¿El sistema provee teclas o botones virtuales de acceso por niveles (field-level prompts) en las pantallas de entrada de datos?
- 2.11. Cuando la pantalla incluye preguntas que debe ser respondidas, ¿el lenguaje de esas preguntas es simple y claro?
- 2.12. Las opciones en los menús, ¿se corresponden lógicamente con categorías que tengan un significado unívoco?
- 2.13. Los títulos de los menús, ¿siguen un mismo estilo gramatical?
- 2.14. El lenguaje de comandos empleado, ¿utiliza la jerga de los usuarios evitando el uso vocablos computacionales específicos?
- 2.15. Los nombres de los comandos, ¿son mas bien específicos antes que generales?
- 2.16. El lenguaje utilizado en los comandos, ¿permite utilizar tanto palabras completas como abreviaturas?
- 2.17. ¿Son entendibles los códigos para ingreso de datos?
- 2.18. Las combinaciones de secuencias de letras extrañas o poco frecuentes, ¿son omitidas siempre que sea posible?
- 2.19. El sistema ingresa/elimina automáticamente espacios en blanco (o ceros) a fin de alinear cifras con respecto al punto decimal?
- 2.20. ¿El sistema ingresa de manera automática los signos de dólar y decimal cuando se insertan valores monetarios?

- 2.21. ¿El sistema ingresa de manera automática comas en valores superiores a 9999?
- 2.22. ¿Los menús gráficos (GUI) ofrecen activación? Esto es, ¿es obvia la manera en que el sistema indica "ahora, haga esto"?
- 2.23. ¿El sistema ha sido diseñado de tal manera que las teclas con nombres similares no ejecuten acciones opuestas (y/o potencialmente peligrosas)?
- 2.24. ¿Las teclas de función están claramente etiquetadas y se distinguen con facilidad, aún cuando esto implique romper la consistencia en las reglas?

### **3. Ítems definidos para PH "Control y Libertad para el Usuario":**

- 3.1 Si configurar una pantalla es una tarea poco frecuente, ¿es esta tarea particularmente fácil de recordar?
- 3.2 En sistemas que permitan el uso de ventanas superpuestas, ¿es fácil reacomodar (reubicar) esas ventanas en la pantalla?
- 3.3 En sistemas que permitan el uso de ventanas superpuestas, ¿es fácil para los usuarios cambiar de una ventana a otra?
- 3.4 Cuando una tarea efectuada por el usuario se completa, ¿el sistema espera alguna señal del usuario antes de procesar la tarea?
- 3.5 ¿Los usuarios pueden escribir por adelantado en un sistema con muchos menús anidados?
- 3.6 ¿Se pregunta al usuario que confirme acciones que tendrán consecuencias drásticas, negativas o destructivas?
- 3.7 ¿Existe una función para "deshacer" al nivel de cada acción simple, cada entrada de datos y cada grupo de acciones completadas?
- 3.8 ¿Los usuarios pueden cancelar operaciones en progreso?
- 3.9 ¿La edición de caracteres está permitida en los comandos?
- 3.10 ¿Los usuarios pueden reducir el tiempo de entrada de datos copiando y modificando datos existentes?
- 3.11 ¿La edición de caracteres está permitida en los campos de entrada de datos?
- 3.12 Si las listas de menús son largas (más de siete ítems), ¿pueden los usuarios seleccionar un ítem tanto moviendo el cursor como escribiendo un código mnemotécnico?
- 3.13 Si el sistema utiliza dispositivos de tipo puntero, ¿los usuarios tienen la opción tanto de hacer "clic" en una lista de ítems como de utilizar atajo usando el teclado?
- 3.14 Los menús son anchos (muchos ítems) antes que profundos (muchos niveles)?
- 3.15 Si el sistema posee menús de niveles múltiples, ¿existe algún mecanismo que permita a los usuarios regresar al menú previo?
- 3.16 Si los usuarios pueden regresar al menú previo, ¿pueden también cambiar su elección en el menú previo nuevamente accedido?
- 3.17 ¿Los usuarios pueden moverse hacia adelante o hacia atrás entre las opciones de campos o cajas de diálogo?

- 3.18 Si el sistema posee múltiples pantallas para entrada de datos, ¿los usuarios pueden moverse hacia delante o hacia atrás entre las páginas en el conjunto?
- 3.19 Si el sistema utiliza una interfaz de preguntas y respuestas, ¿pueden los usuarios regresar a la pregunta anterior o saltar hacia delante una pregunta?
- 3.20 Las teclas de funciones que pueden causar serias consecuencias, ¿poseen una característica para deshacer su acción?
- 3.21 ¿Los usuarios pueden revertir sus acciones de manera sencilla?
- 3.22 Si el sistema permite a los usuarios revertir sus acciones, ¿existe un mecanismo que permita deshacer varias acciones de manera simultánea?
- 3.23 ¿Los usuarios pueden configurar la apariencia de su propio sistema, sesión, archivo, y valores por defecto para la pantalla?

#### **4. Ítems definidos para PH "Consistencia y Estándares":**

- 4.1 ¿Los formatos de la compañía o de la industria han sido respetados de manera consistente a lo largo de las distintas pantallas del sistema?
- 4.2 ¿El abuso de letras en mayúsculas en la pantalla ha sido evitado?
- 4.3 ¿Las abreviaturas no incluyen punto?
- 4.4 ¿Los números enteros están justificados a derecha y los números reales alineados con respecto al punto decimal?
- 4.5 ¿Los iconos poseen etiqueta?
- 4.6 ¿No hay más de 12/20 tipos de iconos?
- 4.7 ¿Existe algún elemento visual que identifique la ventana activa?
- 4.8 ¿Cada ventana posee un título?
- 4.9 ¿Es posible utilizar las barras de desplazamiento horizontal y vertical en cada ventana?
- 4.10 ¿La estructura de menús coincide con la estructura de las tareas?
- 4.11 ¿Han sido establecidos estándares de la compañía o industriales para el diseño de los menús? ¿Están aplicados de manera consistente en todas las pantallas del sistema?
- 4.12 ¿Los menús son presentados de manera vertical?
- 4.13 Si una opción de un menú es la de "salir", ¿esta opción aparece como último ítem del menú?
- 4.14 ¿Los títulos de los menús están centrados o justificados a izquierda?
- 4.15 ¿Los ítems de los menús están justificados a izquierda, con un número o un elemento mnemotécnico precediendo el texto del ítem?
- 4.16 ¿Los apuntadores (prompts) embebidos dentro un ítem de un menú múltiple, se despliegan hacia la derecha de la etiqueta del ítem?
- 4.17 ¿Las instrucciones en línea aparecen en un lugar semejante a lo largo de las diferentes pantallas?
- 4.18 ¿Las etiquetas de campos y los campos se distinguen topográficamente entre sí?
- 4.19 ¿Las etiquetas de los campos mantienen una forma consistente entre una pantalla y otra?

- 4.20 ¿Los campos y las etiquetas están justificadas a izquierda para listas alfabéticas y a derecha para listas numéricas?
- 4.21 ¿Las etiquetas de campos aparecen a la izquierda de los campos sencillos y arriba de las listas de campos?
- 4.22 ¿Las técnicas para atraer la atención del usuario están utilizadas de manera cuidadosa?
- 4.23 Intensidad: sólo dos niveles
- 4.24 Tamaño: hasta cuatro veces
- 4.25 Fuentes: hasta tres tipos
- 4.26 Parpadeo (blink): dos a cuatro hertz
- 4.27 Color: hasta cuatro colores diferentes (colores adicionales utilizados ocasionalmente)
- 4.28 Sonido: tonos suaves para dispositivos de retroalimentación regular y bruscos para condiciones críticas.
- 4.29 ¿Las técnicas para atraer la atención del usuario están utilizadas solamente en condiciones excepcionales o para tareas dependientes del tiempo?
- 4.30 ¿Hay entre cuatro/siete colores como máximo, y pertenecen estos colores al espectro visible?
- 4.31 ¿Se provee una leyenda si los códigos de color son numerosos o difíciles de interpretar?
- 4.32 Se evitan los pares de colores espectralmente extremos y altamente cromáticos?
- 4.33 ¿Los azules saturados no se utilizan para texto u otro elemento pequeño?
- 4.34 ¿La información más relevante está posicionada al comienzo del apuntador (prompt)?
- 4.35 ¿Las acciones del usuario están nombradas de manera consistente a lo largo de los diferentes apuntadores del sistema?
- 4.36 ¿Los objetos del sistema están nombrados de manera consistente a lo largo de los diferentes apuntadores del sistema?
- 4.37 ¿Los apuntadores de nivel de campo proveen más información que una reafirmación del nombre del campo?
- 4.38 Para interfaces de preguntas y respuestas, ¿las entradas válidas para una cuestión están listadas?
- 4.39 ¿Los nombres de las opciones en los menús son consistentes en relación a los demás nombres de ítems de los menús del sistema en cuanto al estilo gramatical y la terminología?
- 4.40 ¿La estructura de los nombres de las opciones en los menús coinciden con su correspondiente título de menú? Does the structure of menu choice names match their corresponding menu titles?
- 4.41 ¿Los comandos son utilizados de manera similar y poseen el mismo significado en todas las partes del sistema?
- 4.42 ¿Los comandos de lenguaje son consistentes, naturales y poseen una sintaxis fácil de memorizar?
- 4.43 ¿Las abreviaturas siguen una regla primaria simple? Si es necesario, ¿puede duplicarse una regla secundaria simple para abreviar una

palabra cuya abreviatura simple coincida con alguna abreviatura previamente utilizada?

- 4.44 ¿Es la regla secundaria del ítem anterior utilizada solo cuando es estrictamente necesario?
- 4.45 ¿Todas las abreviaturas poseen la misma longitud?
- 4.46 ¿La estructura de entrada de valores (datos) es consistente entre las diferentes pantallas?
- 4.47 ¿El método para mover el cursor hacia el campo anterior o posterior es consistente a lo largo del sistema?
- 4.48 Si el sistema posee pantallas múltiples para la entrada de datos, ¿tienen estas pantallas el mismo título?
- 4.49 Si el sistema posee pantallas múltiples para la entrada de datos, ¿las correspondientes pantallas están numeradas de manera secuencial?
- 4.50 ¿El sistema respeta las convenciones de la industria o los estándares de la compañía para asignar funciones a las teclas?
- 4.51 ¿Los colores altamente cromáticos son utilizados para atraer la atención del usuario?

#### **5. Ítems definidos para PH "Ayuda a los usuarios Reconocimiento, diagnóstico y recuperación de errores":**

- 5.1. ¿Los sonidos son utilizados para señalar errores?
- 5.2. ¿Los apuntadores son presentados de manera constructiva, sin necesidad de una crítica manifiesta o no manifiesta de los usuarios?
- 5.3. ¿Los apuntadores implican que el usuario tiene el control?
- 5.4. ¿Los apuntadores son breves e inequívocos?
- 5.5. ¿Los mensajes de error están expresado de manera tal que es el sistema, y no el usuario, quien se hace cargo de los errores?
- 5.6. Si se usan mensajes de error con humor, ¿son apropiados y respetuosos para la comunidad de usuarios?
- 5.7. ¿Los mensajes de error son gramaticalmente correctos?
- 5.8. ¿Los mensajes de error evitan el uso de signos de admiración?
- 5.9. ¿Los mensajes de error evitan el uso de palabras violentas u hostiles?
- 5.10. ¿Los mensajes de error evitan el tono antropomórfico?
- 5.11. ¿Todos los mensajes de error del sistema utilizan un estilo gramatical, una terminología, una forma y abreviaturas consistentes?
- 5.12. ¿Los mensajes colocan al sistema bajo el control del usuario?
- 5.13. ¿El lenguaje de comandos utiliza la sintaxis habitual acción-objeto?
- 5.14. ¿El lenguaje de comandos evita las arbitrariedades y el uso no-inglés de signos de puntuación, con excepción de los símbolos conocidos por el usuario?
- 5.15. Si se detecta un error en un campo de entrada de datos, ¿el sistema posiciona el cursor en ese campo o lo resalta de alguna manera?
- 5.16. ¿Los mensajes de error informan al usuario sobre la severidad del error cometido?
- 5.17. ¿Los mensajes de error sugieren la causa del problema que los ha ocasionado?

- 5.18. ¿Los mensajes de error proporcionan información semántica apropiada?
- 5.19. ¿Los mensajes de error proveen información sintáctica apropiada?
- 5.20. ¿Los mensajes de error indican que acción debe realizar el usuario para corregir el error correspondiente?
- 5.21. Si el sistema esta pensado para que lo utilicen tanto usuarios expertos como novatos, ¿existen diferentes niveles de complejidad en los mensajes de error disponibles?

## **6. Ítems definidos para PH "Prevención de errores":**

- 6.1. Si la base de datos incluye grupos de datos, ¿los usuarios pueden entrar más de un grupo en una única pantalla?
- 6.2. ¿Se han usado puntos o guiones bajos (underscores) para indicar la longitud de los campos?
- 6.3. ¿El nombre de la elección del menú en un menú de nivel superior se usa como título de menú para el menú de nivel inferior?
- 6.4. ¿Las elecciones disponibles en el menú son lógicas, distinguidas entre sí y mutuamente excluyentes?
- 6.5. ¿Las entradas de datos son no sensibles a mayúsculas siempre que sea posible?
- 6.6. Si el sistema muestra múltiples ventanas, ¿es la navegación entre ellas simple y visible?
- 6.7. Aquellas teclas de función que pueden causar las peores consecuencias ¿se encuentran ubicadas en posiciones del teclado difíciles de alcanzar?
- 6.8. Aquellas teclas de función que pueden causar las peores consecuencias ¿se encuentran alejadas de las teclas cuyo uso es intensivo pero no tiene mayores consecuencias?
- 6.9. ¿Se ha minimizado el uso de las teclas calificadoras (qualifier keys)?
- 6.10. Si el sistema utiliza teclas calificadoras, ¿se usa a las mismas consistentemente en todo el sistema?
- 6.11. ¿El sistema previene a los usuarios de cometer errores siempre que esto es posible?
- 6.12. ¿El sistema alerta a los usuarios si están a punto de cometer un error potencialmente serio?
- 6.13. ¿El sistema interpreta inteligentemente las posibles variaciones en los comandos de los usuarios?
- 6.14. ¿Las pantallas para entrada de datos y cajas de diálogo indican el número de espacios en caracteres que están disponibles para un campo?
- 6.15. Los campos en las pantallas de entradas de datos y las cajas de diálogo, ¿contienen valores por defecto cuando corresponde?

## **7. Ítems definidos para PH "Reconocimiento antes que Cancelación":**

- 7.1. ¿Existen pistas visuales y espacios en blanco para distinguir preguntas, apuntadores, puntos de inserción de respuestas e instrucciones en las interfaces de preguntas y respuestas?

- 7.2. ¿El despliegue de datos comienza en la parte superior izquierda de la pantalla?
- 7.3. ¿Las etiquetas de más de una palabra están posicionadas de manera horizontal (no desplegadas de manera vertical)?
- 7.4. ¿Todos los datos que el usuario necesita se muestran en cada paso de una transacción?
- 7.5. ¿Los apuntadores, pistas visuales y mensajes están posicionados en lugares de la pantalla en donde es probable que el usuario dirija su mirada?
- 7.6. ¿Los apuntadores presentan un formato que utilice espacios en blanco, justificaciones y elementos o guías visuales para un fácil reconocimiento?
- 7.7. ¿Las áreas de texto tienen "espacios de respiración" que las rodeen?
- 7.8. ¿Existe una distinción visual obvia entre los menús en donde solo es posible seleccionar una opción y los menús en donde es posible seleccionar múltiples opciones?
- 7.9. ¿Se han preservado las relaciones espaciales entre teclas de función "blandas" (mostradas como elementos en pantalla) y teclas de función "de teclado"?
- 7.10. ¿El sistema muestra un grisáceo o borra las etiquetas de aquellas teclas de función "blandas" que estén actualmente inactivas?
- 7.11. ¿Se usa el espacio en blanco para crear simetría y guiar al ojo del usuario en la dirección apropiada?
- 7.12. ¿Se han agrupado los ítems en zonas lógicas, utilizando encabezamientos para distinguir entre dichas zonas?
- 7.13. Las zonas ¿tienen como máximo entre doce y catorce caracteres de ancho, y entre seis y siete líneas de alto?
- 7.14. ¿Las zonas han sido separadas por espacios, líneas, color, letras, títulos resaltados, líneas de separación o áreas sombreadas?
- 7.15. ¿Las etiquetas de los campos están cercanas a los mismos, pero separadas de éstos por al menos un espacio en blanco?
- 7.16. ¿Los campos en columna que son largos se descomponen en grupos de cinco, separados por una línea en blanco?
- 7.17. ¿Los campos de entrada de datos que son opcionales están claramente marcados?
- 7.18. ¿Los símbolos se usan para cortar cadenas de entrada de gran longitud en "bloques" (chunks)?
- 7.19. ¿Se utiliza video grabado anteriormente o realce de colores para lograr la atención del usuario?
- 7.20. ¿Se utiliza video grabado anteriormente para indicar que un ítem ha sido escogido?
- 7.21. ¿Se utiliza tamaño de letra, realce de fuente, subrayado, color, sombreado o tipografía especial para mostrar la cantidad relativa o importancia de los diferentes ítems en pantalla?
- 7.22. ¿Se utilizan los bordes para identificar grupos significativos?
- 7.23. ¿Se ha utilizado el mismo color para agrupar elementos relacionados?
- 7.24. ¿La codificación de color es consistente dentro de todo el sistema?
- 7.25. ¿El color se usa en conjunción con algún otro elemento redundante?

- 7.26. ¿Existe buen contraste de brillo y de color entre los colores usados para imágenes y fondo?
- 7.27. ¿Los colores suaves, brillantes y saturados se han utilizado para enfatizar datos, mientras que los colores oscuros, opacos y no saturados han sido usados para des-enfatizar datos?
- 7.28. La primer palabra de cada opción del menú, ¿es la más importante?
- 7.29. ¿El sistema provee un mapeo que hace que el usuario perciba que existen relaciones entre los controles y las acciones asociadas?
- 7.30. Los códigos para ingreso de datos ¿son distintivos?
- 7.31. ¿Se han eliminado pares de datos frecuentemente confusos cada vez que fuera posible?
- 7.32. ¿Aquellas secuencias de números o letras que tienen gran longitud se han descompuesto en "bloques" (chunks)?
- 7.33. ¿Los ítems inactivos en un menú aparecen en gris o están omitidos?
- 7.34. ¿Existen elecciones por defecto dentro del menú?
- 7.35. En caso en que el sistema tenga muchos niveles de menús (o niveles de menús complejos), ¿los usuarios tienen acceso a un mapa espacial en línea de los menús existentes?
- 7.36. Los menús GUI ¿poseen "affordance" (esto es, hacen que resulte obvio donde es posible realizar una selección) ?
- 7.37. ¿Existen elementos visuales llamativos para identificar cuál es la ventana activa?
- 7.38. ¿Las teclas de función se encuentran organizadas en grupos lógicos?
- 7.39. ¿Las pantallas de entrada de datos y las cajas de diálogo indican dónde los campos son opcionales?
- 7.40. En las pantallas de entrada de datos y en las cajas de diálogo, ¿los campos dependientes se muestran sólo cuando es necesario?

## **8. Ítems definidos para PH "Flexibilidad y eficiencia de uso":**

- 8.1. Si el sistema soporta tanto a usuarios novicios y expertos, ¿se encuentran disponibles múltiples niveles de mensaje de error?
- 8.2. ¿El sistema permite que los usuarios novicios usen una "gramática de palabras clave" (keyword grammar) y los expertos una "gramática posicional"?
- 8.3. ¿Pueden los usuarios definir sus propios sinónimos para comandos?
- 8.4. ¿Permite el sistema que los usuarios novicios entren la forma más simple y común de cada comando, y permitan a los usuarios expertos añadir parámetros?
- 8.5. ¿Los usuarios expertos tienen la opción de ingresar comandos múltiples en una única cadena de texto?
- 8.6. ¿El sistema provee teclas de función para comandos de alta frecuencia?
- 8.7. Para pantallas de entrada de datos con muchos campos o en los cuales los documentos fuentes pueden estar incompletos, ¿tienen los usuarios la posibilidad de grabar una pantalla parcialmente completada?
- 8.8. ¿El sistema automáticamente ingresa ceros por delante para alineación de valores "(leading zeros)"?

- 8.9. Si las listas de menús son cortas (siete ítems o menos), ¿pueden los usuarios seleccionar un ítem moviendo el cursor?
- 8.10. Si el sistema utiliza la estrategia de teclear por adelantado (type-ahead), ¿los ítems de menú tienen asociados códigos mnemónicos?
- 8.11. Si el sistema usa un dispositivo apuntador, ¿los usuarios tienen la opción de hacer "clic" directamente sobre los campos o utilizar un atajo de teclado?
- 8.12. ¿El sistema ofrece atajos para "encontrar siguiente" y "encontrar previo" en búsquedas en bases de datos?
- 8.13. En las pantallas de entradas de datos, ¿los usuarios tienen la opción de hacer "clic" directamente sobre un campo o utilizar un atajo de teclado?
- 8.14. En los menús, ¿los usuarios tienen la opción o bien de hacer "clic" directamente en un ítem del menú o utilizar un atajo de teclado?
- 8.15. En las cajas de diálogo, ¿los usuarios tienen la opción de hacer "clic" directamente en la opción de la caja de diálogo o de utilizar un atajo del teclado?
- 8.16. ¿Los usuarios expertos pueden saltar las cajas de diálogos anidadas ya sea a través de teclear por adelantado (type-ahead), con macros definidas por el usuario o con atajos de teclado?

## **9. Ítems definidos para PH "Estética de diálogos y diseño minimalista":**

- 9.1. ¿La información esencial para tomar decisiones (y solo esta información) es mostrada en la pantalla?
- 9.2. ¿Los íconos son visualmente distinguibles de acuerdo a su significado conceptual?
- 9.3. Los objetos extensos, las líneas resaltadas y las áreas simples de la pantalla, ¿se distinguen de los íconos?
- 9.4. ¿Cada icono está resaltado con respecto a su fondo?
- 9.5. Si el sistema utiliza interfaces gráficas estándares (GUI) donde la secuencia de los menús ya ha sido especificada, ¿los menús están diseñados respetando esa especificación siempre que es posible?
- 9.6. ¿Los grupos de ítem con significado semejante, están separados por espacios en blanco?
- 9.7. ¿Cada pantalla de entrada de datos incluye un título simple, corto, claro y suficientemente distintivo?
- 9.8. ¿Las etiquetas de los campos son familiares y descriptivas?
- 9.9. ¿Los apuntadores están expresados de manera positiva y escritos utilizando el estilo de la voz activa?
- 9.10. ¿Cada opción de menú posicionada en un nivel inferior está asociada con solo una opción del nivel superior?
- 9.11. ¿Los títulos de los menús son breves pero suficientemente largos como para comunicar su contenido?

- 9.12. Los menús emergentes (pop-up or pull-down menus) con campos para entradas de datos, ¿poseen varias opciones para entrar esos datos definidas de manera correcta?

#### **10. Ítems definidos para PH "Ayuda general y documentación":**

- 10.1. Si los usuarios trabajan desde el disco rígido, ¿las partes del disco rígido que se hallan conectadas en línea (online) están marcadas?
- 10.2. ¿Las instrucciones en línea se distinguen visualmente?
- 10.3. ¿Las instrucciones siguen la secuencia de las acciones del usuario?
- 10.4. Si las opciones de los menús son ambiguas, ¿el sistema provee información aclaratoria adicional cuando un ítem es seleccionado?
- 10.5. ¿Las cajas de entrada de datos y de diálogos pueden ser utilizadas en línea para completar acciones?
- 10.6. Idem a ítem 10.4
- 10.7. ¿Hay ayudas de memoria para los comandos, ya sea a través de referencias rápidas en línea o apuntadores?
- 10.8. ¿La función de ayuda del menú es visible? (por ejemplo, una tecla etiquetada AYUDA o un menú especial?
- 10.9. La interfaz de ayuda del sistema (navegación, presentación, y conversación) ¿es consistente con las interfases de navegación, conversación y presentación de la aplicación que soporta?
- 10.10. Navegación: la información ¿es fácil de encontrar?
- 10.11. Presentación: ¿la disposición visual está bien diseñada?
- 10.12. Conversación: ¿la información es exacta, completa y comprensible?  
¿La información es relevante?
- 10.13. Orientación a la meta (¿qué puedo hacer yo con este programa?)
- 10.14. Descriptivo (¿para qué es esta cosa?)
- 10.15. Procedimental (¿cómo hago yo para hacer esta tarea?)
- 10.16. Interpretativo (¿por qué sucedió eso?)
- 10.17. Navegacional (¿dónde estoy?)
- 10.18. ¿Existe ayuda sensible al contexto?
- 10.19. ¿Puede el usuario cambiar el nivel de detalle disponible?
- 10.20. ¿Pueden los usuarios cambiar fácilmente entre la ayuda y su trabajo?
- 10.21. Tras haber accedido a la ayuda ¿pueden los usuarios continuar con su trabajo desde donde lo dejaron interrumpido?
- 10.22. ¿Es fácil acceder y regresar del sistema de ayuda?
- 10.23. Tras haber accedido a la ayuda ¿pueden los usuarios continuar con su trabajo desde donde lo dejaron interrumpido?

#### **11. Ítems definidos para PH "Habilidades":**

- 11.1. ¿Pueden los usuarios elegir entre la presentación de información en forma de texto o con iconos?
- 11.2. ¿Las operaciones para ventanas son fáciles de aprender y usar?

- 11.3. Si los usuarios son expertos, la utilización es frecuente, o el sistema tiene un bajo tiempo de respuesta, ¿hay en tal caso menos pantallas (más información por pantalla)?
- 11.4. Si los usuarios son novicios, la utilización es infrecuente o el sistema tiene un tiempo de respuesta rápido, ¿hay más pantallas (menos información por pantalla)?
- 11.5. ¿El sistema codifica automáticamente los ítems con color, con esfuerzo escaso o nulo por parte del usuario?
- 11.6. Si el sistema soporta tanto usuarios expertos como novicios, ¿hay múltiples niveles de detalle disponibles?
- 11.7. ¿Son los usuarios los iniciadores de las acciones antes que ser quienes deben responder ante ellas?
- 11.8. ¿El sistema realiza traducciones de datos para los usuarios?
- 11.9. ¿En los valores para campos se evita mezclar caracteres numéricos y alfabéticos siempre que sea posible?
- 11.10. Si el sistema tiene menús profundos (varios niveles), ¿los usuarios tienen la opción de teclear por adelantado?
- 11.11. Cuando el usuario accede a una pantalla o una caja de diálogo, ¿el cursor ya está posicionado en el campo que más probablemente el usuario vaya a necesitar?
- 11.12. ¿Pueden los usuarios moverse hacia adelante y hacia atrás dentro de un campo?
- 11.13. ¿El método para mover el cursor al campo siguiente o previo es simple y a la vez visible?
- 11.14. ¿Se ha evitado la auto-tabulación excepto cuando los campos tienen longitudes fijas o los usuarios son experimentados?
- 11.15. ¿Los dispositivos de entrada escogidos coinciden con las capacidades del usuario?
- 11.16. ¿Las teclas de cursor se encuentran dispuestas en forma de T invertida (mejor disposición para expertos) o en forma de cruz (mejor configuración para novicios)?
- 11.17. ¿Las teclas importantes (por ejemplo ENTER o TAB) son más grandes que las demás teclas?
- 11.18. Hay suficientes teclas de función para soportar funcionalidad, pero no tantas que sea difícil su detección y reconocimiento?
- 11.19. ¿Las teclas de función están reservadas para funciones genéricas, de alta frecuencia e importantes?
- 11.20. Las asignaciones de teclas de función ¿son consistentes a través de pantallas, subsistemas y productos relacionados?
- 11.21. ¿El sistema anticipa y avisa al usuario correctamente acerca de la próxima actividad que sea más probable?

## **12. Ítems definidos para PH "Interacción con el Usuario Placentera y Respetuosa":**

- 12.1. ¿Es cada icono individual un miembro armonioso dentro de una familia de iconos?
- 12.2. ¿Se ha evitado el detalle excesivo en el diseño de iconos?
- 12.3. ¿Se ha usado el color con discreción?

- 12.4. ¿La cantidad de administración de ventanas requerida se ha mantenido a un mínimo?
- 12.5. Si los usuarios están trabajando a partir de una copia impresa ¿el diseño de pantalla coincide con el formulario en papel?
- 12.6. ¿El color se ha usado específicamente para llamar la atención, comunicar la organización, indicar cambios de status y establecer relaciones?
- 12.7. ¿Los usuarios pueden desactivar la codificación automática de color si fuera necesario?
- 12.8. ¿Los requerimientos de tecleo son mínimos para las interfases de pregunta y respuesta?
- 12.9. ¿Los dispositivos de entrada seleccionados concuerdan con las restricciones del medio-ambiente?
- 12.10. Si el sistema utiliza dispositivos de entrada múltiple, ¿se ha minimizado el movimiento de mano y ojos entre los dispositivos de entrada?
- 12.11. Si el sistema soporta tareas gráficas, ¿se ha provisto un dispositivo apuntador alternativo?
- 12.12. ¿El teclado numérico se encuentra localizado a la derecha del área de teclas alfabéticas?
- 12.13. ¿Las teclas de función usadas más frecuentemente se encuentran en las posiciones más accesibles?
- 12.14. ¿El sistema completa entradas parciales inequívocas en un campo de entrada de datos?

### **13. Ítems definidos para PH "Privacidad":**

- 13.1. ¿Las áreas protegidas son completamente inaccesibles?
- 13.2. ¿Puede accederse con ciertas palabras claves a las áreas confidenciales o protegidas?
- 13.3. ¿Es la característica del punto anterior efectiva y exitosa?

## **3.3 Principios heurísticos de Constantine**

En el año 1994 Larry Constantine propone varios principios de usabilidad que deben aplicarse para alcanzar el desarrollo de un interfaz altamente usable (para más detalle sobre los principios de L. Constantine consultar <http://www.foruse.com/>). Esos principios, que actualmente se consideran PHs generales capaces de guiar la definición de una EH, fueron planteados por su autor como la siguiente lista:

**Estructura:** organizar la información agrupada por su significado.

**Simplicidad:** hacer fáciles las tareas comunes que el usuario realice habitualmente.

**Visibilidad:** mostrar toda aquella información necesaria para una tarea que deba realizar el usuario.

**Retroalimentación:** mantener informados a los usuarios en todo momento según las acciones que hayan realizado.

**Tolerancia:** permitir a los usuarios en todo momento: cancelar, deshacer, volver.

**Reutilización:** reducir la necesidad de los usuarios de recordar.

## 3.4 Principios heurísticos para Web de Instone

En el año 1996 Keith Instone redacta un informe técnico denominado "Usability Engineering on the Web" (para más detalle ver <http://instone.org/node/12>). En ese informe, considerado actualmente como uno de los pilares en cuanto a la definición de PHs para la web, la autora enumera a una serie de principios básicos a tener en cuenta al diseñar una interfaz para la World Wide Web. Esos principios se resumen en los siguientes puntos:

**Diálogo simple y natural.** Llevar conversaciones al nivel del usuario.

**Hablar el lenguaje del usuario.** Utilizar el lenguaje del usuario, siempre que sea posible.

**Minimiza la carga de memoria del usuario.** Procurar que los datos que deba recordar el usuario sean de fácil acceso o estén presentes en la interfaz.

**Consistencia.** Ser consistentes en cuanto a diseño de la interfaz.

**Retroalimentación.** El sistema debe informar al usuario de los cambios producidos por alguna acción del usuario.

**Salidas claramente marcadas.** El usuario ha de poder identificar fácilmente como salir de la aplicación.

**Atajos.** Siempre que sea posible, pensar en los usuarios con experiencia y facilitarles atajos para que puedan llegar a la información más fácilmente.

**Buenos mensajes de error.** Cuando suceda un error informar al usuario, siempre que sea posible, de forma clara (no con códigos de errores).

**Prevención de errores.** Intentar minimizar todos los posibles errores que se puedan producir en la aplicación y controlarlos.

**Ayuda y documentación.** Informar al usuario o prestarle ayuda cuando la aplicación lo requiera.

## 3.5 Principios heurísticos para DCU de Mayhew

De manera similar a los autores anteriores, en el año 1999 D. Mayhew propuso una serie de principios para el Diseño de Sistemas Centrados en el Usuario o DCU [MAY99]. Estos principios pueden ser interpretados como guías en el momento del diseño de un sistema interactivo o como PHs a la hora de evaluar la calidad de una determinada interfaz desde el punto de vista del DCU. Los mismos son:

**Compatibilidad del usuario, del producto, de las tareas y de los procesos del sistema:** para que esté todo coordinado adecuadamente para que el producto final se adapte perfectamente al usuario que lo usará.

**Consistencia y robustez:** que el sistema no sea vulnerable a errores.

**Familiaridad:** un usuario que ya esté familiarizado con un sistema similar o anterior se adaptará mejor al nuevo producto.

**Simplicidad:** un sistema simple es más fácil de usar.

**Manipulación directa:** el usuario maneja directamente los elementos del sistema.

**Control:** el usuario en todo momento ha de tener el control del sistema.

**WYSIWYG:** para poder trabajar con un documento con el aspecto real que tendrá.

**Flexibilidad:** si el sistema es flexible, puede adaptarse a cualquier tipo de usuarios.

**Sensibilidad y feedback:** que el sistema **interactúe** con el usuario.

**Tecnología invisible:** que la tecnología usada en un sistema se mantenga invisible al usuario.

**Protección:** un sistema en el que sus datos queden a salvo de intrusos.

**Facilidad de aprendizaje y facilidad de uso:** para que un usuario pueda usar el sistema fácilmente.

## 3.6. Principios heurísticos para páginas de inicio de Nielsen y Tahir

Luego de un exhaustivo estudio de la página de inicio de más de 200 sitios web pertenecientes al contexto de uso anglosajón, Jacob Nielsen y Marie Tahir publican en el año 2002 un compendio de PHs para la evaluación de páginas de inicio [NITA02]. En esta obra, estos autores ilustran su trabajo con la puesta en marcha de evaluaciones de la usabilidad de 50 páginas de inicio pertenecientes al mencionado contexto de uso. Los 10 PHs más significativos planteados en [NITA02] se describen a continuación:

**Visibilidad del estado del sistema:** se necesita informar a los usuarios sobre lo que está aconteciendo, favoreciendo un "feedback" adecuado en un tiempo razonable

**Compatibilidad entre el sistema y el mundo real:** el sistema debe adecuar su lenguaje al utilizado por el usuario en vez de orientarlo hacia el sistema. Se deben seguir las convenciones del mundo real favoreciendo que la información aparezca de una manera natural y lógico para el usuario.

**Control y libertad del usuario:** usualmente los usuarios escogen funciones del sistema incorrectas. Debe proporcionárseles elementos claros que les permita actuar en situaciones indeseadas sin necesidad de recorrer un largo camino para deshacer las opciones que le llevaron al error.

**Consistencia y estándares:** los usuarios no deberían necesitar adivinar que diferentes conceptos o palabras significan cosas iguales. Se deben seguir las convenciones propias de la plataforma computacional que se este utilizando.

**Prevención de errores:** se deben incluir elementos que permitan prevenir los errores antes de favorecer los mensajes de diálogo una vez que los errores han sido cometidos.

**Reconocimiento más que recuerdo:** optar por la visibilidad y dotar a las metáforas gráficas de un gran "affordance". No añadir carteles que el usuario leerá en una parte del sistema necesarios para actuar en otra parte del sistema. Las instrucciones para el uso del sistema deben ser fácilmente accesibles por el usuario cada vez que lo requiera.

**Flexibilidad y eficiencia de uso:** Los usuarios novatos deberían poder convertirse en usuarios expertos con el tiempo de uso. Debe favorecerse distintos tiempos de respuesta del sistema de acuerdo a la experticia o perfil del usuario que esta interactuando con el mismo. Debe favorecerse que los usuarios expertos salteen acciones para ejecutar las tareas de manera más eficiente.

**Diseño estético y minimalista:** debe obviarse la información irrelevante. La información o elementos secundarios deben disminuir su visibilidad en favor de los elementos o la información crucial de la página web.

**Ayuda al usuario en el reconocimiento, diagnóstico y recuperación de errores:** los mensajes de error deben estar expresados de manera clara, simple y concisa. No deben incluir códigos. Es ideal que incluyan una serie de instrucciones que sugieran una solución.

**Ayuda y documentación:** aunque un sistema pueda ser utilizado sin ayuda, es conveniente incluirla en forma de "ayuda en línea". La información descripta en cada ayuda debe ser fácil de encontrar, debe estar dirigida hacia la acción del usuario y no extenderse demasiado.

## 3.7. Principios heurísticos para Web de Tognazzini

En el año 2003, Tognazzini publica en su sitio "Ask Tog" una serie de PHs básicos para el diseño de una interfaz (para visitar el sitio web "Ask Tog" consultar <http://www.asktog.com/basics/firstPrinciples.html>). Esos PHs son:

**Anticipación:** el sitio Web debe anticiparse a las necesidades del usuario.

**Autonomía:** los usuarios deben tener el control sobre el sitio Web. Los usuarios sienten que controlan un sitio web si conocen su situación en un entorno abarcable y no infinito.

**Ceguera al color:** Los colores han de utilizarse con precaución para no dificultar el acceso a los usuarios con problemas de distinción de colores (aprox. un 15% del total).

**Consistencia:** las aplicaciones deben ser consistentes con las expectativas de los usuarios, es decir, con su aprendizaje previo

**Configuraciones por defecto:** Los campos que contienen información por defecto deben ser lo seleccionado, de este modo, los usuarios puede modificar según su criterio rápida y fácilmente.

**Eficiencia del usuario:** los sitios Web se deben centrar en la productividad del usuario, no en la del propio sitio Web. Por ejemplo, en ocasiones tareas con mayor número de pasos son más rápidas de realizar para una persona que otras tareas con menos pasos, pero más complejas.

**Interfaces explorables:** hacer que las acciones sean reversibles, que siempre se encuentre la posibilidad de "deshacer" cualquier acción.

**Ley de Fitts:** que el tiempo para alcanzar un objetivo con el ratón esta en función de la distancia y el tamaño del objetivo. A menor distancia y mayor tamaño más facilidad para usar un mecanismo de interacción (Ver más sobre Ley de Fitts en <http://en.wikipedia.org/wiki/FittsLaw>).

**Objetos de interfaz humanos:** que los objetos de la **interfaz** sean lo mas parecido al mundo real de este modo a los usuarios les será familiares, y de fácil comprensión.

**Reducción de tiempos de latencia:** Hace posible optimizar el tiempo de espera del usuario, permitiendo la realización de otras tareas mientras se completa la previa e informando al usuario del tiempo pendiente para la finalización de la tarea.

**Aprendizaje:** los sitios Web deben requerir un mínimo proceso de aprendizaje y deben poder ser utilizados desde el primer momento.

**Uso de metáforas:** Su uso adecuado facilita el aprendizaje de un sitio Web, pero un uso inadecuado de estas puede dificultar enormemente el aprendizaje.

**Protección del trabajo del usuario:** se debe asegurar que los usuarios nunca pierden su trabajo como consecuencia de un error

**Legibilidad:** el color de los textos debe contrastar con el del fondo, y el tamaño de fuente debe ser suficientemente grande

**Seguir el estado:** conocer y almacenar información sobre el comportamiento previo del usuario ha de permitir posteriormente realizar operaciones frecuentes de manera más rápida

**Navegación visible:** Se deben evitar elementos invisibles de navegación que han de ser inferidos por los usuarios, menús desplegados, indicaciones ocultas, etc.

El conjunto de todos los PHs presentados anteriormente es potencialmente suficiente para llevar a cabo cualquier EH, pero en la práctica son demasiado generales y se muestran insuficientes para una evaluación eficiente. Es por ello que la EH incluye diferentes pasos de adecuación de los PHs generales que es necesario efectuar previamente a la Puesta en Marcha de la misma. Estos pasos son descritos en la próxima sección.

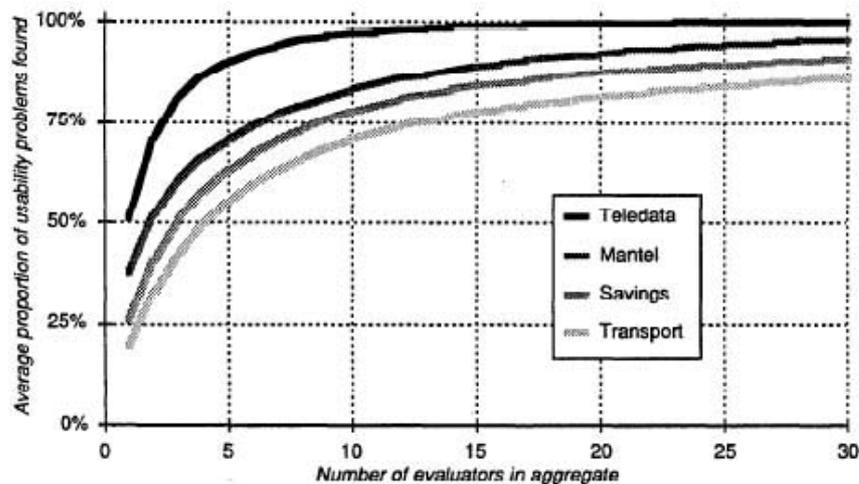
## 4. Pasos de una Evaluación Heurística

### 4.1. Planificación de una Evaluación Heurística

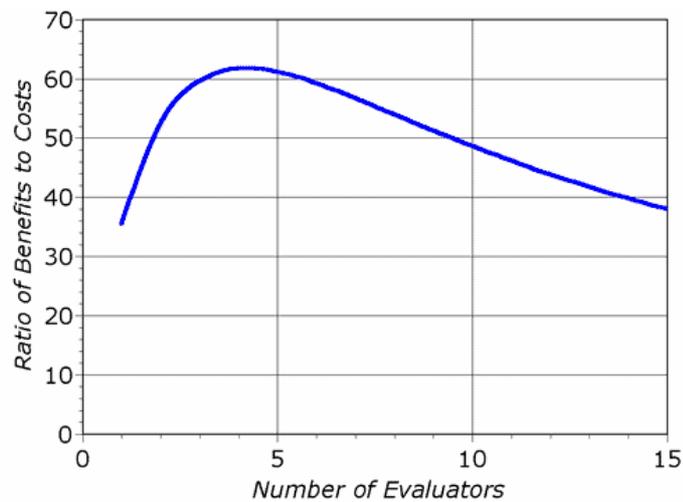
El primer paso de una EH es la planificación. Durante toda esta etapa es fundamental tener en cuenta que cada contexto de uso (universidades, compañías de seguros, entidades bancarias, administraciones públicas, etc.) posee sus normas o convenciones, las cuales deberían reflejarse en la interfaz de sus sitios y aplicaciones y en la forma de trabajo de sus usuarios. Es por ello que un paso crucial dentro de la etapa de planificación de la EH es la adecuación de cada criterio

que se utilizará al contexto de uso al cual pertenece el sistema que se desea evaluar [MOL90]. Primeramente, un paso crucial relacionado con la Planificación de una EH es la selección de los evaluadores que deberán llevar a cabo la etapa siguiente en el proceso de evaluación. Si esta selección se realiza como primer paso dentro de la Planificación, podrá contarse con la recomendable presencia de estas personas durante el resto de la Planificación. En este sentido, se ha observado que diferentes personas aplicando los mismos PHs encuentran diferentes tipos de problemas de usabilidad en un mismo sistema que se está evaluando. Es por ello que diversos autores sostienen que los resultados de una EH que involucre a un solo evaluador experto no serán suficientemente confiables [MOL90, NIE92, NITA02].

En relación a lo discutido en el párrafo anterior, una decisión trascendental para el posterior desarrollo de la EH es la selección de cuantos evaluadores se incluirán en la EH y cuál será el perfil de esos evaluadores. En este sentido, la Figura 1 muestra una gráfica que compara los cuatro experimentos (Teledata, Mantel, Savings y Transport) analizados por [MOL90]. En esta gráfica puede observarse como al realizar la EH con un solo evaluador no se encuentran tantos problemas de usabilidad como con cinco evaluadores. De la misma manera, también se observa que a partir de la inclusión de 15 evaluadores la detección de problemas de usabilidad se estabiliza. Otro resultado interesante con respecto a la cantidad de evaluadores a incluir en una EH se presenta en la Figura 2. En este caso, J. Nielsen analiza la relación costo-beneficio entre la cantidad de evaluadores involucrados en una EH y los beneficios obtenidos (Para más información consultar [http://www.useit.com/papers/heuristic/heuristic\\_evaluation.html](http://www.useit.com/papers/heuristic/heuristic_evaluation.html) )



**Figura 1** Proporción media de problemas de usabilidad encontrados entre treinta evaluadores [MOL90]

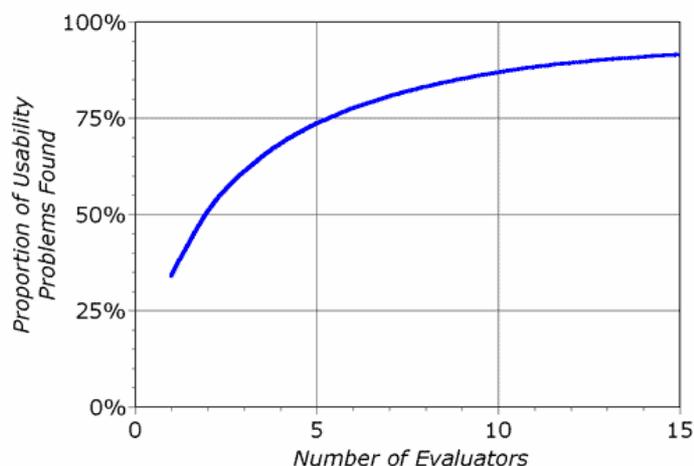


**Figura 2** Costo-beneficio en relación a la cantidad de evaluadores necesarios para llevar a cabo una EH. Extraído de "How to Conduct a Heuristic Evaluation" (J. Nielsen), en [http://www.useit.com/papers/heuristic/heuristic\\_evaluation.html](http://www.useit.com/papers/heuristic/heuristic_evaluation.html)

Por lo expuesto anteriormente, la metodología tradicional sostiene que el número de evaluadores necesario para realizar una EH no tiene que ser demasiado grande. Nielsen y Landauer [NIL93] proponen como ideal seleccionar entre tres a cinco evaluadores ya que, según estos autores, con ellos ya se encuentran aproximadamente el 75 % de los errores en un interfaz del sistema que se analice a través de la EH. Estos autores propusieron una función matemática que calcula la potencialidad de una EH (en función de la cantidad de problemas que serán detectados) de acuerdo a la siguiente ecuación:

$$\text{ProblemasEncontrados}(i) = N(1 - (1 - l)^i)$$

En donde ProblemasEncontrados(i) es el número de problemas de usabilidad encontrado durante i evaluaciones (es decir, i evaluadores en el mismo ciclo iterativo y con el mismo sistema o prototipo), N es el número total de problemas de usabilidad en la interfaz de usuario, e l es la probabilidad de encontrar un problema de usabilidad cuando el sistema se evalúa con un sólo evaluadora través de la metodología de la EH.



**Figura 3** Cálculo promedio de la función "ProblemasEncontrados" para seis interfaces[NIL93]. Extraído de "How to Conduct a Heuristic Evaluation" (J. Nielsen), en [http://www.useit.com/papers/heuristic/heuristic\\_evaluation.html](http://www.useit.com/papers/heuristic/heuristic_evaluation.html)

La Figura 3 muestra el cálculo de esta función para el caso de un ejemplo práctico presentado por J. Nielsen en donde se calcula la función "ProblemasEncontrados" para seis interfaces diferentes (Para detalles sobre las características de la experimentación realizada por J. Nielsen consultar [http://www.useit.com/papers/heuristic/heuristic\\_evaluation.html](http://www.useit.com/papers/heuristic/heuristic_evaluation.html)).

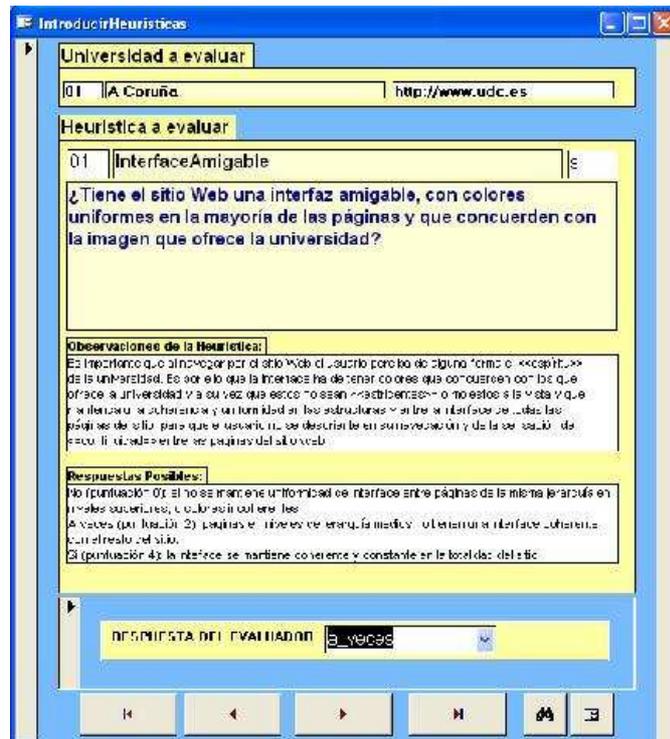
Como ya se menciona anteriormente, otro segundo aspecto relacionado con la selección de los evaluadores es su perfil ideal. Mientras que la metodología tradicional sugiere contar con evaluadores que tengan algún conocimiento sobre los principios de la usabilidad [MOL90]. Posteriormente, se creyó que el método podía ser más efectivo cuando los evaluadores eran expertos en usabilidad. Sin embargo, y en concordancia con el auge del Diseño Centrado en el Usuario, algunos autores proponen involucrar como evaluadores a posibles usuarios de la aplicación e incluso a desarrolladores de aplicaciones [NIE92, MUL98, GPL04]. Veamos las distintas características de cada tipo de evaluador:

**Los expertos en usabilidad:** según las posturas más tradicionales son los más apropiados para éste método porque es difícil que un desarrollador o un usuario pueda encontrar todos los problemas de usabilidad en una interfaz a partir de los criterios heurísticos [MOL90]. Sin embargo, autores como [GPL04] sostienen que el experto en usabilidad no tiene experiencia en el dominio de uso de la aplicación y tiende a informar solamente de problemas potenciales que otros perfiles de evaluadores no detectaría.

**Los desarrolladores:** los desarrolladores sin experiencia específica en usabilidad pueden cumplir el rol de evaluadores de una EH. Sin embargo, resultados obtenidos sobre distintas pruebas sugieren que este perfil de evaluadores no es suficientemente apropiado ya que tiende a concentrarse en problemas técnicos, sugiriendo cambios que están fuera del alcance de la IPO y que se relacionan más con la funcionalidad que con la interfaz del sistema que se desea evaluar.

**Los usuarios potenciales:** Los usuarios inexpertos pueden ser bastante confusos al realizar una EH, principalmente porque no pueden expresar correctamente los problemas que detectan y no son claros a la hora de plantear problemáticas que ayuden a los diseñadores a la posterior mejora de la interfaz evaluada. Sin embargo, se ha observado que si los usuarios conocen medianamente el sistema a evaluar o si son expertos en su uso o han sido involucrados en su desarrollo; entonces tienden a detectar problemas de usabilidad muy eficazmente [GLP04].

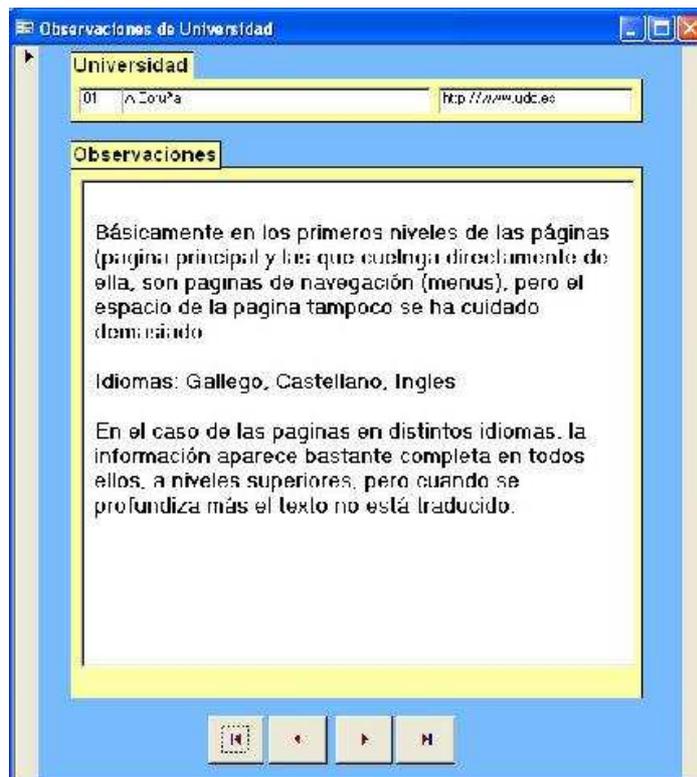
Una vez que el equipo evaluador de la usabilidad ha sido conformado se debe proceder a seleccionar que PHs incluirán en la evaluación a realizar, teniendo en cuenta el cubrimiento de los puntos clave del sistema a evaluar y tratando de evitar la posible solapación de los mismos. Una vez seleccionados los HPs que se incluirán debe encararse la delicada tarea de adecuarlos al contexto de uso a evaluar tal como se indica en el párrafo anterior. Es evidente que cuanto más adecuados al contexto de uso estén los HPs seleccionados, más relevantes serán los resultados obtenidos por la EH. Este tipo de problemáticas es estudiada por la llamada Usabilidad Transcultural [SHWP06].



**Figura 4.** Interfaz de UsabAIPO-GestorHeurística. Pantalla para ingreso de resultados.

En tercer lugar, cada PH considerado puede traducirse en una serie de preguntas o sub-PHs capaces de instanciarse de manera natural en el sistema a evaluar. Deben clasificarse los problemas que van a registrar asociándose a cada PH o sub-PH preguntas que posteriormente deberá contestar el evaluador durante el proceso de inspección. Además, debe elegirse una escala de valores para cada una de las posibles respuestas, indicando el significado de cada valor posible. Esta escala de valores puede incluir parámetros numéricos discretos (del tipo, 1,2,...,5), parámetros numéricos continuos (del tipo 0,5-1), parámetros alfabéticos (del tipo mucho-poco-nada), parámetros alfanuméricos (del tipo A1, A2,...,A10), etc. La definición de una escala de valores clara y concisa mejorará el análisis posterior de resultados, minimizándose la subjetividad de los evaluadores expertos y propiciándose el análisis comparativo de las diferentes puestas en marcha a ejecutar. Sin embargo, a veces algunas preguntas solo podrán ser respondidas por el equipo evaluador utilizando lenguaje natural.

Como resultado de la etapa de planificación de la EH debe obtenerse una especie de "planilla en blanco" donde se liste cada una de las preguntas a evaluar, se indique escala de valores y significado de cada valor, y se deje espacio para que durante la posterior Puesta en Marcha los evaluadores expertos completen la evaluación. Es importante dejar siempre espacio para las anotaciones personales de los evaluadores, las cuales pueden ayudar a interpretar resultados y aclarar puntos oscuros de la evaluación. Si la EH a realizar es muy compleja o muy grande, puede opcionalmente reemplazarse a esa planilla por la utilización de alguna herramienta informática que ayude a gestionar la EH. En este sentido, las Figuras 4 y 5 muestran la interfaz de la plataforma "UsabAIPO-GestorHeurística" [GLP06] creada dentro de la planificación de la Segunda Etapa de la Iniciativa UsabAIPO [LGP05, LGP06, GLPG06] y utilizada durante la Puesta en Marcha de la EH de 69 sitios web pertenecientes a la red "Universia" (ver página de inicio de red Universia en [www.universia.es](http://www.universia.es)).



**Figura 5.** Interfaz de UsabAIPO-GestorHeurística.

Pantalla para ingreso de comentarios en lenguaje natural.

La plataforma UsabAIPO-GestorHeurística [GLPG06] permite almacenar información pertinente a la Puesta en Marcha de la EH que se desee efectuar, datos de las interfaces a evaluar (en este caso páginas web de universidades) y las preguntas heurísticas asociadas a cada PH que se ha decidido incluir en la EH (Figura 4). Además, esta base de datos permite almacenar los resultados obtenidos al realizar la inspección de cada interfaz considerada. Consecuentemente, para el diseño de la interfaz de la plataforma UsabAIPO-GestorHeurística se tuvo en cuenta que toda la información asociada a cada PH incluido en la EH fuera altamente visible. De esta manera se intenta ayudar al evaluador brindándole diferentes datos que beneficien su ubicación temporal (momento de la evaluación) y espacial (interfaz que se está evaluando y PHs a considerar). En particular, UsabAIPO-GestorHeurística provee la facilidad de incluir el nombre de la interfaz que se está evaluando junto a información referida al PH que se está considerando (nombre, texto de la pregunta, observaciones asociadas y significado de cada posible respuesta). En cuanto a la necesidad de brindar al evaluador un sitio en donde introducir comentarios u, ocasionalmente, un nuevo PH a considerar, la plataforma UsabAIPO-GestorHeurística incluye una especie de bloc de notas (Figura 5) en donde el evaluador podrá puntualizar algún dato o añadir comentarios en lenguaje natural asociados a cada PH.

Cabe mencionar que si bien la potencialidad de este tipo de plataformas queda evidenciada en la Puesta en Marcha de la EH, su adecuación y preparación se realiza durante la etapa de planificación, ya que es necesario adecuar al gestor a utilizar introduciendo todos los datos particulares de la EH que se está llevando a cabo.

## 4.2. Puesta en Marcha de una Evaluación Heurística

Una vez concluida la etapa de Planificación debe procederse a la Puesta en Marcha de la EH. En esta etapa es imprescindible contar con la presencia de los evaluadores, quienes deberán preceder a la inspección de la interfaz del sistema que se esté evaluando. Con respecto a esta etapa, la metodología clásica asociada a la EH plantea que los evaluadores deben inspeccionar la interfaz del sistema a evaluar de manera individual y sólo después de la evaluación pueden comunicar sus hallazgos [NIE94, DR99]. Este procedimiento es importante para asegurar evaluaciones independientes e imparciales de cada evaluador. Es recomendable que los evaluadores examinen cada parte de la interfaz del sistema evaluado al menos dos veces para que se familiaricen con su estructura y antes de comenzar con la evaluación propiamente dicha [LAI04].

En cuanto a la temporización de la Puerta en Marcha de una EH, Nielsen [NIE94] recomienda sesiones de evaluación de aproximadamente una o dos horas por cada parte de la interfaz a evaluar. Los evaluadores utilizan la lista de comprobación generada en la etapa anterior y si es estrictamente necesario pueden incorporar nuevos PHs a los ya existentes. En general, las fases de la Puesta en Marcha de una EH son las siguientes [MOL90]:

**Entrenamiento previo a la evaluación:** Hay que dar a los evaluadores conocimiento del tema e información. No obstante, si el interfaz es del tipo de "llegar y usar" los evaluadores no necesitan esta introducción. Este tipo de interfaces son los que los usuarios deberían ser capaces de utilizar por sí mismos sin ningún tipo de instrucción. También es necesario que los evaluadores se familiaricen con los diferentes perfiles de usuarios que se consideran en la EH, ya que durante la segunda fase de la puesta en marcha deberán intentar emular su comportamiento frente a la interfaz del sistema a evaluar.

**Evaluación propiamente dicha:** Los evaluadores expertos evalúan el interfaz con el objetivo de detectar errores de usabilidad. Es conveniente que el entorno en el cual se desarrolla la EH sea similar para las distintas sesiones de inspección (espacio físico, condiciones ambientales, hora del día, nivel de ruido, luminosidad, etc), a fin de minimizar el impacto de factores externos que pueden afectar a la capacidad cognitiva de los evaluadores. Cada medición realizada es plasmada en la lista en blanco que ha sido generada en el paso anterior. Durante la evaluación, los evaluadores no tan expertos pueden guiarse con las heurísticas para chequear de este modo los posibles problemas de usabilidad del sistema.

Según el autor Jacob Nielsen (puede consultarse sobre este tema en <http://www.useit.com/papers/heuristic/severityrating.html>), cuando un evaluador experto puntúa la gravedad de un problema de usabilidad esa medida debería reflejar a los siguientes tres factores:

- La **frecuencia** con la que el problema ocurre, si es común o raro que pase una cosa.
- El **impacto** del problema cuando sucede, si los usuarios tendrán muchos problemas cuando pase.
- La **persistencia** del problema, si el problema es resuelto la primera vez que se use el sitio web o aparece repetidamente

**Puntuación:** En este punto se tiene que determinar la severidad de cada uno de los problemas encontrados de acuerdo a la escala de valores definida durante la Planificación. En cuanto a la jerarquización de la gravedad de los problemas de la interfaz evaluada, también es necesario evaluar el "impacto del mercado" de cada problema detectado, puesto que algunos problemas pueden ser de poca importancia pero afectar mucho a la imagen del sistema. En este sentido comienzan a perfilarse algunas propuestas que proponen añadir una puntuación adicional a los problemas de usabilidad detectados según la siguiente escala (consultar <http://www.useit.com/papers/heuristic/severityrating.html>):

[0=] No es un problema de usabilidad.

[1=] Problema sin importancia: no necesita arreglarse a menos que haya tiempo de sobra.

[2=] Problema de poca importancia: arreglarlo no tiene mucha importancia.

[3=] Problema grave: es importante arreglarlo.

[4=] Catástrofe: Es obligatorio arreglarlo.

Cabe destacar que diferentes autores coinciden en que es difícil conseguir que los evaluadores definan durante una sesión heurística unas buenas estimaciones de la "severidad" asociada a cada problema de usabilidad [MOL90, NIE92, DR99].

**Discusión:** Luego de determinar la puntuación de cada ítem observado en la interfaz que se está evaluando, es necesario establecer el grado de severidad de cada uno de los problemas encontrados. Esto se hace distribuyendo la lista completa de errores encontrados a todos los evaluadores, y puntuándola. Puede considerarse a este paso como parte de la Puesta en Marcha o como parte de la etapa de Análisis de Resultados de la EH (más adecuado en nuestra opinión).

De acuerdo a lo presentado en la Sección 4.1, si la EH que se está poniendo en marcha es lo suficientemente compleja o grande, la utilización de una plataforma que ayude a su gestión puede resultar de sumo interés. Un ejemplo de estas plataformas lo constituye el software UsabAIPO-GestorHeurística [GPL06] presentado dentro de la etapa de Planificación.

## 4.3. Análisis de resultados de una Evaluación Heurística

La EH es una metodología de evaluación de la usabilidad que busca resultados cualitativos que ayuden a enfatizar que problemas de usabilidad presenta la interfaz del sistema interactivo evaluado por sobre el cálculo de funciones matemáticas llamadas métricas, las cuales sintetizan en un solo valor numérico el resultado de la evaluación. Sin embargo, a pesar de su naturaleza cualitativa, la EH también puede brindar valiosos resultados cuantitativos que ayuden a tener una visión global de los resultados obtenidos en la Puesta en Marcha. Por lo tanto, concluiremos que el procesamiento de los resultados obtenidos durante la Puesta en Marcha de una EH puede ser llevado a cabo desde dos puntos de vista no excluyentes y complementarios:

1. Desde el **punto de vista cuantitativo** pueden utilizarse herramientas como la Estadística para condensar en gráficos y funciones matemáticas tanto los resultados obtenidos por un solo evaluador (visión "sumativa" de cada una de las inspecciones realizadas) como los resultados obtenidos por diferentes evaluadores (visión "comparativa" de resultados obtenidos). Estos resultados cuantitativos ayudarán a resumir de manera clara y precisa la ponderación de cada una de las preguntas que cristalizan a los PHs generales con respecto a la interfaz evaluada.
2. Desde el **punto de vista cualitativo** los evaluadores elaborarán una lista de problemas de usabilidad con ítems (problemas hallados) que serán justificados de acuerdo a los PHs que se hayan considerado en la EH. Según la metodología tradicional, el análisis de cada problema se ha de realizar por separado y no en conjunto. Sin embargo, nuevas metodologías como las presentadas en [GGL06] defienden la riqueza de un análisis cualitativo general de resultados que permita extrapolar patrones de comportamiento generales presentes en los datos provenientes de las distintas evaluaciones efectuadas por los diferentes. En este sentido herramientas como las Técnicas del Descubrimiento de Conocimiento en Base de Datos (Datamining) pueden ser utilizadas para hallar de manera algorítmica (es decir de ~~de~~ manera imparcial y objetiva) patrones de comportamiento desconocidos y potencialmente útiles presentes en los datos provenientes de cualquier evaluación cualitativa de la usabilidad, en particular provenientes de la Puesta en Marcha de una EH [GGL06].

Los resultados de la evaluación se pueden registrar como informes escritos de cada evaluador o haciendo que los evaluadores comuniquen verbalmente sus comentarios a un observador mientras inspeccionan la interfaz. Estos informes tienen la ventaja de presentar un expediente formal de la evaluación, pero requieren un esfuerzo adicional para los evaluadores y la necesidad de leerlo y diseñar un documento que integre el trabajo de todos los evaluadores por parte del encargado de la evaluación. En este sentido, el contar con una plataforma de software que ayude a gestionar la EH como la presentada en la etapa de Planificación puede minimizar considerablemente los esfuerzos posteriores a la Puesta en Marcha y ayudar a reflejar de manera más metódica y concreta los resultados obtenidos [GPL06]. Además, el uso de este tipo de soporte digital para la EH supone que los resultados de la evaluación están disponibles de manera inmediata cuando finaliza la etapa de la Puesta en Marcha **descrita** en la sección anterior.

En cuanto a la jerarquización de la gravedad de los problemas de la interfaz evaluada, aquí se pone de manifiesto la importancia de haber definido una escala de valores para cada una de las posibles respuestas asociadas a los HPs considerados durante la etapa de Planificación de la EH. Deberá prestarse especial atención a los comentarios añadidos por los evaluadores así como a posibles nuevos PHs que hayan tenido que incluirse con la finalidad de inspeccionar correctamente la interfaz. Algunos autores sugieren que es conveniente enviar un cuestionario a los evaluadores después de las sesiones reales de la evaluación para recoger los grados de la severidad, enumerando completamente el sistema según los problemas de usabilidad que se han descubierto, y pidiendo que clasifiquen la severidad de cada problema. Para ello, y debido a que cada evaluador ha identificado solamente un subconjunto de los problemas incluidos en la lista, los problemas necesitan ser descritos en la profundidad razonable, usando posiblemente ilustraciones. Las descripciones se pueden sintetizar agregando

comentarios hechos por los evaluadores que habían encontrado cada problema (si se utilizan los informes escritos de la evaluación, las descripciones se pueden sintetizar de las descripciones incluidas en los informes). Estas descripciones permiten que los evaluadores determinen varios problemas muy fácilmente incluso si no los han encontrado en su propia sesión de la evaluación. Típicamente, [MOL90] sugiere que se necesitan alrededor de 30 minutos para realizar estas puntuaciones, pero es importante observar que cada evaluador debe proporcionar grados individuales de la severidad independientemente de los otros evaluadores.

## 5. Conclusiones

La evaluación es un aspecto fundamental a tener en cuenta en el diseño de sistemas interactivos. La evaluación heurística (EH) es un método por inspección de sencilla aplicación y relativamente económico que puede utilizarse en cualquier etapa del desarrollo de un sistema interactivo. Ha sido demostrado empíricamente que la EH detecta un promedio de aproximadamente 75% del total de problemas de usabilidad de la interfaz de cualquier sistema interactivo, lo que lo postula como un método relativamente confiable y altamente recomendable [NIE93, DR99].

En este Capítulo hemos visto que existe una amplia bibliografía a la hora de definir los Principios Heurísticos (PHs) generales que sustenten teóricamente a las heurísticas que guíen a una HE. Cabe destacar que en una HE real no es necesario limitarse a un solo autor de los aquí presentados, sino que los distintos PHs presentados pueden utilizarse de manera combinada de acuerdo al aspecto que desee enfatizarse durante la inspección de la interfaz que será evaluada. Para optimizar el éxito de la aplicación de esta técnica de evaluación de la usabilidad, siempre ha de tenerse en cuenta que una definición adecuada de criterios heurísticos debe derivarse de un profundo conocimiento de las características propias del contexto que se desee evaluar.

Finalmente queremos enfatizar que una EH consta de diferentes pasos, siendo cada uno de ellos crucial para lograr un mayor éxito y una mayor efectividad. En este sentido, los recursos y el tiempo que se destinen a la Planificación de la HE se verán compensados por una Puesta en Marcha más controlada, efectiva y sustentable desde una postura científica. De la misma manera, un adecuado Procesamiento de los resultados obtenidos será fundamental para poder asegurar la validez y la relevancia de las conclusiones a las que se arribe tras haber evaluado un sistema interactivo a través de la metodología de la EH.

## Agradecimientos

Agradecemos al Dr Toni Granollers de la Universitat de Lleida por sus sugerencias para mejorar la calidad del presente capítulo.

## Referencias

- [CW02] Cockton G., Woolrych, A. Sale must end: Should discount methods be cleared off HCI's shelves?. En revista Interactions, núm. 9, vol. 5,

pág. 13-18, 2002.

- [DKA92] Desurvire H., Kondziela J., Atwood, M. What is Gained and Lost When Using Evaluation Methods Other Than Empirical Testing. En Proceedings del Congreso HCI 1992. En Monk, A., Diaper, D., and Harrison, M.D (eds.) Cambridge University Press, pág. 15-18, 1992.
- [DMJ04] Dumas J.S., Molich R., Jeffries R. Describing Usability Problems - Are We Sending the Right Message?. En Revista Interactions, July-August 2004, pag 24-29.
- [DR99] DUMAS J. S., REDISH J. C.. A Practical Guide to Usability Testing. Intellect Books, 1999.
- [GGL06] GONZÁLEZ M. P., GRANOLLERS T., LORÉS J.. A Hybrid Approach for Modelling Early Prototype Evaluation under User-Centred Design through Association Rules. XIII International Workshop on Design, Specification and Verification of Interactive System DSV-IS '06. A publicarse en revista LNCS, 2006 (en prensa).
- [GLPG06] GONZÁLEZ M. P., LORÉS J., PASCUAL A., Granollers A. Evaluación Heurística de Sitios Web Académicos Latinoamericanos dentro de la Iniciativa UsabAIPO. Remitido a Int. Conf. Interacción 2006, en proceso de evaluación.
- [GPL04] GRANOLLERS A., PERDRIX A., LORÉS J. Incorporación de Usuarios en la Evaluación de la Usabilidad por Recorrido Cognitivo. En Proceedings de Interacción 2004, Lleida, pág. 291-295.
- [ISO98] ISO Standards No. 9241-11: Guidance on usability. Geneva, Switzerland: ISO, 1998.
- [JMWU91] JEFFRIES R., MILLER J. R., WHARTON C., UYEDA K. User interface evaluation in the real world: a comparison of four techniques. En Proceedings de la SIGCHI Conference on Human Factors in Computing Systems CHI '91. ACM Press, pág. 119–124, 1991. (<http://doi.acm.org/10.1145/108844.108862>)
- [JEF94] Jeffries, R. Usability Problem Reports: Helping Evaluators Communicate Effectively with Developers. In Nielsen, J., and Mack, R.L. (Eds.), Usability Inspection Methods, John Wiley & Sons, pág. 273-294, 1994.
- [JD92] Jeffries R., Desurvire H. Usability testing vs. heuristic evaluation: Was there a contest?. En revista SIGCHI, num 24, vo. 4, pág. 39-41, 1992.
- [KR97] Kantner, L. and Rosenbaum, S. Usability Studies of WWW Sites: Heuristic Evaluation vs. Laboratory Testing. ACM, 1997. En Proc. del 15 th International Conference on Computer Documentation SIGDOC '97: Crossroads in Communication. Snowbird, UT. New York, NY: ACM Press, pág. 153-160, 1997.
- [LAI04] LAI-CHONG E. L., HVANNBERG E. T. Analysis of strategies for improving and estimating the effectiveness of heuristic evaluation. En Proceedings de NordiCHI '04. ACM Press, 1994.  
(<http://doi.acm.org/10.1145/1028014.1028051>)
- [LGP06] LORÉS J., GONZÁLEZ M. P., PASCUAL A. Iniciativa UsabAIPO. Primeros Resultados. En revista Interacción, vol 1. Asociación AIPO, 2006 (en prensa).
- [LGP05] LORÉS J., GONZÁLEZ M. P., PASCUAL A. Primera fase de análisis del Proyecto UsabAIPO. En Proceedings del VI Congreso Español de Interacción Persona Ordenador (INTERACCIÓN'2005), dentro del I Congreso Español de Informática CEDI'05, pág. 217-221, 2005.

- [MAY99] MAYHEW D. J. The Usability Engineering Lifecycle: a practitioner's Handbook for User Interface Design. ACM Press, 1999.
- [MD] Molic R., Ede M. R., Kaasgaard K., Karyukin B. Comparative usability evaluation. En Journal Behav. Inf. Tech., vol.23, num. 1. Taylor & Francis Inc., pág 65—74, 2004
- [MOL90] MOLICH R., NIELSEN J. Improving a human-computer dialogue. En Communications of the ACM, num33, vol. 3. ACM Press, pág 338-348, 1990.
- [MUL98] MULLER M. J., MATHESON L., PAGE C., GALLUP R. Methods & tools: participatory heuristic evaluation. En Proceedings de Interaction'98. ACM Press, 1998.  
(<http://doi.acm.org/10.1145/285213.285219>)
- [MW00] MARCUS A., WEST E. G. Crosscurrents: Cultural Dimensions and Global Web-User Interface Design. En ACM Interactions, vol VII, num. 4. ACM Press, 2000, pág. 32-46.
- [NITA02] NIESLEN J., TAHIR M. Usabilidad de páginas de inicio. Análisis de 50 sitios Web. Prentice Hall, 2002.
- [Nie99] Nielsen, J. Usability engineering at a discount. En G. Salvendy & M.J. Smith (Eds.), Designing and using human-computer interfaces and knowledge based systems (pp 394-401). Amsterdam, The Netherlands: Elsevier Science Publishers, B.V, 1999.
- [NIE94] NIELSEN J. Heuristic evaluation. In Nielsen, J., and Mack, R.L. (Eds.), Usability Inspection Methods, John Wiley & Sons, 1994.  
(<http://www.useit.com/jakob/inspectbook.html>)
- [NIE94a] NIELSEN J. Enhancing the explanatory power of usability heuristics. En Proceedings del ACM CHI'94 Conf. ACM Press, pág 152-158.
- [NIE93] NIELSEN J. Usability Engineering. John Wiley & Sons, 1993.
- [NIL93] NIELSEN J., LANDAUER T. K. A mathematical model of the finding of usability problems. En Proceedings de CHI'93. ACM Press, 1993.  
(<http://doi.acm.org/10.1145/169059.169166>)
- [NIE92] NIELSEN J. Finding usability problems through heuristic evaluation. En Proceedings de CHI '92. ACM Press, 1992.  
(<http://doi.acm.org/10.1145/142750.142834>)
- [NIE90] NIELSEN J, MOLICH R. Heuristic evaluation of user interfaces. En Proceedings of ACM CHI 1990. ACM Press, 1990, Pág. 249-256, 1990.
- [NIE89] Nielsen, J. Usability engineering at a discount. In G. Salvendy & M.J. Smith (Eds.), Designing and using human-computer interfaces and knowledge based systems (pág. 394-401). Elsevier Science Publishers, B.V, 1989.
- [Pie04] PIEROTTI D. Heuristic Evaluation - A System Checklist. Society for Technical Communication. © 1998-2004. Recuperado en junio 2006.  
(<http://www.stcsig.org/usability/topics/articles/he-checklist.html>)
- [SC99] Slavkovic A., Cross K. Novice Heuristics Evaluation of a Complex Inter-face. Publicado en Extended Abstracts of the ACM CHI 1999 Conference on Human Factors in Computing Systems, 1999.
- [UNET06] Usability Net: International standards for HCI and usability. Disponible en [http://www.usabilitynet.org/tools/r\\_international.htm](http://www.usabilitynet.org/tools/r_international.htm) (Consultada en Marzo de 2006).

[SHWP06] SHEN S. T., WOOLEY M., PRIOR S. Towards culture-centred design. En revista *Interacting with Computers* (en prensa), 2006, pág. 1-33.

[WRLP94] WHARTON C., RIEMAN J., LEWIS C., POLSON P. The cognitive walkthrough method: A practitioner's guide. En J. Nielsen & R. L. Mack (Eds.), *Usability Inspection Methods*, New York: John Wiley & Sons Inc., 1994, pág. 105-140.

# Bibliografía

ALVA M. E., MARTINEZ A. B., CUEVA J. M. Usabilidad: Medición a través de métodos y herramientas. En Proceedings de Interacción'03. Vigo, 2003.

GRANOLLERS A. MPlu+a. Una Metodología que integra la Ingeniería del Software, la Interacción Persona-Ordenador y la Accesibilidad en el contexto de equipos de desarrollo multidisciplinares. Tesis doctoral. Universidad de Lérida, 2004.

(<http://griho.udl.es/catala/equip/invest/granollers.html>)

HASSAN MONTERO Y., MARTIN FERNANDEZ F. J. Guía de Evaluación Heurística de Sitios Web. Publicado en NSU No Solo Usabilidad Magazine, versión digital online, 30 de marzo de 2003. Recuperado en junio 2006.

(<http://www.nosolousabilidad.com/articulos/heuristica.htm#identidad>)

MANCHON EDUARDO. Un Caso real: Evaluación Heurística de renfe.es. Recuperado de Gestipolis.com en junio de 2006.

(<http://www.gestipolis.com/canales5/ger/ainda/32.htm>)

MULLER J. M., McCLARD A. Validating an extension to participatory heuristic evaluation: quality of work and quality of work life. En Proceedings de CHI'95. ACM Press, 1995.

(<http://doi.acm.org/10.1145/223355.223457>)

PREECE J., ROGERS I., SHARP H. Interaction Design. Beyond Human Computer Interaction. Chapter 13: Ask Users and Experts. John Wiley & Sons, 2002, pág. 359-389.