

Desarrollo de Sistemas Interactivos

Raquel Hervás Ballesteros (raquelhb@fdi.ucm.es)

Baltasar Fernández Manjón (balta@fdi.ucm.es)

Agustín Vila Lorenzo (agvila01@ucm.es)

Virginia Francisco Gilmartín (virginia@fdi.ucm.es)

Curso 2023/2024

Contents

Prefacio	2
Tema 5: Evaluación	3
5.1 Tipos de evaluación	3
5.2 Evaluaciones heurísticas	7
5.3 Evaluaciones con usuarios	29
5.4 Otros métodos de evaluación	45
Anexo: Cuestionarios estandarizados: UTAUT	58
Anexo: Cuestionarios estandarizados: IBM	62

Prefacio

Estos son los apuntes de la asignatura *Desarrollo de Sistemas Interactivos*, escritos por los profesores Pablo Moreno Ger, Guillermo Jiménez Díaz y Antonio Sánchez Ruiz-Granados, del Departamento de Ingeniería del Software e Inteligencia Artificial, y revisados por Raquel Hervás Ballesteros.

Este material ha sido desarrollado a partir de distintas fuentes, destacando como referencia principal las notas de la asignatura *Human Computer Interaction* del Prof. Keith Andrews de la Universidad Tecnológica de Graz, el material de la asignatura *Human Computer Interaction* impartido por Scott Klemmer a través de Coursera y los libros *About Face 3: The Essentials of Interaction Design* de Alan Cooper e *Interaction Design. Beyond Human Computer Interaction* de Rogers, Sharp y Preece.

Tema 5: Evaluación

En un tema anterior hemos cubierto la metodología de diseño de principio a fin, incluyendo la fase final de refinamiento. Aún así, muchas de las etapas del Diseño Guiado por Objetivos necesitan evaluar diseños en distintos estados de evolución (especialmente las etapas de definición del framework y de refinamiento).

En este tema vamos a estudiar distintas técnicas para la evaluación de la usabilidad y aceptabilidad de las distintas tecnologías.

5.1 Tipos de evaluación

Las evaluaciones pueden tener distintos objetivos, presupuestos y necesidades. Antes de explorar las técnicas en detalle, vamos a revisar algunos conceptos generales sobre evaluación.

5.1.1 Evaluación sumativa y formativa

En cualquier ámbito, la evaluación es un proceso que se realiza para obtener información sobre la ejecución y resultados de un proceso. Estos procesos pueden ser, por ejemplo, procesos educativos, procesos de producción industrial, aplicaciones informáticas o, en nuestro caso, el diseño de una interfaz gráfica.

En general, distinguimos dos tipos de evaluación según el objetivo de cada evaluación.

- Evaluación formativa (o de diagnóstico)
- Evaluación sumativa (o de verificación)

Evaluación formativa

La evaluación formativa se realiza para analizar el propio proceso: sus carencias, sus puntos fuertes, los aspectos que necesitamos mejorar, etc.

En el contexto específico de las evaluaciones de usabilidad, se suelen hacer evaluaciones formativas de los diseños. Estas evaluaciones se suelen realizar en las etapas intermedias de un proyecto con el objetivo de *guiar el diseño* y de *encontrar problemas de usabilidad*.

Evaluación sumativa

La evaluación sumativa se realiza para valorar los resultados del proceso, no el proceso en sí. Su objetivo suele ser la verificación de si el proceso está obteniendo los resultados esperados.

En el contexto específico de las evaluaciones de usabilidad, las evaluaciones sumativas de los diseños se pueden realizar en las primeras etapas (para cuantificar qué diseño obtiene mejores resultados) y en ocasiones se hacen también hacia el final como última validación para garantizar que el sistema producido es realmente usable.

5.1.2 Evaluación heurística y con usuarios

Como ya adelantamos en el tema anterior, distinguimos dos tipos básicos de evaluación de usabilidad, que se emplean en distintos momentos del ciclo de diseño y con distintos objetivos.

- Evaluaciones heurísticas
- Evaluaciones con usuarios

Evaluaciones heurísticas

Las evaluaciones heurísticas son procesos de evaluación que pretenden identificar y cuantificar problemas de usabilidad siguiendo directrices relativamente universales (las *heurísticas*). Este enfoque emplea expertos en usabilidad (que pueden ser parte del equipo de desarrollo) y facilitan la identificación temprana de problemas de usabilidad.

Esto resulta beneficioso en las primeras etapas del proceso, donde un análisis directo de usabilidad ayuda a identificar de manera temprana malas decisiones de diseño, sin necesidad de incorporar a usuarios finales en el proceso.

También son parte del proceso de validación final, en aquellos casos en los que sea importante validar de forma cuantitativa la usabilidad de una interfaz.

Evaluaciones con usuarios

Las evaluaciones heurísticas, dado su carácter general, son un primer punto de apoyo, pero no sirven para identificar problemas aislados y no siempre cuantifican adecuadamente los detalles de la Experiencia de Usuario.

La forma más directa de entender cómo el usuario final va a interactuar con la interfaz es permitiendo a los usuarios trabajar directamente con el programa y observar hasta qué punto son capaces de realizar determinados conjuntos de tareas.

Aunque la información que se puede obtener a partir de evaluaciones con usuarios es mucho más detallada que la que se puede obtener a partir las evaluaciones heurísticas, su coste es mucho mayor, por lo que lo ideal es combinar ambos tipos de evaluaciones.

5.1.3 ¿Cuándo evaluar?

Estudios exploratorios

Los estudios exploratorios se realizan al comienzo del ciclo iterativo de diseño, cuando el producto todavía está siendo definido (por eso es un tipo de evaluación “formativa”). Habitualmente, ya habremos definido las primeras iteraciones de los frameworks y tendremos distintos diseños alternativos.

Objetivos

El propósito de este tipo de estudio suele ser comparativo, analizando distintas alternativas y tratando de obtener respuestas a preguntas de alto nivel:

- ¿El usuario es capaz de formular acciones?
- ¿Se entienden bien los flujos de ejecución?
- ¿Faltan grupos de controles importantes?
- ¿Funciona mejor el uso de textos o de gráficos?
- ¿Es fácil encontrar las funcionalidades y los datos que se necesitan?
- ¿Concuerda nuestro diseño con las expectativas generales de los usuarios?

La clave de estas evaluaciones preliminares, en las que todavía no se prueban controles concretos, es centrar el framework general de interacción y centrarse en un modelo general específico. Más adelante será posible hacer cambios en los controles específicos, pero cambiar completamente de modelo general requeriría empezar de nuevo. De ahí la importancia de una primera validación temprana.

Metodología

Típicamente estas evaluaciones se realizan para varios diseños alternativos en paralelo, usando prototipos de muy baja fidelidad (*mockups* en papel, *wireframes*, etc.).

En general, usaremos **prototipos horizontales** que muestren la organización general pero no implementen los procesos en detalle.

Típicamente se usan entrevistas interactivas, en lugar de proponer al usuario que explore libremente (la funcionalidad es reducida en esta etapa) y se cuenta con un moderador/entrevistador. Según el tipo de prototipo, puede ser necesario también un **operador** (para modificar el mockup en papel o para tareas de tipo *magos de Oz*).

Para añadir profundidad y averiguar más sobre la interacción, se pueden introducir preguntas como “*si tuvieses que hacer esta tarea, ¿dónde empezarías?*” o “*¿en qué menú esperarías encontrar la función X?*”.

Pruebas de evaluación

Las pruebas de evaluación son las más típicas. Se realizan sobre prototipos preliminares pero más funcionales, cuando el diseño general ya se ha poblado con los distintos controles y se pueden realizar tareas en profundidad (es decir, con **prototipos verticales**).

Objetivos

Las pruebas de evaluación se emplean para comprobar si el diseño cumple las expectativas del usuario, si concuerda con el modelo mental del usuario y si el diseño general es consistente. Ya no se busca la exploración horizontal inicial, sino que se centra en comprobar si el usuario es capaz de realizar tareas complejas de forma autónoma.

El resultado de este tipo de evaluación es, típicamente, una lista con los puntos de la interfaz en los que los usuarios se quedan atascados, cometen errores o encuentran inconsistencias.

Metodología

Al contrario que en las pruebas preliminares, se asume que:

- La interacción con el entrevistador será mínima y el trabajo del usuario será autónomo.
- El usuario debe encontrar la forma de ejecutar la tarea y accionarla, no simplemente indicar cómo lo haría.
- Es muy importante que el usuario explique en voz alta sus pensamientos, dudas y procesos mentales, ya que esto ayuda a identificar los puntos problemáticos y las brechas de ejecución y evaluación.
- Se anotarán datos *cuantitativos* (número de errores, tiempo empleado en completar las tareas, número de veces que el usuario no ha sabido continuar, etc.) pero lo más importante es la información *cualitativa* (dónde se producen los errores, cómo de graves son, qué ha podido producirlos, etc.).

Pruebas de verificación

Las pruebas de verificación se realizan hacia el final del proceso, habitualmente con un producto funcional detallado.

Objetivos

Se parte de la idea de que el diseño es válido para la mayoría de usuarios, y se intenta garantizar que cumpla una serie de métricas y estándares objetivos.

Los dos motivos habituales para una prueba de verificación son:

- Validar estadísticamente las apreciaciones sobre el número de errores aceptables.
- Comprobar si los últimos cambios han encontrado los errores detectados en una prueba de evaluación anterior.

Aunque siempre es deseable que el proceso dé como resultado sugerencias específicas de cambios, el propósito principal de este tipo de prueba es validar, no identificar mejoras.

Recordemos que en el proceso de Ingeniería de la usabilidad descrito anteriormente también podíamos fijar los objetivos de usabilidad que debía cumplir un sistema (sección

2.4.3) en términos como tiempos de respuesta del sistema, la estabilidad y la falta de errores. Estas evaluaciones también servirán precisamente para validar dichos objetivos.

Metodología

La metodología es similar a la de las pruebas de evaluación, con tres diferencias principales:

- Antes de la prueba se fijan las métricas y estándares que se espera que la aplicación cumpla.
- Los participantes reciben tareas y son observados, pero no se les pide que piensen en voz alta (esto podría modificar el resultado esperado, sobre todo si estamos midiendo los tiempos de respuesta del sistema).
- El objetivo principal es la recolección de datos *cuantitativos*.

Estudios a posteriori

Los estudios a posteriori se corresponden con las tareas de **seguimiento**. Se realizan una vez que el producto ya está en el mercado, típicamente centrándose en usuarios avanzados y no en usuarios principiantes.

Objetivos

Con la aplicación ya publicada, el objetivo suele estar relacionado con la planificación de nuevas iteraciones de las aplicaciones o con estudios de la competencia.

Metodología

Los procesos pueden ser presenciales, en cuyo caso se organizan como las pruebas de evaluación, pero con usuarios que ya conozcan la aplicación. También se pueden realizar estudios no presenciales, mediante el uso de informes de interacción que pueden ser generados por la herramienta y enviados a los diseñadores.

5.2 Evaluaciones heurísticas

El proceso de evaluación heurística de usabilidad fue propuesta inicialmente por Nielsen a partir del trabajo realizado por el propio Nielsen y Rolf Molich. Como ya se ha indicado al principio del tema, las evaluaciones heurísticas son procesos de evaluación que pretenden identificar y cuantificar problemas de usabilidad siguiendo directrices relativamente universales (las *heurísticas*). Este enfoque permite emplear expertos en usabilidad (que pueden ser parte del equipo de desarrollo) y facilitan la identificación temprana de problemas de usabilidad.

Aunque más adelante veremos distintos conjuntos de heurísticas, podemos poner como primer ejemplo las [10 heurísticas](#) de diseño revisadas propuestas por Nielsen:

1. Visibilidad del estado del sistema

2. Relación entre el sistema y el mundo real (Metáforas y lenguaje familiares)
3. Control y libertad del usuario
4. Consistencia y estándares
5. Prevención de errores
6. Reconocimiento mejor que recuerdo
7. Flexibilidad y eficiencia
8. Estética y diseño minimalista
9. Ayudar a los usuarios a reconocer, diagnosticar y recuperarse de errores
10. Ayuda y documentación

En las próximas secciones veremos la estructura general de una evaluación heurística y discutiremos más detalles sobre las posibles listas de heurísticas que se pueden emplear.

5.2.1 Proceso general para una evaluación heurística

Típicamente, las evaluaciones heurísticas siguen un proceso que contempla las siguientes fases:

1. Preparación previa
2. Evaluación
3. Informe de severidad (individual)
4. Puesta en común (debriefing)
5. Lista final de cambios y prioridades

1. Preparación previa

El primer paso en la preparación previa es identificar al conjunto de expertos que van a llevar a cabo la evaluación. Es recomendable que la evaluación la realicen varios expertos en paralelo, ya que los expertos no suelen identificar todos los posibles fallos de usabilidad. Al aumentar el número de expertos es probable que se detecten más fallos. En la Figura 5.1 se puede ver la matriz de evaluación de un caso real.

Al contrario que en las evaluaciones con usuarios (que se llevan a cabo con usuarios finales), las evaluaciones heurísticas se realizan con expertos en usabilidad. Esto supone dos diferencias importantes a la hora de preparar una evaluación:

- El experto en usabilidad posiblemente no conozca a fondo el campo de aplicación. Por tanto, se suele contemplar una sesión de entrenamiento en la que al experto se le informa sobre el campo de aplicación, los objetivos de los usuarios y las características de las personas. Incluso se le puede mostrar un escenario de uso con los pasos que el usuario realiza para que el evaluador comprenda cuál es el eventual modo de uso del sistema.
- Es posible (e incluso recomendable) repetir expertos entre dos evaluaciones. Aún así, sigue siendo recomendable tener algo de rotación para evitar vicios.

La siguiente fase en la preparación previa es decidir qué actividades va a llevar a cabo cada experto. En un sistema pequeño intentaremos que todos los expertos analicen el

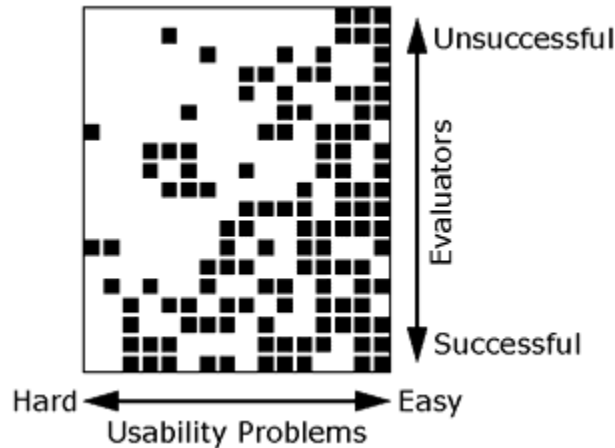


Figure 5.1: Matriz de evaluación de un caso real: las filas representan evaluadores y las columnas problemas específicos de usabilidad encontrados. Fuente: Nielsen Group

sistema completo. En un sistema grande, distintos expertos evaluarán distintos aspectos del sistema (pero siempre con solapamiento).

Cada experto recibe **un guión con las actividades** que debe llevar a cabo en su evaluación y **una lista de las heurísticas empleadas** en la evaluación. A continuación comienza el proceso de evaluación propiamente dicho.

2. Evaluación

Cada evaluador inspecciona la interfaz individualmente, tomando el tiempo que necesite para analizar todos los comportamientos, mensajes y diálogos que se producen durante la interacción hasta completar las tareas.

Aunque no existe una forma única de desarrollar la evaluación, sí se suelen seguir unas pautas generales comunes:

- Los evaluadores deberían evaluar las interfaces en solitario, sin hablar con otros evaluadores.
- Sí se contempla la posibilidad de que haya un miembro del equipo de trabajo presente (el “experimentador”), que ayude al evaluador y tome notas.
- Al contrario que en las evaluaciones con usuarios, es deseable y aceptable que el experimentador aclare conceptos al evaluador, especialmente las dudas relacionadas con el contexto y el campo de aplicación (en los que el evaluador no es experto).
- Dado que suelen ser iteraciones tempranas, es posible que el prototipo presente fallos, en cuyo caso también es aceptable que el experimentador ayude al evaluador.
- Es recomendable que el evaluador realice al menos dos recorridos por la interfaz: un primer recorrido de familiarización y un segundo recorrido más pausado tomando notas.
- Durante el segundo recorrido, el evaluador debe ir comparando los elementos de la interfaz con la lista específica de principios de usabilidad (las heurísticas).

El resultado final de la actividad debería ser una lista de los puntos problemáticos de la interfaz, con referencias a las heurísticas violadas en cada caso (según el criterio del evaluador), siempre de forma separada.

La lista de problemas es de pares (**elemento, heurística violada**). Así, si un cierto diálogo viola tres principios distintos, se deberían añadir tres entradas a la lista. Esto no impide que la solución a los tres problemas no pueda ser única (cambiar el diálogo). También se debe incluir para cada problema la información necesaria para observar dicho problema o, si es posible, una captura de pantalla que ayude a identificarlo e interpretarlo.

La lista la puede escribir el evaluador o la puede escribir el experimentador, en caso de querer aliviar la carga de trabajo del evaluador.

3. Informe de severidad (individual)

Después de completar la lista (no mientras se construye) el evaluador debe asignar a cada problema identificado una valoración de su severidad y su posible impacto sobre la experiencia de usuario.

Este proceso es relativamente corto y se suele hacer dentro de la misma sesión, a menudo en colaboración entre el evaluador y el experimentador.

4. Puesta en común (debriefing)

La evaluación heurística no ofrece de forma directa ninguna indicación de cómo solventar los problemas. Para transformar la lista de problemas y severidades en una lista concreta de cambios, se realiza una sesión de puesta en común en la que participan los evaluadores, los experimentadores y el resto del equipo de diseño.

El objetivo de la sesión es poner en común los problemas encontrados y discutir soluciones. Para ello, se suele comenzar con una primera pasada de puesta en común, en la que se agregan todas las observaciones y se agrupan por elementos de la interfaz. Para cada elemento de interfaz se discute el principio violado y la severidad reportada por cada evaluador (incluyendo a aquellos que no hubiesen identificado el problema como problema).

El resultado de esta puesta en común es una lista unificada de ternas (**elemento, heurística violada, severidad**). Aprovechando la presencia de los evaluadores, se pueden identificar posibles cambios en la interfaz que aliviarían el problema.

5. Lista final de cambios y prioridades

El último paso se suele realizar ya sin los evaluadores. Para cada problema, se propone un cambio específico, y se valida por parte de los diseñadores. Una vez decidido el cambio, se estima el coste de cambiarlo.

Finalmente, se combina la severidad del problema con el coste de resolución para atribuir una *prioridad* al cambio.

5.2.2 Conjuntos de heurísticas aplicables

En el paso 2 de la evaluación heurística se indica que el evaluador debe contar con una lista de principios. De acuerdo a ello, algunos elementos de interfaz se identificarán como problemáticos cuando violen alguno de estos principios. Por tanto, uno de los pasos iniciales por parte del equipo de trabajo es identificar dicho conjunto de heurísticas antes de comenzar la evaluación. Existen distintos conjuntos de principios propuestos por distintos autores, y también es habitual proponer conjuntos basados en la combinación de los principios propuestos por distintos autores. En esta asignatura presentaremos los siguientes:

5.2.3 Los 10 principios de diseño de Nielsen

Éstos son los principios de diseño propuestos por [Nielsen](#) para evaluar la calidad del sistema.

N01. Visibilidad del estado del sistema

El sistema siempre debería informar al usuario sobre lo que está pasando, aportando feedback adecuado con un retardo razonable. Ejemplos de interfaces que violan esta heurística se pueden ver en las Figuras 5.2 y 5.3, y que sí cumplen esta heurística en las Figuras 5.4 y 5.5.

N02. Relación entre el sistema y el mundo real (Metáforas y lenguaje familiares)

El sistema debe emplear un lenguaje próximo al usuario, con palabras, frases y conceptos que sean familiares para él. Se deben evitar términos internos del sistema, y hacer que la información aparezca en un orden lógico y natural. Ejemplos de interfaces que violan esta heurística se pueden ver en las Figuras 5.6 y 5.7, y que sí cumplen esta heurística en las Figuras 5.8 y 5.9.

N03. Control y libertad del usuario

Los usuarios a menudo activan controles por error que generan estados no deseados y necesitan la posibilidad de abandonar estos estados sin tener que realizar conjuntos de acciones complejos. Es recomendable que se pueda deshacer y rehacer. Un ejemplo de interfaz que viola esta heurística se puede ver en la Figura 5.10, y que sí cumple esta heurística en la Figura 5.11.

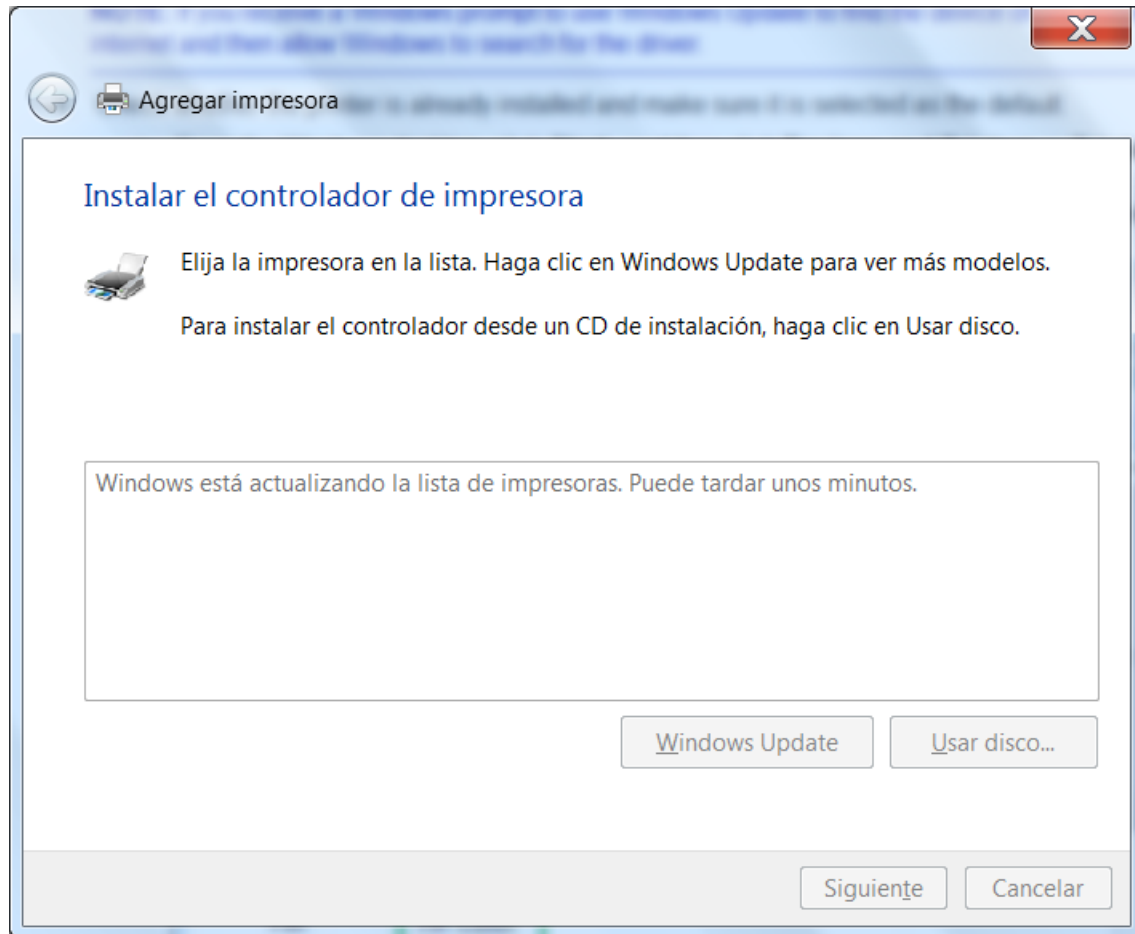


Figure 5.2: MAL (N01): Después de pulsar en Windows Update el sistema se queda tal y como aparece en la figura

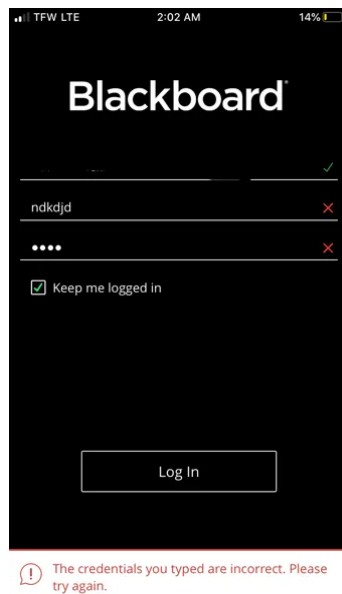


Figure 5.3: MAL (N01): No está claro si hemos puesto mal el usuario, la contraseña, o ambos (Fuente: Medium.com)



Figure 5.4: BIEN (N01): Estado del sistema en Gmail: cuánto espacio tengo disponible y cuándo fue mi último acceso

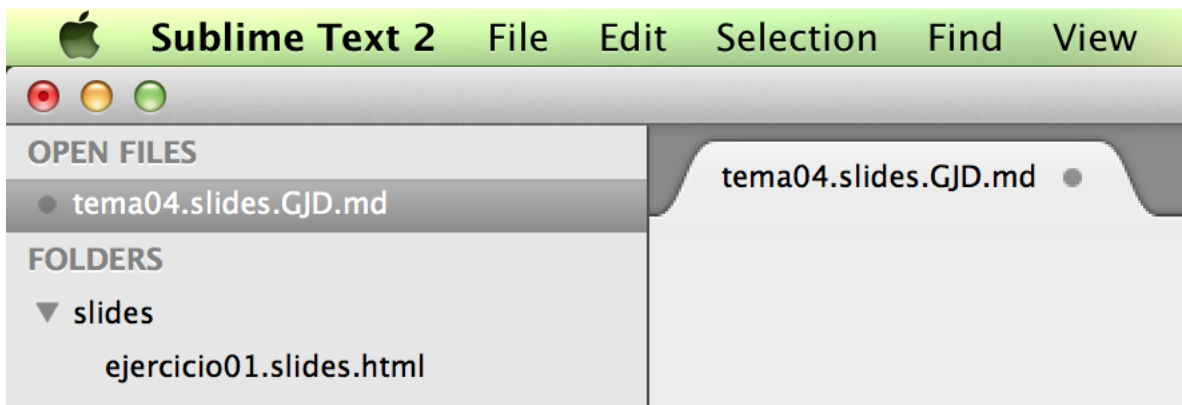


Figure 5.5: BIEN (N01): Estado del sistema en Sublime Text 2: el archivo ha sido modificado pero no está guardado en disco

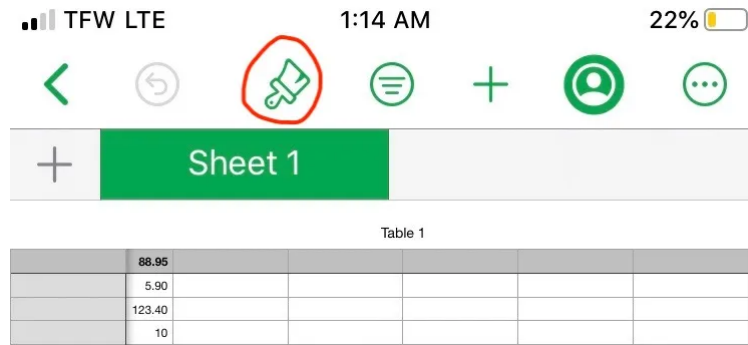


Figure 5.6: MAL (N02): El icono sugiere un pincel con el que pintar, pero lleva a las opciones de formato (Fuente: Medium.com)

The figure shows two screenshots of the GEA system interface for the Universidad Complutense de Madrid. Both screenshots show a navigation bar with options: CALIFICAR ACTA, PUBLICAR ACTA, CERRAR ACTA, IMPRIMIR ACTA, and ESTADÍSTICAS. The left screenshot shows the 'Estadísticas' screen for 'Grupo: Grupo C' with the following data:

Calificación	Porcentaje	Nº Estudiantes
NOTABLE	6.9 %	2
SUSPENSO	13.79 %	4
APROBADO	37.93 %	11
NO PRESENTADO	41.38 %	12
Totales	100.0 %	29

The right screenshot shows the 'Estadísticas' screen for 'Grupo: Grupo B' with the following data:

Calificación	Porcentaje	Nº Estudiantes
NOTABLE	13.33 %	6
APROBADO	46.67 %	21
SUSPENSO	13.33 %	6
NO PRESENTADO	26.67 %	12
Totales	100.0 %	45

Figure 5.7: MAL (N02): GEA, sistema de gestión de actas de la UCM: ¿Cómo se ordenan las filas de las estadísticas?

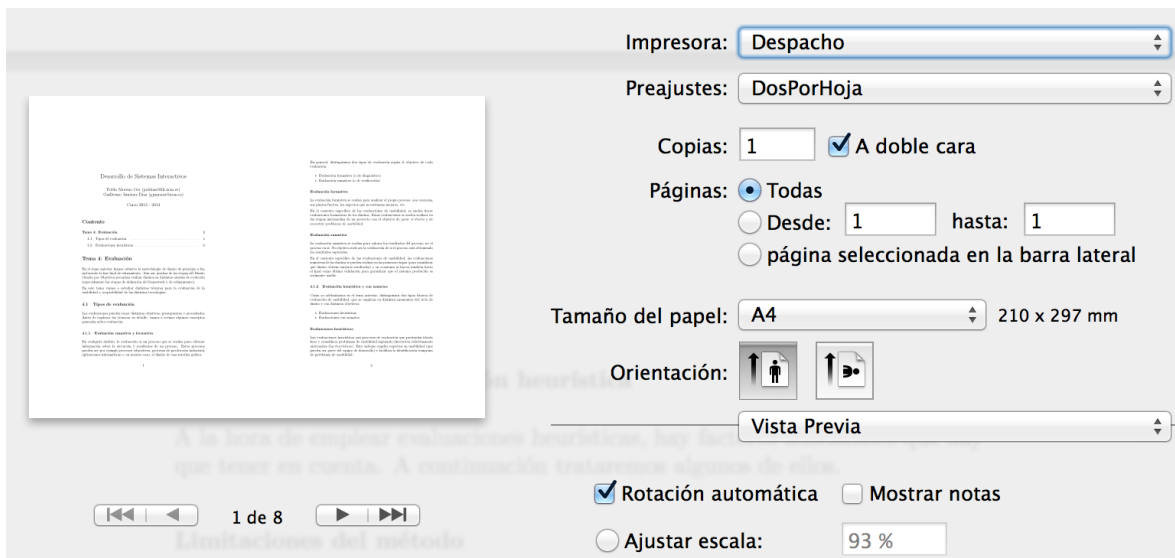


Figure 5.8: BIEN (N02): El servicio de impresión muestra cómo va a quedar el documento impreso

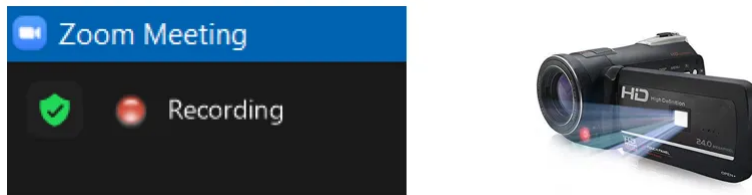


Figure 5.9: BIEN (N02): La metáfora del punto encendido viene de las cámaras de vídeo reales (Fuente: Medium.com)

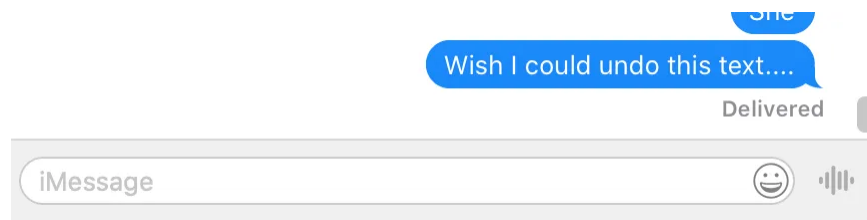


Figure 5.10: MAL (N03): En versiones antiguas de las aplicaciones de mensajería no era posible editar los mensajes enviados

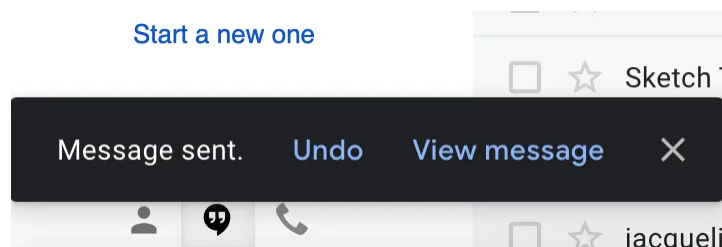


Figure 5.11: BIEN (N03): Deshacer envío de manera temporal

N04. Consistencia y estándares

Los usuarios nunca deberían verse obligados a preguntarse si palabras, situaciones o acciones distintas significan lo mismo. Para esto hay que seguir las convenciones de las plataformas y ofrecer experiencias consistentes. Un ejemplo de interfaz que viola esta heurística se puede ver en la Figura 5.12, y que sí cumple esta heurística en la Figura 5.13.



Figure 5.12: MAL (N04): Hay algo que parece un botón de configuración ... pero ni siquiera es un botón

N05. Prevención de errores

Mejor que escribir buenos mensajes de error es proponer un diseño adecuado que evite que el problema ocurra. Para esto, hay que eliminar las condiciones propensas a errores y, cuando no se puedan evitar, ofrecer vías fáciles para deshacer el error o solicitar confirmación antes de un cambio imposible de deshacer. Un ejemplo de interfaz que viola esta heurística se puede ver en la Figura 5.14, y que sí cumple esta heurística en la Figura 5.15.

N06. Reconocimiento mejor que recuerdo

Reducir la carga de memoria del usuario haciendo que los objetos, las acciones y las opciones sean visibles. Los usuarios no deberían verse obligados a recordar información entre una parte de un diálogo y otra. Las información sobre cómo proseguir debería estar siempre visible o fácilmente alcanzable. Un ejemplo de interfaz que viola esta heurística se puede ver en la Figura 5.16, y que sí cumple esta heurística en la Figura 5.17.

N07. Flexibilidad y eficiencia

Los aceleradores (por ejemplo, atajos de teclado) no son visibles para el usuario novato, pero mejoran la eficiencia del usuario experto, con lo que aumentan la flexibilidad del sistema. Es importante permitir a los usuarios acelerar o personalizar acciones frecuentes. Sublime, por ejemplo, permite hasta personalizar los atajos según lo que le venga mejor al usuario (Figura 5.18).

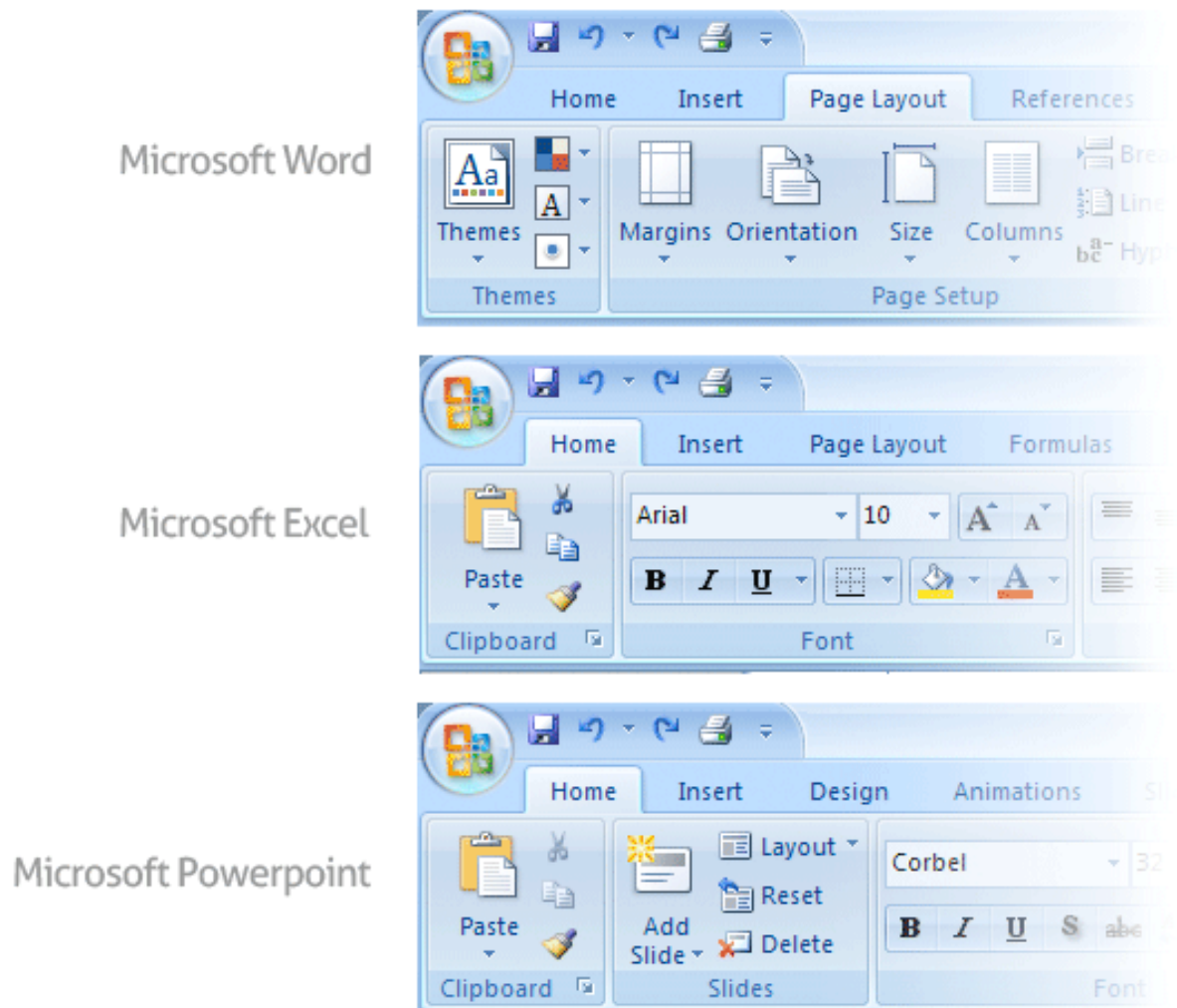


Figure 5.13: BIEN (N04): Consistencia en la suite de Office



MY AVIS Personal Data

To sign up for My Avis and the Wizard program at once, simply complete and submit the following personal profile information. Upon completion, you'll receive a Wizard Member Number and be registered for My Avis Service.
Please note that information marked with an asterisk (*) is optional.

Contact Information

Title:	<input type="text" value="None"/>
First Name:	<input type="text"/>
* Middle Initial:	<input type="text"/>
Last Name:	<input type="text"/>
* Company Name:	<input type="text"/>
Address 1:	<input type="text"/>
* Address 2:	<input type="text"/>
City:	<input type="text"/>
State:	<input type="text" value="Please Select One"/>
Zip / Postal Code (Please do not use any spaces or dashes):	<input type="text"/>
Country:	<input type="text" value="U S A"/>
Telephone Number:	<input type="text"/>

Figure 5.14: MAL (N05): AVIS (página antigua): ¿Qué tengo que rellenar obligatoriamente?

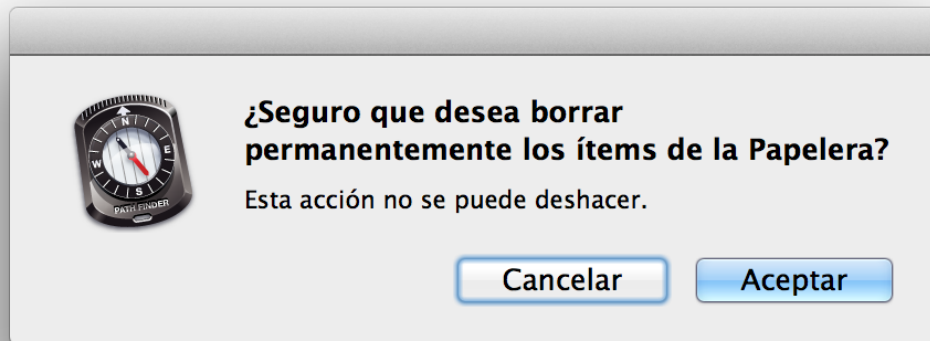


Figure 5.15: BIEN (N05): Papelera: ¿Estás seguro de que quieres borrar?

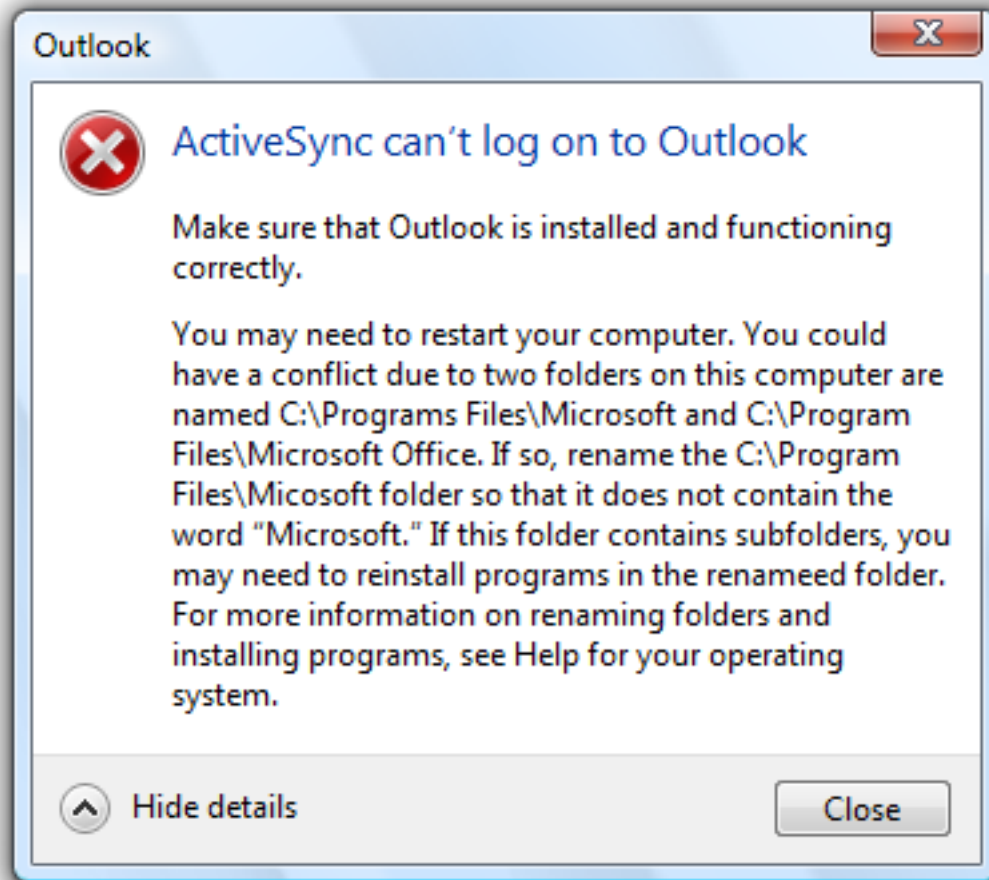


Figure 5.16: MAL (N06): ¿Qué era lo que tenía que renombrar?

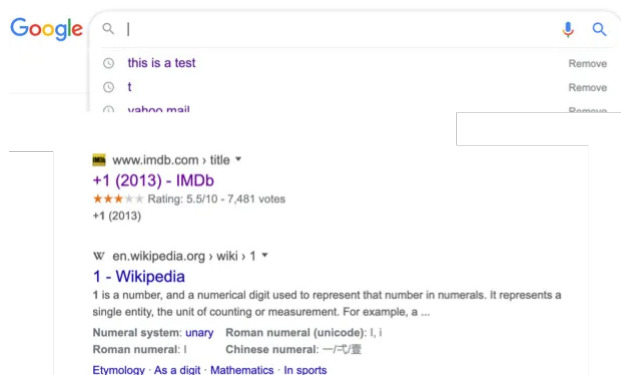


Figure 5.17: BIEN (N06): El buscador de Google recuerda búsquedas anteriores o frecuentes, y señala los sitios que ya se han visitado

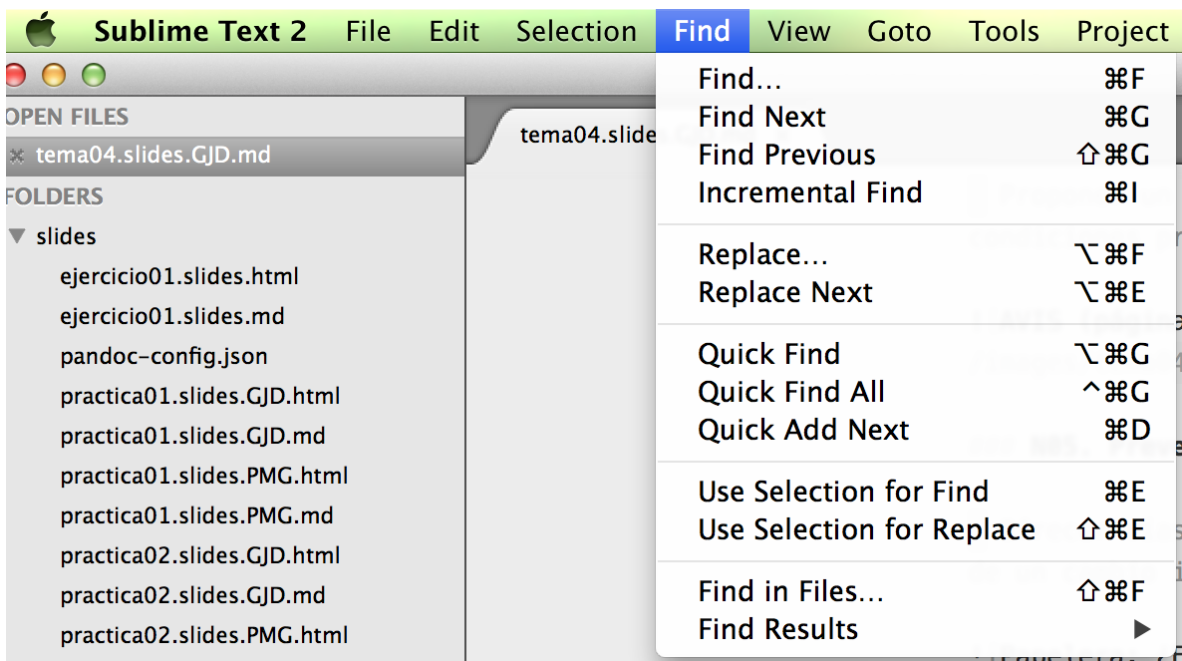


Figure 5.18: BIEN (N07): Sublime Text 2: Permite hasta configurar tus propios atajos

N08. Estética y diseño minimalista

Los diálogos no deberían incluir información irrelevante o poco necesaria. Cada pieza de información en un diálogo o pantalla compite con el resto de unidades de información relevantes y disminuye su visibilidad relativa. Recordad que *menos es más*. Un ejemplo de interfaz que viola esta heurística se puede ver en la Figura 5.19, y que sí cumple esta heurística en la Figura 5.20.

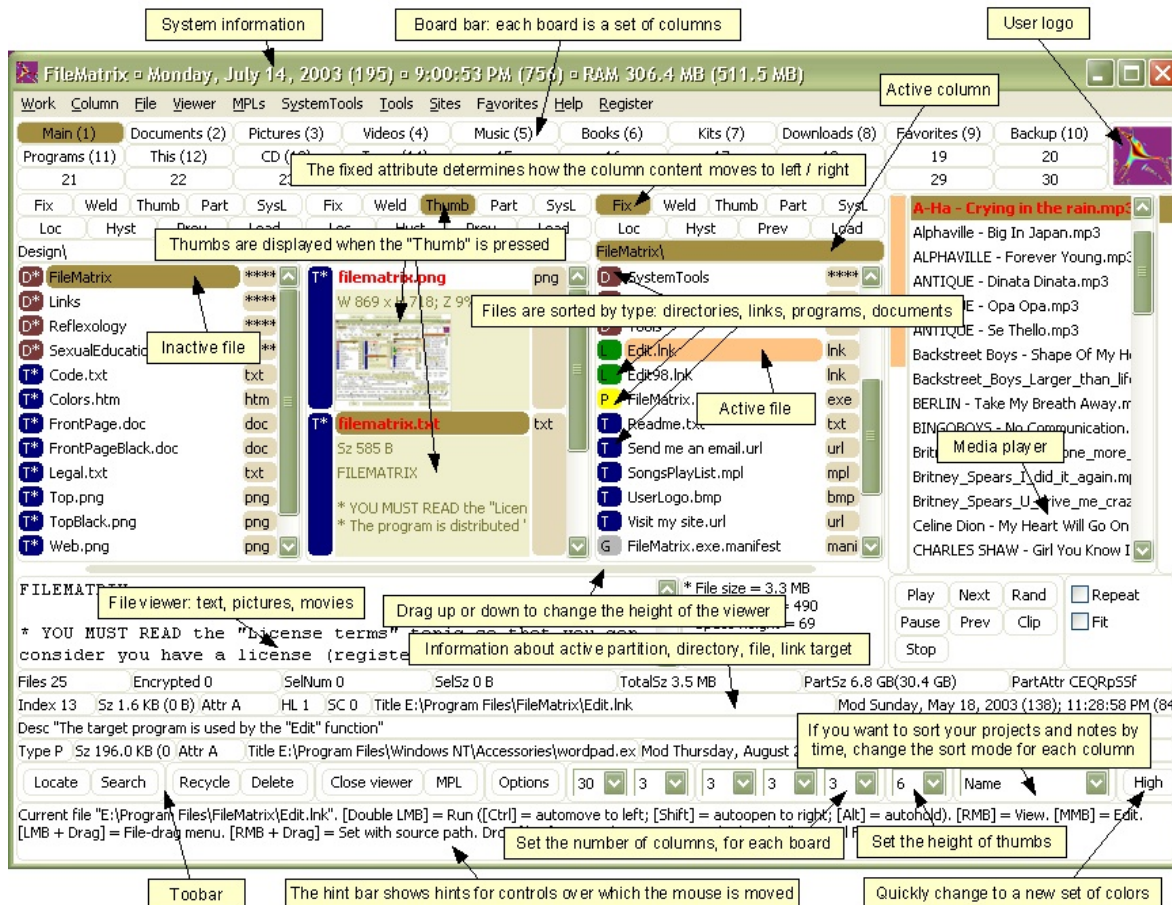


Figure 5.19: MAL (N08): FileMatrix: lo opuesto al minimalismo

N09. Ayudar al usuario a reconocer, diagnosticar y recuperarse de errores

Los mensajes de error deben explicarse en lenguaje sencillo, sin incluir códigos, deben indicar el problema de forma precisa y sugerir una solución de forma constructiva. Un ejemplo de interfaz que viola esta heurística se puede ver en la Figura 5.21, y que sí cumple esta heurística en la Figura 5.22.

N10. Ayuda y documentación

Aunque siempre es deseable que el sistema se pueda usar sin consultar documentación, puede ser necesario añadir ayudas y documentación adicional. Dicha información debería

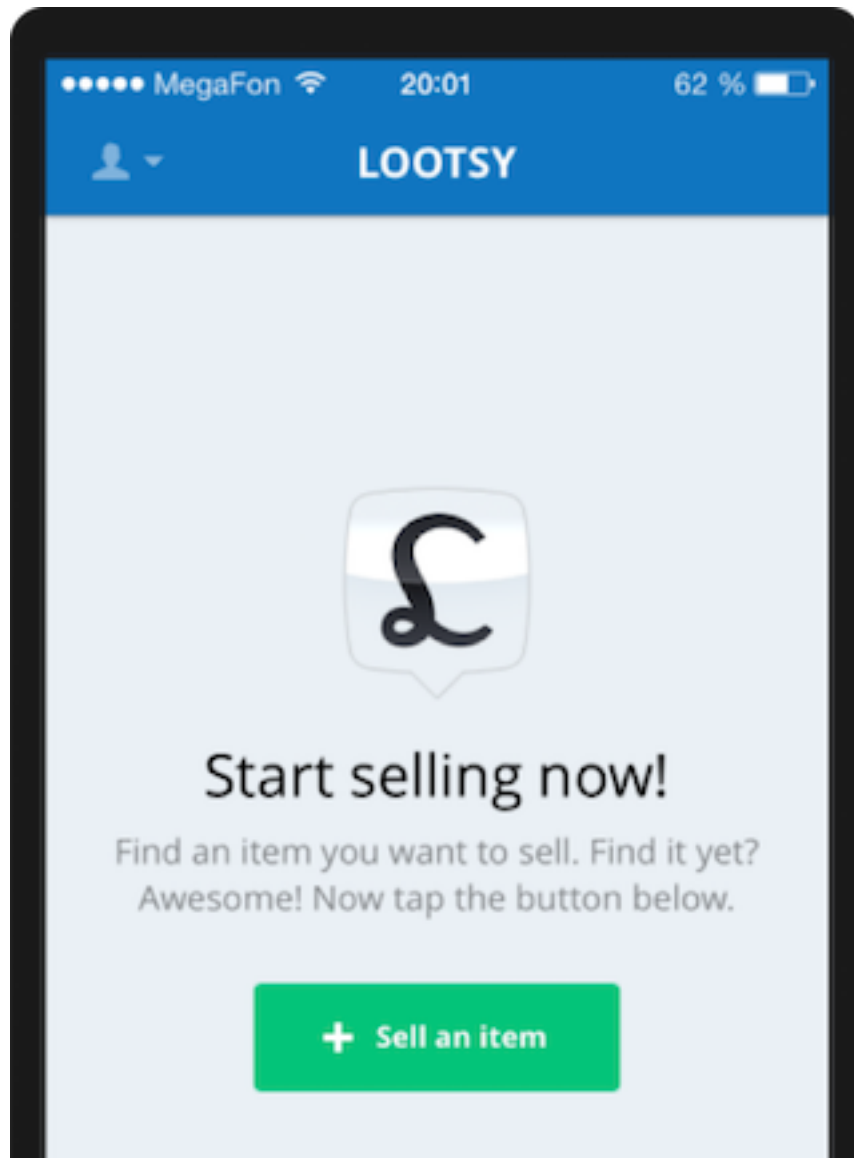


Figure 5.20: BIEN (N08): Lootsy me facilita qué puedo hacer si acabo de empezar a usarlo

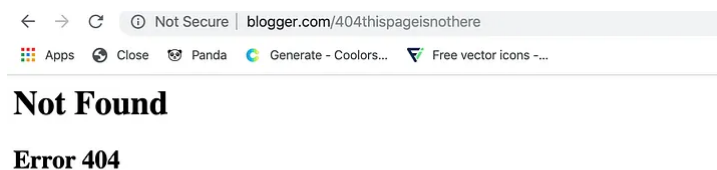


Figure 5.21: MAL (N09): No sabemos qué ha pasado ni cómo solucionarlo



Figure 5.22: BIEN (N09): Wordpress: no solo te indica el error, te da la oportunidad de solucionarlo

ser fácil de encontrar, permitir búsquedas, centrarse en las tareas del usuario, listar pasos específicos y ser breve. Un ejemplo de interfaz que viola esta heurística se puede ver en la Figura 5.23, y que sí cumplen esta heurística en las Figuras 5.24 y 5.25.

5.2.4 Las 8 reglas de oro de Shneiderman

Aunque los principios de Nielsen son los más comunes, otros autores han propuesto sus propios conjuntos de principios, y se siguen usando en la actualidad. Como muestra, Ben Shneiderman propuso [las 8 reglas de oro](#) del diseño, que son una alternativa a los principios de Nielsen:

S01. Buscar la consistencia

Las situaciones similares requieren secuencias de acciones consistentes. Se debería usar terminología idéntica en todos los diálogos, menús y pantallas de ayuda, y los comandos posibles en cada momento deberían ser consistentes.

S02. Proveer accesibilidad para todos

Proporcionar funcionalidades especiales para usuarios novatos como explicaciones. A medida que aumenta la frecuencia de uso, los usuarios desearán reducir el número de interacciones y aumentar su ritmo de interacción. Las abreviaturas, las teclas de función, los comandos ocultos y las macros son elementos útiles para usuarios expertos.

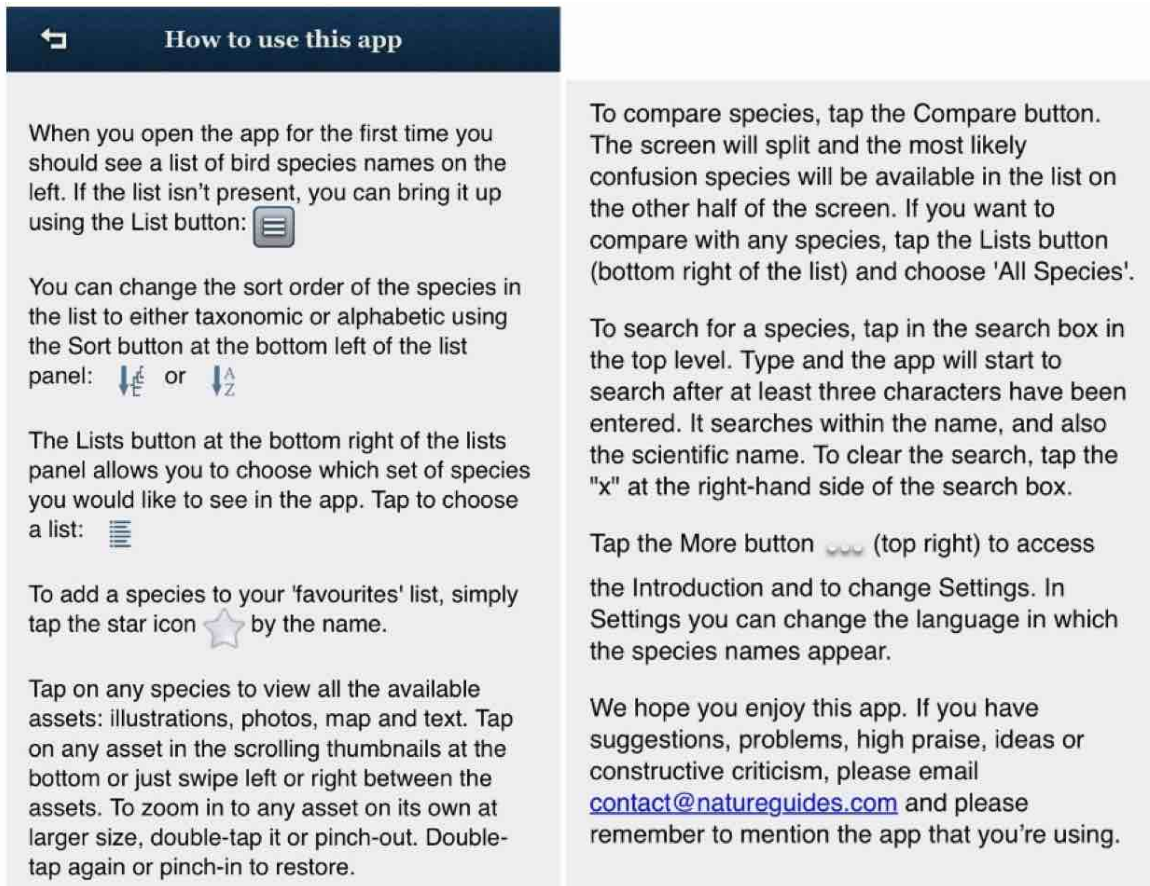


Figure 5.23: MAL (N10): La ayuda de esta aplicación es difícil de seguir y leer

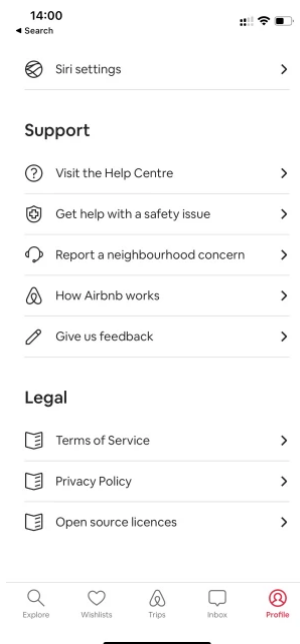


Figure 5.24: BIEN (N10): La ayuda no siempre será solo de las cosas que se pueden hacer en la aplicación



Figure 5.25: BIEN (N10): La ayuda de Slack está perfectamente integrada y su uso es muy natural

S03. Ofrecer feedback informativo

Para toda acción del usuario debería haber una respuesta del sistema. Para las acciones frecuentes y menores, la respuesta puede ser modesta, mientras que las acciones mayores o infrecuentes requieren respuestas consistentes.

S04. Diseñar el dialogo para aportar *cierre*

Las secuencias de acciones deberían tener un principio, una mitad y un final. El feedback informativo al completar un conjunto de acciones aporta al operador una sensación de logro y de alivio por completar la tarea. Esto señala también que las opciones y planes alternativos que pudiera tener el usuario en la cabeza ya no son necesarios, y que el sistema está listo para emprender el próximo conjunto de acciones.

S05. Ofrecer mecanismos simples de gestión de errores

Siempre que sea posible, diseñar los sistemas para que los usuarios no puedan cometer errores graves. Si se produce un error, el sistema debería ser capaz de detectarlo y ofrecer mecanismos simples y fáciles de entender para gestionar el error.

S06. Permitir deshacer acciones fácilmente

Esta regla reduce la ansiedad haciendo que el usuario sepa que sus errores tienen arreglo. A su vez, esto fomenta la exploración de opciones menos familiares. Las unidades reversibles pueden ser una acción individual, una entrada de datos o un grupo complejo de acciones.

S07. Permitir el Locus de Control

Los usuarios experimentados desean sentir que controlan el sistema y que el sistema responde a sus acciones. Hay que diseñar los sistemas para que el usuario sea quien inicie las acciones, no quien responda a los comandos del sistema¹.

S08. Reducir la carga de la memoria de trabajo

La limitación de capacidad de la memoria de trabajo requiere que las pantallas sean simples, que los conjuntos de pantallas se consoliden, que se reduzcan los cambios de ventana y que se dedique tiempo a aprender códigos y secuencias de acciones.

5.2.5 Más sobre evaluación heurística

A la hora de emplear evaluaciones heurísticas, hay factores adicionales que hay que tener en cuenta. A continuación trataremos algunos de ellos.

¹Wikipedia: El [Locus de Control](#) es la emoción de una persona sobre lo que determina (controla) el rumbo de su vida. Es el grado en que un sujeto percibe que el origen de eventos, conductas y de su propio comportamiento es interno o externo a él.

Limitaciones del método

La evaluación heurística a menudo se considera un método de ingeniería de la usabilidad “de saldo”: puede ofrecer resultados adecuados con un coste muy reducido, pero no da los mejores resultados posibles. En particular, el propio Nielsen afirma que *“la evaluación heurística no garantiza resultados perfectos, ni que se encuentren todos los problemas de usabilidad en una interfaz”*. Por tanto, las evaluaciones detalladas para proyectos de alto presupuesto tienden a preferir realizar evaluaciones con usuario, ya que ofrecen una perspectiva más detallada.

Por otro lado, muchos autores consideran que aunque los resultados no sean perfectos, el coste es bajo y ayuda a identificar problemas potenciales con un coste mucho menor que el de otros procesos de evaluación.

Número de evaluadores

A la hora de evaluar, la experiencia dice que un evaluador individual suele obtener resultados muy modestos. En un estudio de Nielsen de 6 evaluaciones independientes entre sí, la media de errores encontrados por cada revisor era cercana al 35%.

También observó que distintos evaluadores encontraban problemas distintos, y de ahí la deducción de que **vale la pena tener evaluadores redundantes**.

Cuanto más evaluadores participen, más problemas de usabilidad se encontrarán.

Por otro lado, la curva no es lineal (Figura 5.26). A partir de cierto punto, añadir más evaluadores aumenta relativamente poco el número de errores encontrados, mientras que el coste sigue creciendo linealmente a medida que añadimos evaluadores.

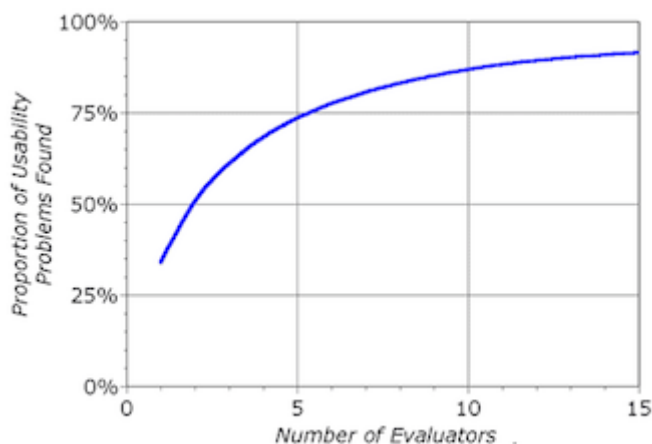


Figure 5.26: Número de evaluadores vs. problemas encontrados. Fuente: Nielsen Group

Asumiendo que cada error encontrado es un beneficio para el proyecto, podemos relacionar los costes y los beneficios de añadir nuevos evaluadores de acuerdo a la Figura 5.27.

Como se puede ver, aumentar excesivamente el número de evaluadores hace que se aumente el coste del proyecto sin que se compense con un gran beneficio.

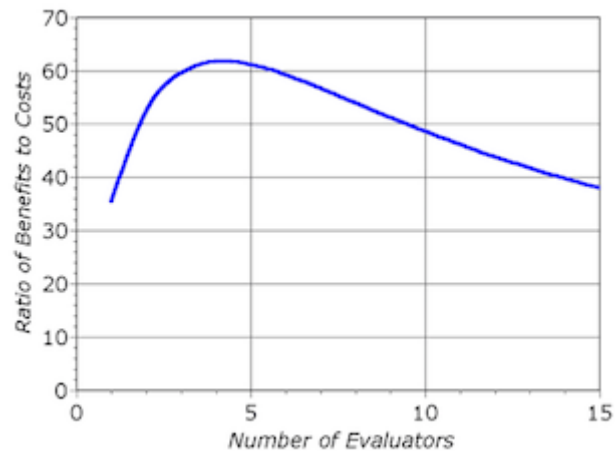


Figure 5.27: Número de evaluadores vs. ratio de beneficio. Fuente: Nielsen Group

Cuándo usar el método

Hay cuatro momentos importantes para realizar evaluaciones heurísticas:

Antes de empezar a hacer pruebas con usuarios

Las pruebas con usuarios son caras y los usuarios, difíciles de encontrar. La evaluación heurística sirve para detectar los problemas más evidentes y las pequeñas distracciones. Así, los usuarios se centrarán en completar la tarea e identificarán los problemas complicados.

Antes de un rediseño

Si hay problemas graves en las pruebas de usuario y éstos no consiguen una interacción fluida con el sistema, antes de cambiar el diseño conviene hacer evaluaciones heurísticas. Debido a la participación de expertos suele ser más fácil entender las causas de un problema que en una evaluación con usuarios.

Cuando sospechas que hay problemas, pero necesitas evidencias

En un desarrollo temprano, es posible encontrarse con la sensación de que la interfaz no va a funcionar bien. También es posible que en una aplicación ya lanzada los usuarios estén sintiéndose perdidos y no se sepa por qué. En estos casos, la evaluación heurística puede hacer un diagnóstico y justificar la inversión en un rediseño.

Justo antes de lanzar el producto

Después de todo el proceso, puede ser recomendable una última evaluación pausada para detectar problemas de última hora, sobre todo en relación con el uso del lenguaje y

en pequeños detalles de la interfaz.

5.3 Evaluaciones con usuarios

Las evaluaciones con usuarios simulan ejecuciones del sistema en entornos controlados por parte de usuarios finales. En general, una evaluación con usuarios suele desarrollarse según las siguientes etapas:

1. Preparar el *plan de evaluación*
2. Preparar el entorno de evaluación
3. Encontrar y seleccionar a los participantes
4. Preparar los materiales para la evaluación
5. Desarrollar las sesiones de evaluación
6. Debriefing con los participantes y los observadores
7. Analizar los datos y las observaciones
8. Informe de hallazgos y recomendaciones

Entre estas etapas, la más importante es la primera (la preparación del plan de evaluación), dado que es cuando se toman las decisiones relevantes sobre cómo se enfocarán el resto de las etapas.

5.3.1 Preparar el *plan de evaluación*

Debido al alto coste de realizar evaluaciones con usuarios, es muy importante tener un plan de evaluación específico que sirva como guía para realizar todo el proceso de evaluación.

Un buen plan de evaluación debe cubrir el *qué*, el *cómo*, el *cuándo* y, sobre todo, el *por qué*. También describe las tareas e instrumentos que se van a emplear en cada una de las etapas siguientes del proceso. Típicamente, la elaboración de un *plan de evaluación* incluye las siguientes subetapas:

1. Identificar el propósito y los objetivos de la evaluación
2. Formular las preguntas de investigación
3. Identificar los requisitos para los participantes
4. Describir el diseño experimental

Identificar el propósito y los objetivos de la evaluación

Según el propósito de nuestra evaluación, los elementos a medir pueden ser distintos. El plan de evaluación debe comenzar por describir **qué queremos evaluar y por qué queremos hacerlo**. Esta etapa sirve para asegurarse de que vale la pena lanzar el test y para tener una guía general de los objetivos de la evaluación. Suele formularse con conceptos de alto nivel que luego son refinados en la etapa siguiente.

Durante la etapa de diseño, el objetivo suele ser recabar información sobre un *hito* en el proceso de diseño:

- Se ha completado una iteración completa de gran tamaño en el proceso de diseño y hace falta realimentación sobre su adecuación.
- La interfaz se considera terminada y lista para empezar el desarrollo.
- Se ha completado el desarrollo y hace falta una comprobación final.

En cambio, con aplicaciones ya desplegadas suele venir definido por la identificación de posibles carencias:

- Se han recibido quejas mencionando que los usuarios encuentran problemas.
- Al hacer un estudio de logs de interacción se ha detectado que los usuarios dejan a medias ciertos procesos.

A partir de esta necesidad específica, se formulan **los objetivos de alto nivel**. Generalmente, los objetivos se pueden resumir en una sola frase. Sin embargo, si hay más de uno, se pueden escribir en forma de lista (corta) de objetivos.

Formular las preguntas de investigación

Una vez establecidos los objetivos del test, el siguiente paso es refinar dichos objetivos y formular **preguntas específicas de investigación**. Las preguntas de investigación son las preguntas que se espera que conteste la evaluación. No hay que confundirlas con las preguntas que queremos que respondan los usuarios.

Estas preguntas deben ser:

- Específicas (no genéricas ni abiertas)
- Precisas
- Medibles (u observables)

Malos ejemplos de preguntas de investigación:

- ¿La interfaz es usable?
- ¿El producto está listo para lanzarse al mercado?

Buenos ejemplos de preguntas de investigación:

- En una página web:
 - ¿Los usuarios saben qué elementos son accionables?
 - ¿Es fácil completar el proceso de registro?
 - ¿Cómo buscan los usuarios dentro de la página?
 - ¿Consiguen encontrar el contenido que buscan?
- Aplicación móvil
 - ¿Los usuarios cambian fácilmente de una pantalla a otra?
 - ¿Se entienden los iconos y los símbolos de la aplicación?
 - ¿Se pueden hacer las acciones importantes con pocas pulsaciones?
 - ¿Es cómodo accionar los controles?
- Aplicación empresarial

- ¿Los flujos de aplicación son fáciles de entender?
- ¿Los usuarios siempre encuentran los controles que necesitan?
- ¿Están bien reflejados los modelos mentales de los usuarios?
- ¿Se producen errores de interpretación?

Un buen plan de evaluación debería formular entre 5 y 10 preguntas de investigación y responder a todas ellas de forma cuantitativa o cualitativa.

Identificar los requisitos para los participantes

En el caso de procesos de diseño, los participantes se categorizan de acuerdo con las **personas** del Diseño Guiado por Objetivos: es importante que la población de usuarios elegida para las pruebas concuerde con las **personas** y que cubra los objetivos planteados al principio del diseño.

En cambio, en el caso de procesos de evaluación de aplicaciones existentes, los participantes deberían ser demográficamente representativos de la población de usuarios ya existente.

Describir el diseño experimental

El proceso de evaluación generalmente tendrá un **diseño experimental** que describa el contexto de la evaluación, el rol que va a tener el moderador y el análisis que se va a llevar a cabo de los datos observados, según las distintas opciones e instrumentos que veremos a continuación.

Describir la lista de tareas a realizar.

En los diseños experimentales **guiados** (*scripted tests*) es imprescindible que el plan de evaluación indique qué tareas se van a realizar durante la prueba. Estas tareas se elaborarán de acuerdo con una serie de directrices:

- Las tareas deberían cubrir todos los escenarios *key path* y numerosos escenarios de validación.
- Si la aplicación es muy grande, la lista de tareas para cada usuario puede ser distinta, enfocado de manera parcial a un subconjunto de funcionalidades del sistema.
- Todas las tareas deberían ser realizadas por más de un usuario.
- Si procede, el orden de las tareas debería alternarse.

Las tareas pueden ser descritas de dos formas distintas:

- Tareas directas: La tarea son unas instrucciones de lo que el usuario ha de realizar. Son más técnicas pero pueden perjudicar la experiencia de usuario. Ej. “Encuentra una receta con carne de cerdo en Food Networks”.
- Tareas escenario: Al igual que los escenarios, presentan al usuario objeto de la evaluación una tarea más cercana a como se la puede encontrar en la vida real. Ayudan a que el usuario no se sienta que está siendo evaluado pero hay que

tener cuidado para que sean realistas y no despisten al usuario del objetivo de la evaluación. Ej. “Este fin de semana tienes la cena de graduación. Busca un traje para este evento en El Corte Inglés”.

Esta descripción de tareas no es necesaria en caso de que el tipo de prueba sea **natural**, es decir, en el caso de que se desee que el usuario haga un uso libre del sistema con el fin de obtener unos datos más fidedignos de la interacción del usuario con el sistema a costa de sacrificar el control de la evaluación.

Describir el entorno y las herramientas que vamos a emplear.

El plan debe contemplar los aspectos logísticos de la evaluación.

Ejemplo: Las sesiones se realizarán en una sala de la Facultad de Informática, en la que estarán presentes el moderador y el usuario. El usuario dispondrá de un ordenador Windows en el que estarán instaladas la aplicación, un software de Webcam y un capturador de pantalla. La webcam captará la imagen del usuario y del moderador, y el capturador de pantalla capturará las acciones del usuario. También habrá una grabadora de audio para registrar la transcripción de forma redundante a modo de backup.

Especificar las tareas del moderador.

Típicamente el rol del moderador se centra en facilitar la sesión, pero es conveniente anotar previamente las “reglas de interacción”, especialmente si va a haber distintos moderadores:

- ¿Qué información le va a dar el moderador al usuario?
- ¿Debe el moderador ayudar al usuario si se atasca?
- ¿En qué condiciones es aceptable la intervención?

Además de dichas reglas, típicamente la función del moderador incluye tareas adicionales de observación y apoyo:

- Tomar nota de los eventos más relevantes
- Gestionar las herramientas, dispositivos de grabación, etc.
- Animar al usuario a que exprese sus pensamientos

Identificar los datos que se van a recolectar.

La observación por sí misma no suele ofrecer información suficiente, por lo que es necesario acordar qué datos se van a medir durante y después de la evaluación.

Los datos recogidos podrán clasificarse en dos tipos distintos de métricas: las *cuantitativas*, que son las que nos dan cifras de la evaluación (tiempo tardado, tasa de errores,...); y las *cualitativas*, que nos dan resultados subjetivos tanto del observador como del usuario (preguntas y test de satisfacción). Las más comunes son las siguientes:

- Tasa de éxito o número de tareas completadas sin ayuda.

- Porcentaje de tareas que no se han podido completar.
- Tasa de errores. Podemos hacer una distinción más fina entre tasa de acciones incorrectas (el usuario ha realizado una acción indebida o indeseada) y tasa de inacciones incorrectas (el usuario no ha activado un control que tenía disponible).
- Tiempo medio que se tarda en completar la tarea.
- Evaluación de la satisfacción del usuario, dificultad de la tarea, etc.
- Errores de interpretación de los mensajes escritos.

Descripción de la metodología de análisis de datos.

Los datos grabados necesitan ser revisados después de la evaluación. En el plan de evaluación se debe identificar la metodología de análisis que se va a seguir y cómo se van a convertir las observaciones y los datos en informes de evaluación.

5.3.2 Preparar el entorno de evaluación

El entorno físico donde se realiza la evaluación afecta mucho al desarrollo y al coste de la misma. A la hora de diseñar un entorno, existen diversos factores que puede ser necesario equilibrar:

- Las experiencias en el contexto final son más auténticas.
- Las experiencias en las instalaciones del desarrollador son más controlables
- Es deseable que mucha gente vea la interacción (como diseñadores, solemos no creernos los fallos hasta que los vemos).
- Si hay mucha gente, el usuario acaba confundido (miedo escénico).
- Cuanto más activamente se involucre el moderador, más cómodo se sentirá el usuario (pero la experiencia es menos auténtica).

A la hora de equilibrar estos factores, tendremos que tomar decisiones sobre los materiales a usar, la ubicación de las pruebas, el diseño de la habitación y las tecnologías concretas.

Podemos distinguir distintos escenarios en los que realizaremos las pruebas:

Pruebas en las instalaciones del usuario.

Como ya hemos mencionado, las pruebas en las instalaciones de los usuarios finales (e.g. en la empresa que ha contratado la aplicación) producen resultados más auténticos. Además, en el caso de desarrollos empresariales, pueden reducir el coste de reclutamiento de los usuarios al no obligarles a desplazarlos. Por otro lado, puede ser complicado encontrar y montar una sala para hacer las observaciones en buenas condiciones y es habitual encontrar interrupciones.

Pruebas en las instalaciones del desarrollador.

Las pruebas en las instalaciones del desarrollador son menos auténticas y resultan más caras, pero es más fácil controlar el tiempo y el espacio. En empresas de gran tamaño,

es fácil contar con salas específicamente diseñadas para hacer pruebas con usuarios, en un entorno controlado y debidamente equipado.

Aquellas empresas que son intensivas en pruebas con usuario suelen tener salas específicas para la realización de estas evaluaciones. Estas salas suelen disponer de:

- Un equipo para el usuario (ordenador o portátil)
- Instrumentos de grabación (interacción, usuario, moderador)
- Disposición de espacio para observadores (si procede)

Existen disposiciones de sala, cada una con sus ventajas e inconvenientes (Figura 5.28). Para más información sobre estas salas consultad *The Usability Handbook* en la bibliografía.

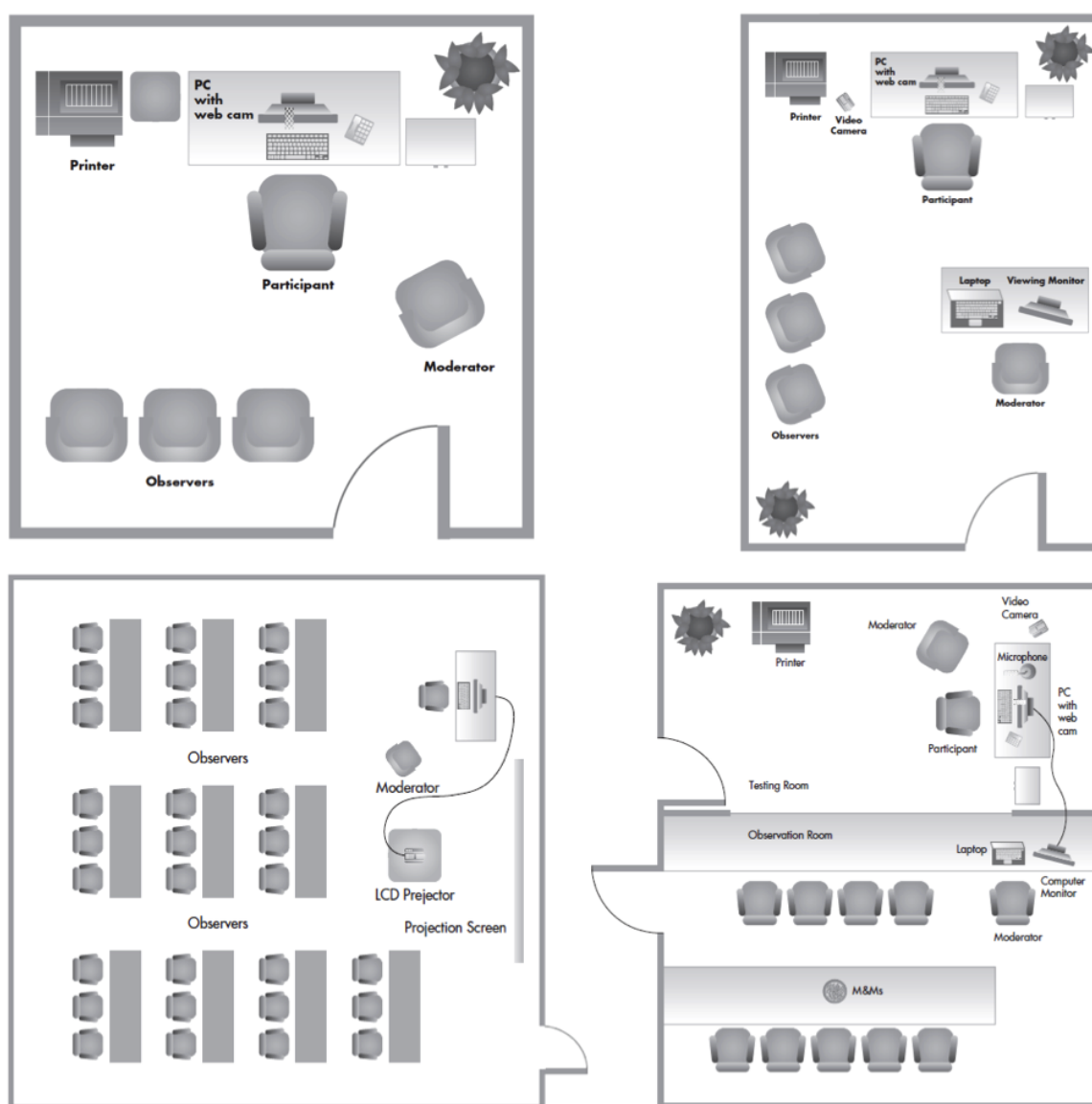


Figure 5.28: Disposiciones de salas. Fuente: The Usability Handbook

Laboratorio portátil minimalista.

Los diseños anteriores no son abordables por parte de la mayoría de los grupos de desarrollo, ni tampoco cuando las pruebas se desarrollan en las instalaciones del usuario. En pequeños desarrollos no suele ser necesario un proceso tan elaborado, por lo que en realidad, la mayoría de las pruebas se desarrollan con una versión minimalista consistente en un portátil que tenga:

- Una webcam
- Software para grabar con la webcam
- Software capturador de pantalla
- El prototipo a evaluar

OBS (para Windows, Mac y Linux, Figura 5.29) es una aplicación de software libre y de código abierto para grabación de video y transmisión en vivo.



Figure 5.29: Interfaz de OBS (Open Broadcaster Software)

Screenflow (para Mac, Figura 5.30) es un laboratorio de usabilidad en un sólo producto. Cuando se activa, graba a través de la webcam y del micrófono, y también graba todo lo que sucede en pantalla. Al terminar, produce un archivo que contiene el audio grabado, el vídeo de la webcam y el vídeo de la pantalla, y permite editarlos conjuntamente.

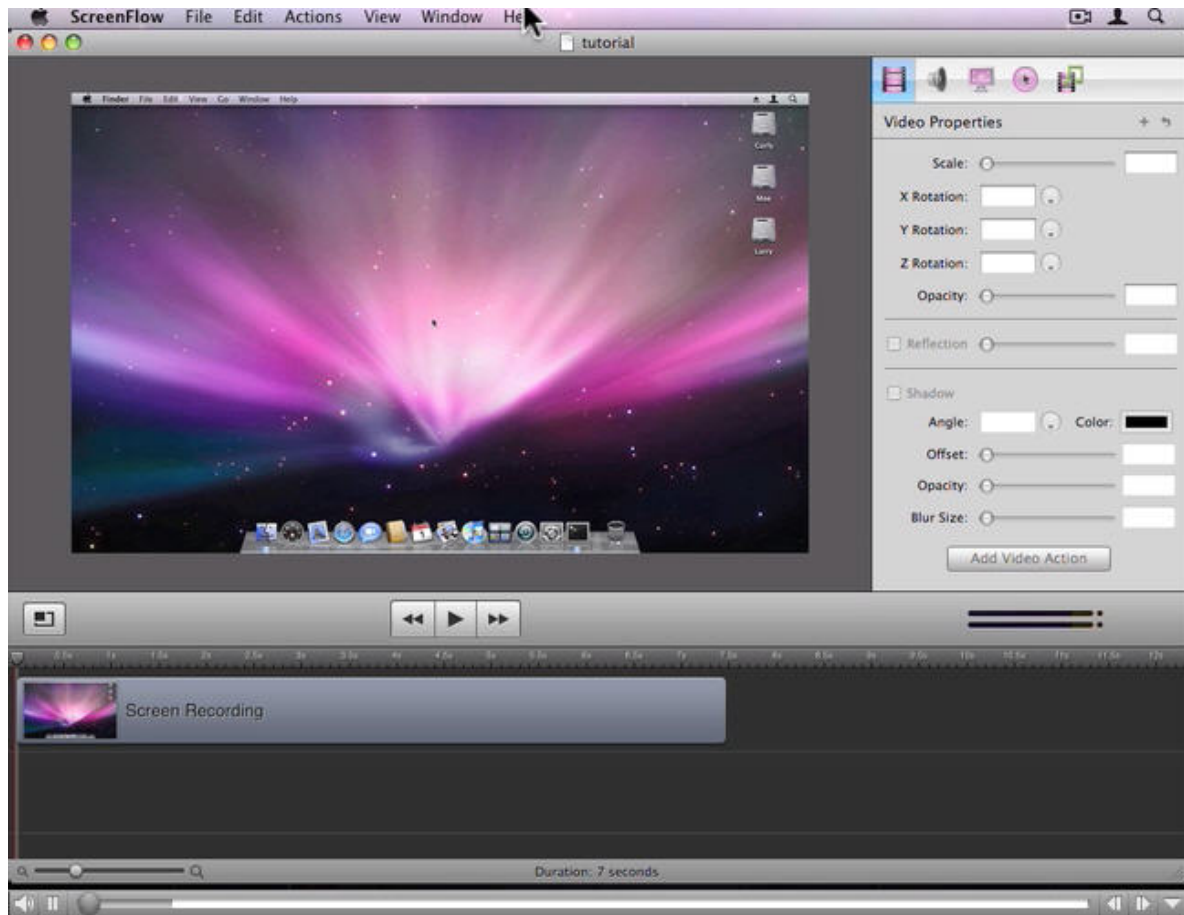


Figure 5.30: Interfaz de ScreenFlow

Pruebas de usabilidad remotas.

En estas pruebas el participante y el moderador se encuentran físicamente en lugares distintos, debido principalmente a querer reducir el coste de las pruebas o a problemas de tiempo. En este caso, el participante realiza las pruebas en su propio equipo y se le ha de proporcionar el software necesario para que la prueba pueda ser grabada para, posteriormente, ser enviada al moderador.

En este caso, podemos diferenciar dos tipos de pruebas remotas:

- Pruebas moderadas remotamente: el participante y el moderador están en el mismo espacio *virtual*, es decir, durante la prueba el moderador puede observar al usuario y tiene comunicación con él a través de chat, comunicación por voz, etc. El usuario se siente más seguro durante la realización de la prueba pero el moderador puede no saber cuándo tener que preguntar o cuándo escuchar al usuario, ya que se pierde la comunicación no verbal que hay en las evaluaciones en las que ambos están en el mismo lugar.
- Pruebas remotas no moderadas. El usuario está solo, graba la sesión y se la envía al evaluador para que la analice. El sistema debe recolectar toda la información necesaria, lo que puede obligar a incluir preguntas o cuestionarios que nos ayuden a conseguir un mayor feedback de la evaluación. La principal desventaja es que no se puede preguntar nada al usuario, por lo que puede ocurrir cuál es el motivo por el que un usuario ha terminado una sesión sin haber completado la tarea, o por qué se han producido ciertos errores no esperados. Este tipo de evaluación no se recomienda para la evaluación global del sistema sino más bien para evaluar elementos muy específicos.

5.3.3 Encontrar y seleccionar a los participantes

El siguiente paso en la ejecución del plan de diseño es la identificación y reclutamiento de los participantes. A la hora de buscar usuarios, es importante definir cuántos y qué tipos de usuarios necesitamos.

Identificar los usuarios adecuados

La aplicación siempre se debería probar con usuarios representativos del público objetivo. Dado que estas pruebas suelen realizarse con el proceso de Diseño Guiado por Objetivos ya avanzado, con lo que el desarrollo contará con un conjunto bien definido de *personas*.

Los usuarios escogidos para las pruebas deberían ser equiparables a las *personas*, y si hay *personas* muy distintas, necesitaremos usuarios muy distintos.

Resumen de la relación entre personas y usuarios: Como ya vimos en el tema anterior, el primer paso es la formulación de la *hipótesis de persona*. A partir de esta hipótesis se reclutan los usuarios para la primera etapa de investigación. A partir de esta investigación, se formulan las *personas* y se desarrolla el diseño. Finalmente, a partir de las *personas* se identifican los

usuarios para las pruebas de evaluación.

Determinar el número de participantes

En general, el número de participantes a reclutar depende de:

- El nivel de confianza en los resultados de la evaluación que queremos alcanzar.
- Los recursos humanos y técnicos disponibles para la evaluación.
- La disponibilidad de posibles participantes.
- La duración de la prueba de evaluación.

En general se busca siempre encontrar un equilibrio, dado que el coste de cada evaluación es muy elevado.

Screening

Es recomendable realizar un cuestionario previo para asegurarnos de la conveniencia de los usuarios elegidos. Típicamente queremos que los participantes nos informen sobre:

- Su nivel de formación
- Datos demográficos
- Su experiencia con aplicaciones similares
- Su nivel de conocimientos de informática de usuario

Gracias a estos cuestionarios, podremos conocer mejor a los usuarios que han realizado las evaluaciones, lo que a su vez permite:

- Asegurarnos de que el grupo de usuarios es suficientemente representativo del grupo de personas.
- Entender mejor las motivaciones de los usuarios al trabajar con el programa.
- Durante el procesamiento de los datos, conocer mejor el nivel del usuario para facilitar el diagnóstico de los problemas.

5.3.4 Preparar los materiales para la evaluación

La clave del éxito en una prueba con usuarios es la preparación previa. Antes de comenzar las pruebas es recomendable tener preparados todos los materiales y documentos necesarios para desarrollar la prueba.

Algunos elementos típicos que conviene tener preparados serían:

Guión para la orientación previa.

Antes de que el usuario empiece a interactuar con el sistema es común explicarle el contexto general, describir en qué va a consistir la prueba y consejos generales para el desarrollo de la misma. Habitualmente es un texto que el moderador lee en voz alta o un documento que se entrega a los participantes antes de comenzar.

Cuestionario previo.

Este cuestionario sería de naturaleza similar al cuestionario de *Screening* mencionado en la sección anterior (y a menudo es el mismo, ya que se hace una única vez al principio de la prueba). El objetivo es entender mejor al usuario y asegurarnos de que cumple los requisitos de la prueba. Típicamente se usan cuestionarios en papel o cuestionarios electrónicos (e.g. [Google Forms](#)).

Herramientas para recoger datos.

Antes de la sesión hay que tener preparados los recursos necesarios para capturar la información de cómo se desarrolla la sesión. Además de medios técnicos de grabación (vídeo, audio, etc.) es recomendable tener material para anotar las incidencias o momentos especialmente significativos.

Prototipos a evaluar.

Obviamente, es fundamental tener preparada una versión estable y lista para ejecutar de los prototipos que queremos evaluar.

Tareas y escenarios.

Es un documento que indica las tareas que debe realizar el usuario (de acuerdo con el plan de evaluación). Según el tipo de evaluación, puede ser un documento resumen para el usuario o un guión para que el moderador vaya indicando qué tareas debe hacer el usuario.

Cuestionario y/o entrevista final.

Al finalizar la prueba, es importante que el usuario dedique unos minutos más a indicar su grado de satisfacción con la aplicación, los problemas que ha encontrado o posibles sugerencias. Suele emplearse un cuestionario (en papel u online) o una entrevista estructurada con preguntas predefinidas (o ambas cosas). En ambos casos, es importante tener preparado el cuestionario y/o el guión de la entrevista final.

5.3.5 Desarrollar las sesiones de evaluación

Durante la sesión, el usuario debe ir realizando sus tareas de acuerdo con el plan de evaluación mientras que el moderador observa y facilita la sesión. Para conseguir los mejores resultados, es recomendable seguir ciertas buenas prácticas descritas a continuación.

Técnica de *Think-aloud*

La técnica de *think-aloud* consiste en pedir a los participantes que hablen durante la prueba, explicando qué están pensando durante la prueba.

Los comentarios deberían centrarse en expresar cuando se sienten confundidos, frustrados o incluso gratamente sorprendidos.

El uso de esta técnica tiene numerosas ventajas:

- Se recibe realimentación sobre lo que está pensando el usuario, lo cual ayuda a entender su modelo mental.
- Se detectan las posibles brechas de ejecución y de evaluación, al tener información sobre las intenciones del usuario (y contrastarlas con las acciones realizadas) y sobre sus expectativas (lo que espera encontrar, lo que encuentra y lo que interpreta).
- Ayuda a que el usuario se centre en la tarea y reflexione sobre las acciones que va a realizar.

Por otro lado, también presenta desventajas:

- Para algunos participantes resulta extraño ir hablando mientras realizan la tarea y les hace sentirse más observados.
- La técnica induce al usuario a pensar y realizar sus acciones más despacio (el ritmo verbal es más lento). Esto hace que se cometan menos errores.
- Si la prueba es larga, resulta fatigoso para el usuario.

Por último, cuando se usa esta técnica existen algunas recomendaciones a tener en cuenta:

- Es conveniente explicar bien la técnica a los usuarios, para que entiendan el tipo de realimentación que buscamos. La mejor forma sería ofrecer un ejemplo en el que el moderador interactúe brevemente con una herramienta distinta a la de la prueba.
- Algunos usuarios tienden a quedarse callados tras unos minutos. Se les puede recordar amablemente que se acuerden de verbalizar sus pensamientos.
- Los puntos que hacen que los usuarios se queden callados son especialmente relevantes, pues pueden indicar un aumento en la concentración.
- Es conveniente que el moderador cuide la función fática del lenguaje², ya que esto ayuda a que el usuario sienta que está en una conversación en lugar de estar siendo observado.
- Usar los silencios: en ocasiones ayuda dejar un silencio ante una pregunta del usuario o cuando encuentre un obstáculo: es probable que el usuario intente llenar el hueco con información adicional.
- En ocasiones existen usuarios con los que no es posible interactuar: o bien presentan mucha resistencia a hablar en voz alta o, aunque no se quejen, se quedan continuamente callados. En estos casos es preferible no insistir y sacar la información que se pueda de la prueba a presionar al usuario y no obtener ninguna información.

²Las unidades que usamos para iniciar, mantener y finalizar una conversación (Por supuesto, claro, naturalmente, entiendo, cómo no, OK, perfecto, bien, ya, de acuerdo, etc.). Ver [Funciones del lenguaje](#) en Wikipedia.

Actitud del moderador

La clave del éxito de la prueba es la actitud del moderador, que tiene que animar a los usuarios a explorar, a que no se sientan observados (¡aunque lo están siendo!) y debe facilitar la sesión sin guiar al usuario.

Para esto, hay distintas buenas prácticas que ayudan a obtener mejores resultados:

- No mostrar sorpresa (ni positiva ni negativa), ya que en el primer caso se contamina la prueba y en el segundo se genera una relación hostil con el usuario.
- Centrarse en averiguar las expectativas del usuario: si sucede un problema o el usuario se atasca, lo importante no es saber por qué no ha encontrado un botón, sino qué esperaba encontrar realmente.
- No hacer preguntas directas ni muy específicas, ya que ponen al usuario a la defensiva. No es buena idea preguntar (por ejemplo) “¿Por qué has pulsado ahí?”, sería preferible usar preguntas de tipo “¿Cómo vas?” o “Me he perdido la última acción, ¿puedes explicármela?”.
- Mantener un tono relajado: no pasa nada por crear una atmósfera tranquila, ni hay que ponerse a la defensiva sobre la aplicación si el usuario la critica.
- No buscar soluciones: si hay problemas, no es el momento de intentar discutir posibles soluciones. Es mejor dejar esto para la sesión de *debriefing* o para el equipo de diseñadores.

Cuándo intervenir

En general, el moderador no debería intervenir nunca para ayudar a un usuario, ya que es importante saber si el usuario sería capaz de encontrar la forma de hacer las cosas por sí mismo.

Esta política de no intervención resulta complicada ya que, cuando un usuario se atasca, los moderadores sienten el impulso de ayudar al usuario. Esto es normal, ya que es la reacción natural al ver a una persona pasar un mal rato.

Aún así, existen momentos en los que sí que resulta necesario intervenir:

- Si el usuario está irremediablemente perdido o confundido, ha pasado un tiempo prolongado y no hay opción de que consiga seguir con la prueba. En este caso, la prueba queda contaminada y la tarea se debe marcar como “no completada”.
- Si el usuario se encuentra muy frustrado y puede que abandone. En este caso es preferible ayudar al usuario, permitir que continúe con la prueba y anotar el evento para preguntar al usuario en la sesión de *debriefing*.
- Si falta información debido a que la aplicación no está terminada. En este caso es aceptable intervenir para explicar al usuario qué información falta y cómo se representaría.
- En caso de un *bug* o error de programa que requiera una intervención para seguir funcionando.

5.3.6 Debriefing con los participantes y los observadores

Al final de la sesión (normalmente tras un breve descanso) es el momento de obtener información adicional sobre las percepciones de los usuarios. Los dos instrumentos principales para esto son los cuestionarios y las entrevistas estructuradas (lo ideal es una combinación de ambas).

Cuestionarios de evaluación

La principal ventaja de emplear cuestionarios de evaluación es que son una posible fuente de información cuantitativa. Esto permite por ejemplo ver si la percepción de los usuarios mejora de una versión a otra.

Aunque no existe un formato único, lo más común es preguntar sobre las distintas características de las herramientas empleando preguntas de escala numérica (también conocidas como [escalas de Likert](#)), en las que el usuario expresa su grado de concordancia con expresiones enunciativas.

Ejemplo: Indique su grado de acuerdo en una escala de 1 a 5 (donde 1 es completamente en desacuerdo y 5 completamente de acuerdo) con las siguientes frases:

- *La aplicación me ha resultado fácil de usar*
- *He conseguido completar la tarea X sin esfuerzo*
- *Los mensajes de la aplicación eran claros en todo momento*

Para cada apartado del cuestionario se debería ofrecer un campo de texto **opcional** para que el usuario añada comentarios sobre su valoración.

Es recomendable también que el cuestionario empiece por preguntas más generales, y que tenga las preguntas más específicas después. Finalmente, se puede ofrecer un último campo general para recomendaciones, opiniones o críticas.

Hay un tipo de cuestionario especial llamado [SEQ \(Single Ease Question\)](#) y que se suele utilizar después de que el usuario complete cada tarea realizada durante la evaluación. Consiste en una única pregunta con el siguiente enunciado: “¿Cómo de fácil o de difícil has encontrado la tarea?”. La respuesta se suele dar en una escala Likert con 7 elementos (donde el 1 representa “Muy difícil” y el 7, “Muy fácil”). A pesar de su simplicidad, suele dar unos resultados muy buenos ya que nos ayuda a comprender otras métricas correlacionadas. Como veremos más adelante, es recomendable que, además de la pregunta, demos al usuario la posibilidad de que explique su respuesta.

En la próxima sección veremos algunos instrumentos específicos para elaborar este tipo de cuestionarios.

Entrevistas estructuradas

Al contrario que el cuestionario, donde las respuestas en los campos de texto suelen ser escuetas o inexistentes, las entrevistas estructuradas cara a cara son una mejor fuente de información cualitativa.

Es recomendable que en las entrevistas esté presente el usuario (puede haber varios usuarios a la vez), el moderador y otros miembros del equipo de desarrollo. Para realizar con éxito estas entrevistas, hay algunos principios básicos de utilidad:

- Antes de la sesión de entrevista, es conveniente revisar brevemente los resultados de la prueba y las notas del moderador para decidir qué problemas encontrados requieren de más información por parte del usuario y cuáles están claros.
- Si la prueba es mixta (cuestionario + entrevista) es recomendable aprovechar para consultar el resultado del cuestionario por si surgen preguntas adicionales.
- La entrevista debería comenzar por permitir al usuario que resuma brevemente su primeras impresiones. Si arrancamos con la pregunta “*Bueno, ¿qué te ha parecido?*” tendremos como respuesta el aspecto al que esa persona ha dado mayor importancia. Además, si el usuario ha acabado frustrado, le da la opción de verbalizar esa frustración antes de empezar a explorar los detalles.
- Al igual que con los cuestionarios, la entrevista debería empezar con preguntas generales sobre la aplicación, sin entrar en funciones detalladas ni en preguntas específicas para un usuario determinado.
- Terminar con las preguntas específicas surgidas a partir de las observaciones durante la sesión o a partir de las respuestas a los cuestionarios.
- Finalmente, es conveniente dejar un rato de discusión abierta entre los usuarios, seguida de discusión abierta incluyendo a los observadores y otros miembros del equipo de trabajo.

5.3.7 Analizar los datos y las observaciones

Tras las pruebas, dispondremos de una enorme cantidad de datos “crudos” pendientes de procesar:

- Resultados de los cuestionarios previos
- Notas del moderador de la sesión
- Notas de otros observadores
- Grabación de la interacción
- Cuestionarios finales
- Notas de la sesión de *debriefing*

En este punto, resulta necesario compilar todos los datos, analizarlos y agruparlos en un informe final que incluya todas las observaciones.

Compilar datos.

Debido a lo variado de las fuentes, esto representa un reto organizativo. Los archivos de audio o vídeo tienen gran tamaño, las notas del moderador probablemente estén en un cuaderno, los resultados de las entrevistas pueden ser también archivos de sonido o notas manuscritas, etc. Es importante contar con los medios para agrupar toda esta información.

Analizar datos.

El análisis de los datos toma muchas formas, y según las actividades incluidas en la evaluación será necesario emplear distintas técnicas de análisis. Aún así, éstas son algunas de las técnicas más comunes:

- **Análisis de las grabaciones:** Los archivos de audio o vídeo de las sesiones contienen mucha información, pero es importante “hacerla visible”. Para esto, es recomendable revisar los vídeos y añadir marcas de tiempo para los eventos más importantes (errores encontrados, frases significativas, intervenciones del moderador, etc). Es recomendable que más de una persona observe y anote cada vídeo, para así asegurar que los eventos del vídeo son entendidos por todo el equipo. Se puede ver un ejemplo en la Figura 5.31.

Código	Comienzo	Final	Tarea	Usuario	Comentarios	Frase
1	02:30	02:36	Lista	Confuso	Desconoce a que se refiere cada columna	[...]Y está columna será las cantidades supongos
1	02:40	02:52	Sugerencias	Un poco Confuso	No ha asociado el botón correspondiente a sugerencias	Dónde están las sugerencias aquí
0	03:32	03:34	Aceptar sugerencias	Agradado	Le gusta el tick de aceptar	Muy fácil de entender
1	04:22	04:27	Ingredientes	Sorprendido	Al volver desde la pantalla ingredientes a la pantalla producto se han borrado las selecciones de marca y supermercado	Esperate que seleccione la marca y el supermercado que se me ha ido
0	03:45	03:47	Ir a producto	Gratamente sorprendido	Se sorprende de la apariencia de la pantalla producto	¡Uy! ¡Con foto y todo!
0	04:46	04:48	Comentarios	Divertido	Le gusta la pantalla de comentarios	¡Ja, ja, ja!
2	05:19	05:21	Mapeo	Impasible	El usuario no reconoce la funcionalidad mapeo y la confunde con un diálogo, al leer la lista de tareas	¿Qué significará mapeo? Pues esto que tengo aquí delante, ¿no?
1	05:27	05:37	Elegir factores	extrañado	Se muestra confuso con los radio buttons	Tengo que quitar lo que yo quiera para elegir lo que yo prefiera
0	05:55	05:59	Finalizar compra	Sorprendido	Le llama la atención el precio final de la compra	¡80€! Creo que me he pasado un poco...

Figure 5.31: Anotaciones de los eventos en un vídeo

- **Análisis de las notas del moderador:** Las notas manuscritas del moderador se deberían organizar también de acuerdo a sus marcas de tiempo. Estas anotaciones se pueden unir a las anteriores, y esto resulta más sencillo si durante la sesión el moderador va anotando marcas de tiempo para sus observaciones.
- **Análisis de los cuestionarios:** El análisis cuantitativo de los cuestionarios intenta evaluar el grado de satisfacción de los usuarios. La teoría detrás de este tipo de análisis está fuera del ámbito de esta asignatura, pero si caben dos consideraciones importantes:
 - Más que la media, nos interesa conocer la distribución de las respuestas, tomando como medida los percentiles de las respuestas.

- Si ha habido eventos negativos durante la prueba de un usuario (un error técnico, un pico de frustración, etc.) es posible que las respuestas de dicho usuario sean muy distintas. Debido a que solemos contar con pocos usuarios, estos casos especiales (llamados *outliers*) pueden afectar mucho al resultado, por lo que puede ser recomendable tratarlos por separado.
- **Análisis de las entrevistas:** Las entrevistas, al igual que las observaciones, dan lugar a mucha información desestructurada. De nuevo, existen técnicas avanzadas de análisis cualitativo que no entran en el ámbito de esta asignatura.

Informe agregado.

El resultado final de todo el análisis sería un informe final que agrega todas las observaciones, estadísticas y resultados de los análisis, pero no necesariamente incluye respuestas a los problemas.

5.3.8 Informe de hallazgos y recomendaciones

Al igual que en las evaluaciones heurísticas, el objetivo final es obtener un documento que resuma los hallazgos, proponga soluciones y especifique prioridades.

Además, es importante contestar a las preguntas de investigación planteadas al principio en el plan de evaluación a través de todos los datos obtenidos y los análisis realizados.

Finalmente, los pasos a seguir para obtener la lista priorizada de cambios son los mismos que en el caso de las evaluaciones heurísticas y no los detallaremos de nuevo.

5.4 Otros métodos de evaluación

Por el momento hemos visto los dos principales formatos para una evaluación de usabilidad (mediante heurísticas o mediante pruebas con usuarios). Pero estos dos formatos no cubren todas las posibilidades de evaluación. En la Figura 5.32 se pueden ver otros muchos métodos de evaluación con usuarios.

A continuación veremos brevemente otras formas y métodos de evaluación.

5.4.1 Evaluación basada en uso real

Ninguno de los dos modelos de evaluación vistos previamente resulta efectivo para comprobar el **uso real en el contexto final de aplicación**. Toda prueba controlada supone unas restricciones adicionales que contaminan los datos (ya hemos mencionado por ejemplo que en las pruebas con técnicas *think-aloud* se cometen **menos** errores que en los escenarios reales).

La mejor forma de evaluar la usabilidad es por tanto permitiendo a los usuarios usar el sistema de forma autónoma en escenarios reales. Para ello, es necesario poder monitorizar

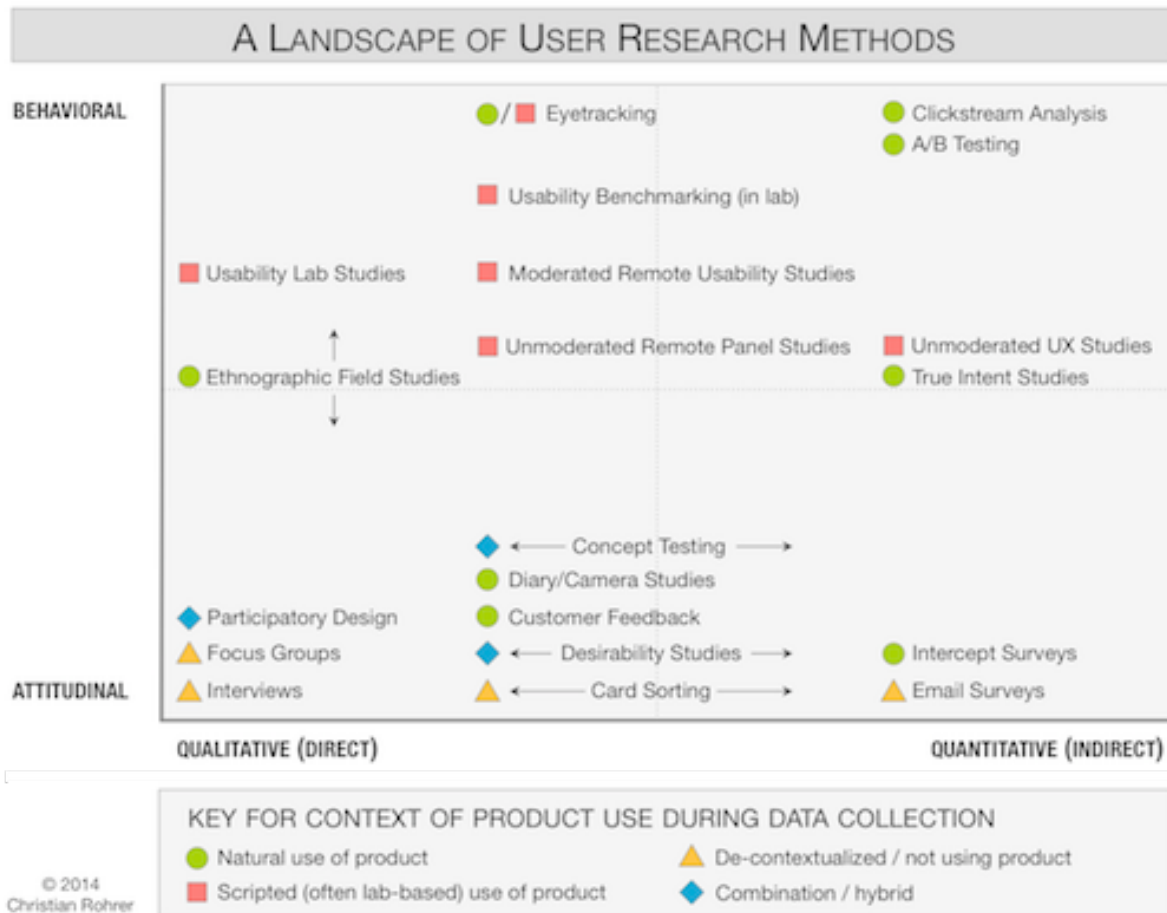


Figure 5.32: Métodos de investigación y evaluación de usuarios. Fuente: [Nielsen/Norman Group](#)

remotamente la interacción, lo cual se suele conseguir mediante sistemas que registran los eventos del sistema y los recolectan para su análisis posterior.

Ventajas

Este tipo de análisis es especialmente útil para identificar:

- Las funciones que se usan más a menudo (por ejemplo, para hacerlas más prominentes dentro del sistema).
- Las funciones que apenas se usan o no se usan nunca (por ejemplo, para hacerlas más fáciles de usar o eliminarlas directamente).
- La frecuencia de los errores a lo largo del tiempo (si el sistema es capaz de detectarlos).

Desventajas

En cambio, este tipo de análisis resulta limitado si queremos medir cosas como:

- La eficiencia a la hora de realizar tareas (no sabemos realmente cuánto tiempo ha tardado el usuario, dado que puede haberse distraído con otra tarea)
- Los mensajes de error que más confunden a los usuarios.
- Los elementos de control que los usuarios no encuentran cuando los necesitan.

Retos tecnológicos

Existen diversas barreras tecnológicas y decisiones de diseño que afectan a la calidad de la información recibida en este tipo de seguimientos:

Nivel de abstracción

Las dos alternativas tradicionales serían la captura de eventos de bajo nivel y la captura de eventos de alto nivel. Capturar a bajo nivel requiere seguir la actividad del ratón y el teclado (o las pulsaciones en una pantalla táctil), lo cual a su vez puede requerir poder acceder directamente a los drivers o a la estructura de seguridad del sistema³. Además, los registros de eventos capturados a bajo nivel son más complicados de interpretar, y se requiere un filtrado activo de los eventos. Por otro lado, la captura de eventos a alto nivel nos permite centrarnos en eventos relevantes desde el punto de vista de la usabilidad, y resulta mucho más sencillo procesar los datos recibidos. En cambio, es más fácil perder información importante si el sistema de captura de eventos no es capaz de reconocer eventos relevantes importantes.

Envío de datos

El envío de datos suele requerir formatos compactos, que saturan poco la red (sobre todo si se están enviando eventos de bajo nivel). Por esto mismo los datos se suelen

³En Windows, los programas antivirus suelen bloquear los programas que intentan capturar directamente eventos de ratón y teclado.

enviar en fragmentos separados (para reducir el tamaño de los paquetes) que luego deben unirse. Si además el envío se hace de forma anónima, es necesario idear un sistema para identificar las sesiones registradas en el lado del servidor.

Interpretación de los datos

Como hemos comentado antes, la interpretación de los datos siempre es dificultosa, y más si se han recolectado datos de bajo nivel. Según los datos recogidos, tenemos distintas opciones:

- Generar informes de texto de alto nivel con la secuencia de pasos del usuario y las opciones que ha elegido (esto suele requerir capturar eventos de alto nivel). Estos informes son luego leídos por humanos.
- Generar informes centrados en aspectos relevantes: aparición de mensajes de error, estadísticas de cuántas veces se ha pulsado un determinado botón, etc. Normalmente este procesamiento se realiza de forma automática, y es posible hacerlo independientemente del nivel de los eventos recogidos.
- Reproducir sesiones: Una alternativa interesante cuando capturamos eventos de bajo nivel es la posibilidad de reproducir la interacción en una versión especial de la aplicación. Esto permite a los diseñadores observar los pasos realizados de una forma muy evidente. Por otro lado, el coste de desarrollo de un sistema así es muy elevado.

5.4.2 Cuestionarios de satisfacción del usuario

Todas las evaluaciones con usuarios (ya sean pruebas tradicionales o grupos de discusión) suelen aprovechar la presencia de los usuarios para administrar cuestionarios que permitan entender mejor (y analizar de forma directa) sus impresiones acerca del producto.

Además de los consejos genéricos aportados en la sección 4.3, existen diversos instrumentos predefinidos que facilitan esta labor, con conjuntos de preguntas que ya han sido validadas previamente. A continuación se describen brevemente algunos de modelos de evaluación e instrumentos concretos para los cuestionarios.

Marco general: Technology Acceptance Model

Desarrollado originalmente por Fred Davis (1989), el Modelo de Aceptación de la Tecnología (TAM) describe los factores que influyen en la aceptación de los sistemas por parte de los usuarios:

- Factores externos
- Utilidad percibida
- Facilidad de uso percibida
- Actitud hacia la posibilidad de usar el sistema
- Intención de usar el sistema

Estos elementos influyen los unos en los otros (en la Figura 5.33, el sentido de las flechas indica que un factor influye sobre el factor al que apunta). A partir del estudio de dichos elementos, podemos predecir el más importante: El uso real del sistema.

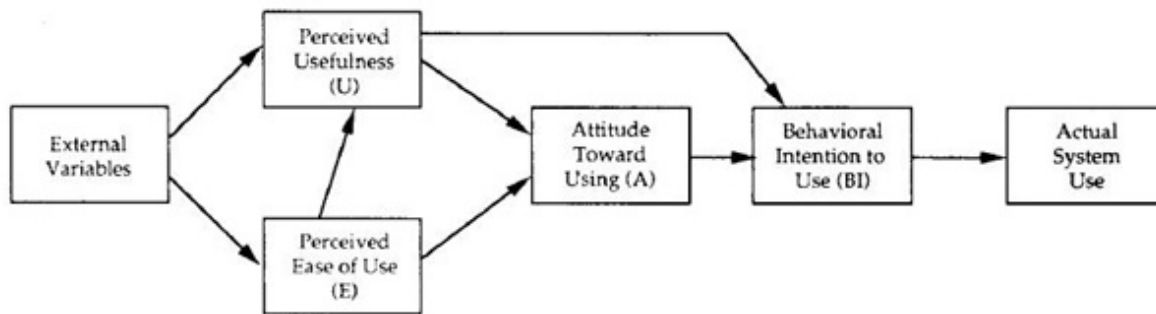


Figure 5.33: Primera versión: Technology Acceptance Model (TAM). Fuente: V. Venkatesh

Esta primera visión del modelo ha sufrido varias iteraciones a medida que el tiempo ha avanzado y aparecían nuevas consideraciones. En las Figuras 5.34 y 5.35 se pueden observar las dos siguientes iteraciones (TAM2 y TAM3) cada vez más complejas. De nuevo, el sentido de las flechas indican qué factores influyen en qué otros, y cuando las flechas apuntan a otras flechas es porque los factores afectan a la relación entre otros factores.

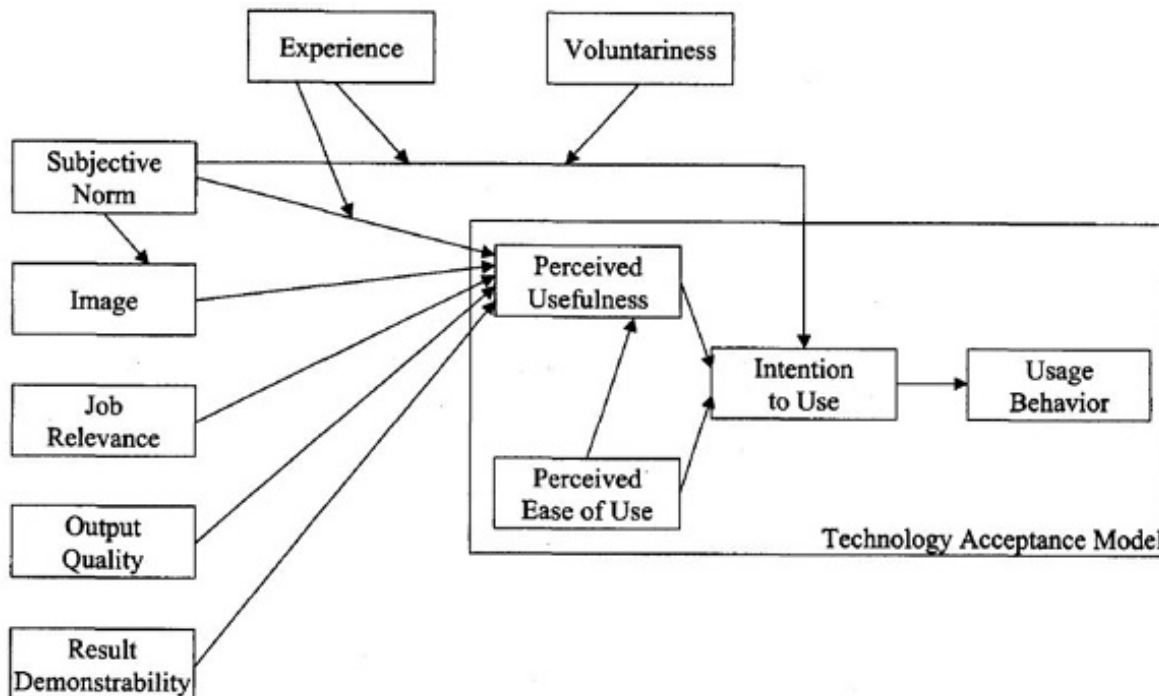


Figure 5.34: Technology Acceptance Model 2 (TAM2). Fuente: V. Venkatesh

A pesar de este crecimiento, la forma más aceptada de los modelos TAM en la actualidad es la denominada Teoría Unificada de la Aceptación y Uso de la Tecnología (UTAUT por

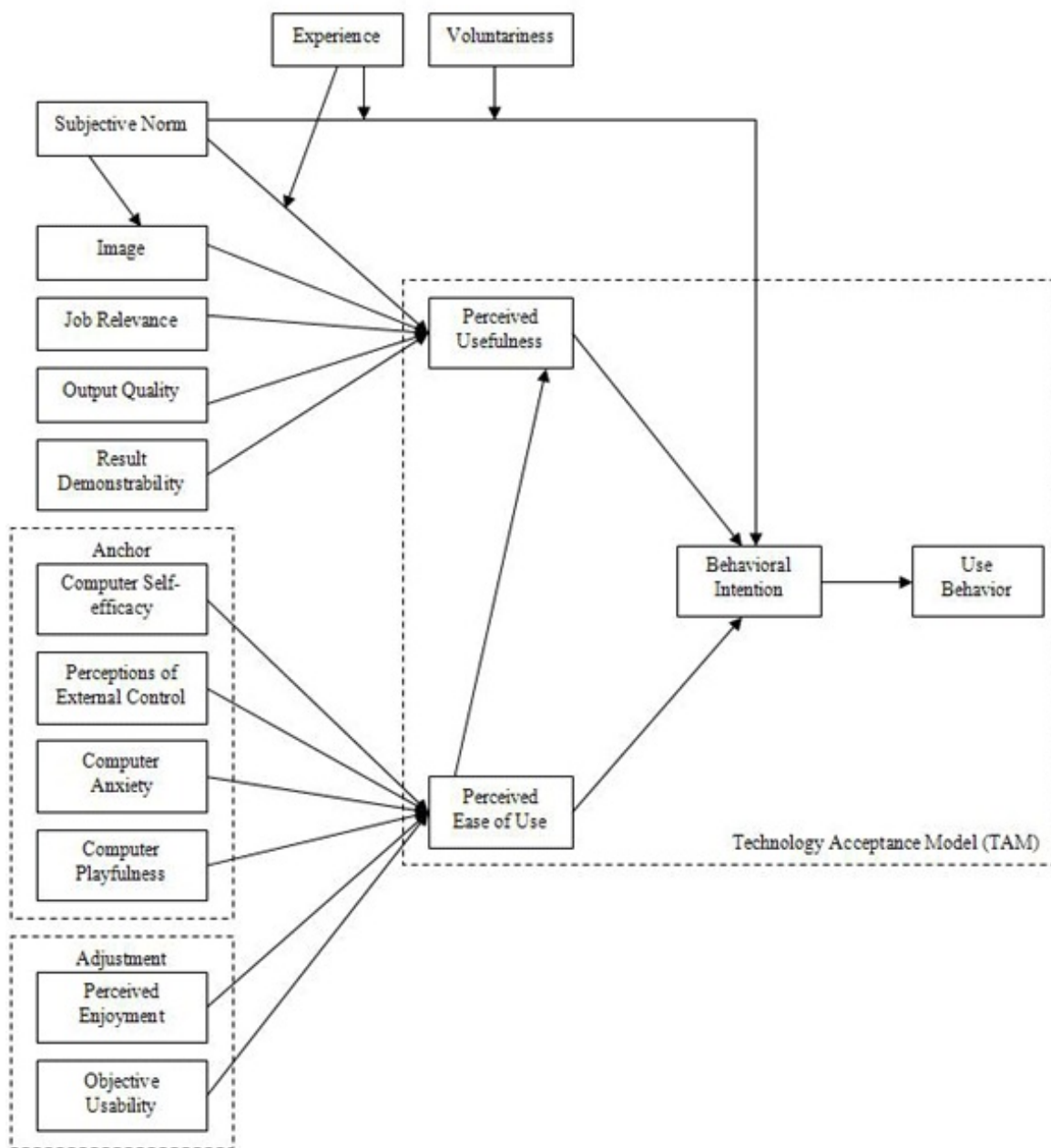


Figure 5.35: Technology Acceptance Model 3 (TAM3). Fuente: V. Venkatesh

sus siglas en inglés, Figura 5.36) que agrupa los elementos de TAM y otros elementos de cuestionarios alternativos para dar lugar a un único modelo.

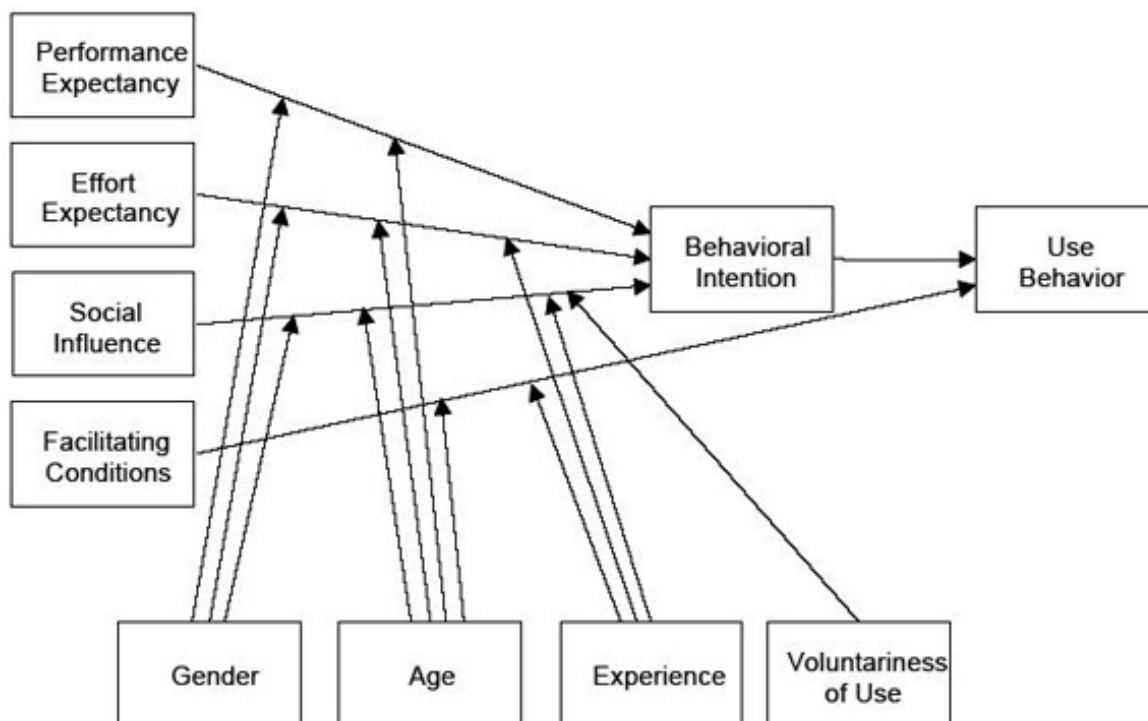


Figure 5.36: Unified Theory of Acceptance and Use of Technology (UTAUT). Fuente: V. Venkatesh

El modelo general de UTAUT se acompaña además de un cuestionario con preguntas predefinidas relativamente generalizables. El cuestionario completo se incluye en el anexo que acompaña a este tema.

Otros instrumentos

Aunque UTAUT es probablemente el instrumento más potente, a menudo se critica su exceso de complejidad. Por este motivo, algunas instituciones han optado por recurrir a cuestionarios simplificados.

Dentro de estos cuestionarios simplificados, algunos de los más extendidos en la actualidad son los propuestos por IBM para sus pruebas de usabilidad.

Post-Study System Usability Questionnaire (PSSUQ).

Es un cuestionario que se emplea justo después de realizar una evaluación con usuarios como las descritas en la sección 4.3. Se emplea para calcular cuatro variables específicas: La puntuación general de satisfacción (OVERALL), la utilidad del sistema (SYSUSE), la calidad de la información (INFOQUAL) y la calidad de la interfaz (INTERQUAL).

CSUQ.

Es un cuestionario prácticamente idéntico al anterior, pero generalizado para poder usarlo también en evaluaciones genéricas de programas existentes y no sólo inmediatamente después de una prueba de laboratorio. El resultado final se calcula de forma similar al anterior.

Ambos cuestionarios se pueden encontrar en el anexo que acompaña a este tema.

System Usability Scale (SUS).

Es uno de los cuestionarios más usados sobre percepción de usabilidad de un sistema. Se compone de 10 enunciados con respuestas en una escala de Likert de 5 valores (completamente en desacuerdo – completamente de acuerdo). Proporciona una puntuación, no un porcentaje, que es necesario normalizar para que quede en una escala de notas similar a la usada en la educación norteamericana (A..F). En su lugar, se pueden usar percentiles. Una puntuación normalizada por encima de 68 implica que el sistema está por encima de la media desde el punto de vista de usabilidad. Es un cuestionario rápido, bastante fiable y está validado pero no permite detectar problemas de usabilidad.

Este cuestionario está disponible [a través de este artículo de MeasuringU](#), en [Usability Geek](#) y en la página [UsabilityGov](#).

5.4.3 Grupos de discusión

Los grupos de discusión (o grupos focales, del inglés *focus groups*) son una técnica informal que se puede usar para evaluar las necesidades y las reacciones de los usuarios ante las herramientas.

Consiste en reunir a un grupo de usuarios (típicamente entre seis y nueve) para discutir conceptos e identificar problemas y posibles soluciones. Las sesiones se organizan para que los usuarios se sientan cómodos (una conversación entre amigos) pero con un moderador que se encarga de que se cubran todos los aspectos relevantes para la evaluación.

Los grupos de discusión se distinguen de las evaluaciones con usuarios en que:

- En los grupos de discusión no es necesario que el sistema evaluado sea mínimamente funcional, ya que los usuarios pueden discutir sobre interfaces y conjuntos de funcionalidades descritos en papel.
- Se pueden emplear grupos de discusión para valorar las ventajas y desventajas de varios diseños o bocetos alternativos en paralelo, mostrando todas las opciones y dirigiendo la discusión hacia ventajas y desventajas .
- Dan una visión más cercana al usuario final que una evaluación heurística pero sin los requisitos de formalidad de una evaluación con usuarios.

Aún así, los grupos de discusión no están exentos de problemas:

- Requieren de una preparación importante por parte de los moderadores, que querrán fomentar la discusión abierta pero en realidad deben seguir un guión estructurado.
- Las dinámicas de grupo pueden generar posiciones enfrentadas y convertir la conversación en un ejercicio dialéctico (parecerá mejor la interfaz preferida por el usuario más locuaz). También pueden darse situaciones de conflicto en las que un usuario defiende una función sólo por no admitir que estaba equivocado.
- Cuando invitamos a los usuarios a exponer en público sus ideas frente a desconocidos, los usuarios querrán proyectar una imagen positiva de sí mismos. Esto puede hacer que se resistan a reconocer que desconocen ciertos conceptos, o que no pidan ciertas funcionalidades útiles que les puedan hacer parecer vagos o irresponsables en el momento de pedirlos.
- Como sucede en otros métodos basados en preguntar al usuario, **los usuarios no son diseñadores** y lo que *creen* que necesitan no es siempre lo que realmente necesitan.

Por todos estos motivos, en los grupos de discusión es preferible centrarse más en estudiar las impresiones, los argumentos y los comentarios aislados que en las cosas que los usuarios piden explícitamente.

5.4.4 Evaluaciones A/B

Las [Evaluaciones A/B](#) son Evaluaciones de uso real, sin una tarea concreta que realizar, en las que se pretende evaluar una determinada alternativa de diseño (Figura 5.37). Distintos grupos de participantes trabajan con dos versiones distintas del sistema, cada una de las cuales presenta una variación concreta de un mismo elemento. Los elementos que se pueden evaluar son muchos y muy distintos:

- Una determinada organización o *layout* de la información del sistema.
- Mensajes. Ej: cuál es la diferencia entre “Cómpralo ya” o “Añádelo al carrito”.
- Formularios en una o varias páginas.
- Colores de iconos.
- ...

El objetivo de esta evaluación suele estar especialmente relacionada con objetivos de negocio del producto: tasa de conversión de usuarios, número de usuarios registrados con distintos formularios y llamadas a la acción (*call to action*), etc.

Algunas recomendaciones relacionadas con las evaluaciones A/B:

- Se recomienda que se realice el estudio de ambas variaciones simultáneamente. Por ejemplo, si se realiza sobre una página web, entonces se redireccionará la mitad del tráfico a una variación mientras que la otra mitad será redirigida a la variación alternativa.
- Evitar usar las variaciones con usuarios conocedores del sistema. Puede producirles cierta confusión y añaden sesgo a la evaluación.
- Hay que ser consistentes con la variación, de modo que aparezca igual en todas

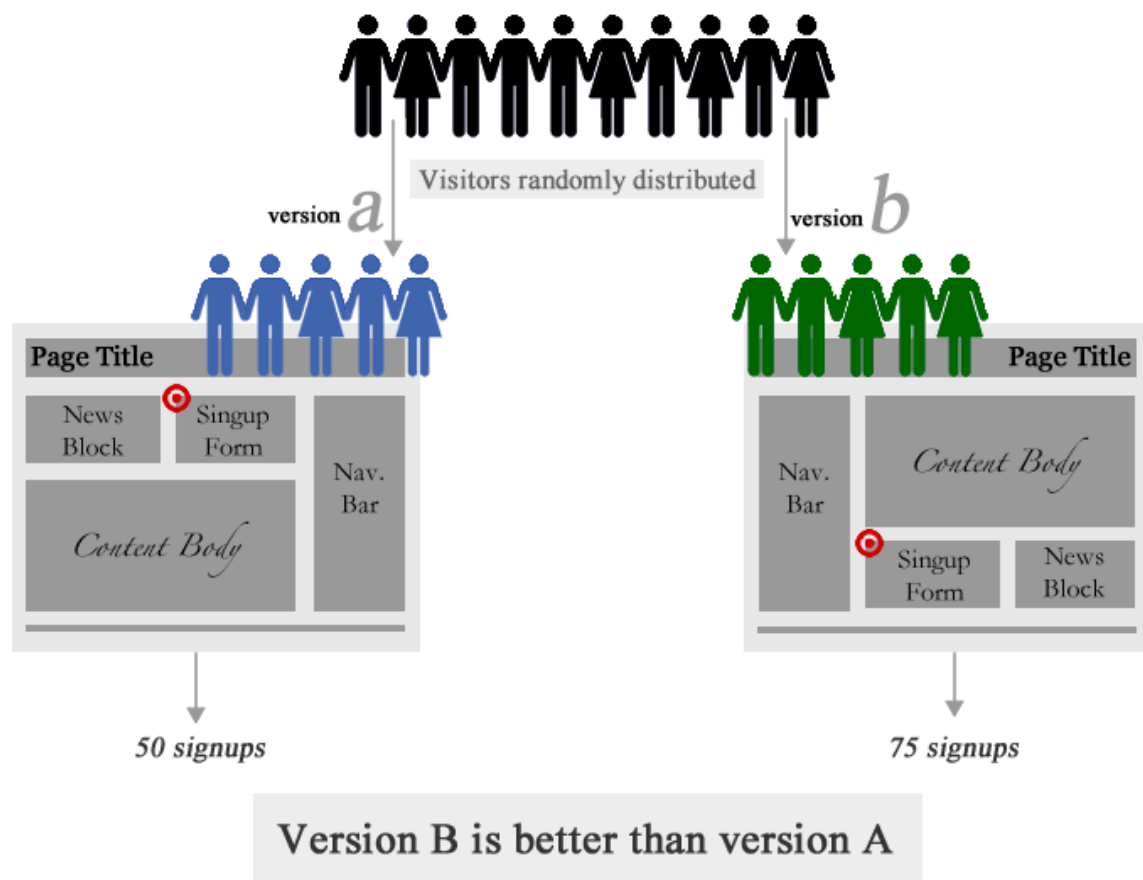


Figure 5.37: Evaluaciones A/B. Fuente: [The Ultimate Guide to A/B Testing](#)

las partes del sistema.

- Usar suficientes personas para evaluar la significancia estadística. Estas evaluaciones requieren verificar que el cambio o variación ha producido una mejora *real* de acuerdo a los objetivos formulados por lo que es necesario utilizar pruebas estadísticas que lo verifiquen. Estas pruebas necesitan un mínimo de usuarios para verificar los resultados con un cierto intervalo de confianza por lo que hay que echar mano de la estadística para conocer el número de usuario que han de ser objeto de la evaluación para obtener resultados estadísticamente significativos.

5.4.5 Evaluación con entrenamiento previo

Habitualmente evaluamos queriendo conocer el comportamiento del usuario al enfrentarse a una aplicación por primera vez, sin ningún tipo de ayuda. En general esto es consistente con los requisitos de cualquier aplicación desarrollada:

- Si es una aplicación para usuarios finales, se espera que el usuario aprenda por sí mismo a usar una herramienta.
- Si es una aplicación empresarial, es deseable que la aplicación se pueda usar sin necesidad de entrenamiento previo.

Pero este concepto no es necesariamente útil para determinados saltos e innovaciones, como ha pasado en muchas ocasiones en el mundo de la informática:

En 1987, todavía inmersos en la “era de la consola de comandos”, Tombaugh realizó un estudio sobre qué interfaces visuales eran más efectivas para una aplicación de lectura de textos: una interfaz de ventana única frente a una interfaz con múltiples ventanas. Tenían un texto delante y tenían que localizar respuestas a una serie de preguntas consultando el texto. En la primera prueba, los usuarios de ventana única tardaban 72 segundos en responder, y los usuarios con ventanas múltiples tardaban 85 (la ventana única era mejor). En un segundo test con otros usuarios, comenzaron con una sesión de entrenamiento de 30 minutos sobre el uso de la aplicación y el uso del ratón (¡era 1987!). Los nuevos resultados fueron 66 segundos para la ventana única y 56 segundos para la ventana múltiple.

En el ejemplo anterior, es evidente que el primero de los diseños era mejor para “walk-up-and-use”, pero el otro era mejor para uso continuado. Una evaluación directa sin entrenamiento hubiese invitado a escoger la interfaz menos eficiente en escenarios reales.

5.4.6 Un caso muy especial: usabilidad y experiencia de usuario en videojuegos

Todo lo estudiado en este tema 4 se ha presentado de forma genérica, ofreciendo una perspectiva general de los objetivos y necesidades de una evaluación. Ahora bien, a la hora de diseñar y evaluar es conveniente tener en cuenta el propósito de la aplicación, ya que en ocasiones encontramos que algunas familias de productos desafían las convenciones de usabilidad vistas en este tema.

Características únicas de los videojuegos

Ya en el tema 1 definíamos la usabilidad de acuerdo con el estándar ISO 9241 ([parte 11: Orientaciones sobre Usabilidad](#)) como:

la cualidad por la que un producto puede ser usado por un *usuario específico* para conseguir unos *objetivos específicos* con eficiencia, efectividad y satisfacción en un *contexto de uso específico*.

Sin embargo, los juegos digitales son un tipo muy específico de software con requisitos únicos, lo cual supone retos únicos de usabilidad. La mayoría instrumentos y métricas de usabilidad existentes están orientadas a las herramientas de productividad general (empresarial o personal), y por lo tanto se centran en aspectos como la productividad, la eficacia, y el número de errores.

Pero los videojuegos son completamente diferentes, centrándose más en el proceso que en los resultados, teniendo como objetivo principal el disfrute (no la productividad) y primando la variedad y la sorpresa sobre la uniformidad y la consistencia.

Similarmente, los principios de usabilidad vistos para el software de productividad nos sugieren que debería ser trivial para el usuario adquirir un alto nivel de competencia con la función, y que los puntos donde el usuario duda o se atasca deberían considerarse como eventos negativos. En cambio, los juegos aspiran a que los usuarios exploren, se equivoquen y a menudo no sepan qué hacer.

En particular, una parte importante del mecanismo de la diversión en los juegos es que presentan desafíos al usuario e intentan mantenerle siempre un paso más allá de su nivel de habilidad. También se fomentan los errores y el uso comedido de la frustración⁴ y de los errores del usuario para incrementar la satisfacción cuando se consigue la victoria.

En resumen, el diseño de un videojuego viola sistemáticamente muchos de los principios de usabilidad vistos en este tema.

Estudios de usabilidad en videojuegos

El hecho de que los videojuegos violen muchos de los principios de usabilidad vistos en este tema no es excusa para ignorar la importancia de la usabilidad y de las evaluaciones de usabilidad. Simplemente requiere mecanismos de análisis más profundos que tengan en cuenta las características únicas del medio.

Aunque los juegos intenten ser frustrantes y fomenten los errores para mejorar la diversión, hay que distinguir estos bucles de dificultad estudiada (centrados en la habilidad del jugador) de las frustraciones derivadas de un mal diseño de la interfaz de usuario de los juegos.

Es necesario desarrollar métricas que distingan la frustración y los errores que suceden como consecuencia del diseño de jugabilidad de aquellos derivados del diseño de sus sistemas de control.

⁴James Paul Gee describió los videojuegos como *experiencias placenteramente frustrantes*.

Existen metodologías específicas para medir la usabilidad de los videojuegos que toman en consideración sus características únicas. Aunque el estudio de los detalles específicos de estas metodologías queda fuera del alcance la asignatura, a modo de ampliación del material se propone el estudio del siguiente artículo:

Pablo Moreno-Ger, Javier Torrente, Yichuan Grace Hsieh, and William T. Lester, “Usability Testing for Serious Games: Making Informed Design Decisions with User Data,” *Advances in Human-Computer Interaction*, vol. 2012, Article ID 369637, 13 pages, 2012. doi:10.1155/2012/369637

Diseño de usabilidad para videojuegos

Además de medir la usabilidad de los videojuegos, existe un creciente interés por el estudio de cómo adaptar los métodos de diseño centrados en el usuario (como el Diseño Guiado por Objetivos) para contemplar las características únicas de este medio.

Por ejemplo, el concepto de *persona* se puede extender para cubrir el estudio de las denominadas *play-personas*, definidas como aquellas *personas* que incluyen indicaciones sobre sus actitudes preferentes de interacción y navegación en relación con las distintas mecánicas de juegos.

Este concepto es compatible con los procesos de definición de *personas* vistos en el tema 3, comenzando por la elaboración de la *hipótesis de personas* basándose en estudios de mercado y mecánicas de otros juegos, que luego son completadas mediante un proceso de **investigación con usuarios** al igual que las personas tradicionales.

Aunque el proceso ingenieril y la ejecución de diseños guiados por objetivos para videojuegos también queda fuera del alcance de la asignatura, se propone el estudio del siguiente artículo:

Canossa, Alessandro, and Anders Drachen. “Play-personas: behaviours and belief systems in user-centred game design.” *Human-Computer Interaction–INTERACT 2009*. Springer Berlin Heidelberg, 2009. 510-523.

Anexo: Cuestionarios estandarizados: UTAUT

El cuestionario UTAUT completo, junto con la explicación detallada de cómo usarlo se puede encontrar en el artículo siguiente:

Venkatesh, V.; Morris; Davis; Davis (2003), “User Acceptance of Information Technology: Toward a Unified View”, *MIS Quarterly* 27 (3), pp. 425–478

Los ítems específicos del cuestionario se reproducen a continuación.

Perceived Usefulness

The degree to which a person believes that using a particular system would enhance his or her job performance.

1. Using the system in my job would enable me to accomplish tasks more quickly.
2. Using the system would improve my job performance.
3. Using the system in my job would increase my productivity.
4. Using the system would enhance my effectiveness on the job.
5. Using the system would make it easier to do my job.
6. I would find the system useful in my job.

Job-fit

How the capabilities of a system enhance an individual’s job performance.

1. Use of the system will have no effect on the performance of my job (reverse scored).
2. Use of the system can decrease the time needed for my important job responsibilities.
3. Use of the system can significantly increase the quality of output on my job.
4. Use of the system can increase the effectiveness of performing job tasks.
5. Use can increase the quantity of output for the same amount of effort.
6. Considering all tasks, the general extent to which use of the system could assist on the job.

Relative Advantage

The degree to which using an innovation is perceived as being better than using its precursor.

1. Using the system enables me to accomplish tasks more quickly.
2. Using the system improves the quality of the work I do.
3. Using the system makes it easier to do my job.
4. Using the system enhances my effectiveness on the job.
5. Using the system increases my productivity.

Outcome Expectations

Outcome expectations relate to the consequences of the behavior. Based on empirical evidence, they were separated into performance expectations (job-related) and personal expectations (individual goals).

If I use the system...

1. I will increase my effectiveness on the job.
2. I will spend less time on routine job tasks.
3. I will increase the quality of output of my job.
4. I will increase the quantity of output for the same amount of effort.
5. My coworkers will perceive me as competent.
6. I will increase my chances of obtaining a promotion.
7. I will increase my chances of getting a raise.

Perceived Ease of Use

The degree to which a person believes that using a system would be free of effort.

1. Learning to operate the system would be easy for me.
2. I would find it easy to get the system to do what I want it to do.
3. My interaction with the system would be clear and understandable.
4. I would find the system to be flexible to interact with.
5. It would be easy for me to become skillful at using the system.
6. I would find the system easy to use.

Complexity

The degree to which a system is perceived as relatively difficult to understand and use.

1. Using the system takes too much time from my normal duties.
2. Working with the system is so complicated, it is difficult to understand what is going on.
3. Using the system involves too much time doing mechanical operations (e.g., data input).
4. It takes too long to learn how to use the system to make it worth the effort.

Ease of Use

The degree to which using an innovation is perceived as being easy to use.

1. My interaction with the system is clear and understandable.
2. I believe that it is easy to get the system to do what I want it to do.
3. Overall, I believe that the system is easy to use.
4. Learning to operate the system is easy for me.

Subjective Norm

The person's perception that most people who are important to him think he should or should not perform the behavior in question.

1. People who influence my behavior think that I should use the system.
2. People who are important to me think that I should use the system.

Social Factors

The individual's internalization of the reference group's subjective culture, and specific interpersonal agreements that the individual has made with others, in specific social situations.

1. I use the system because of the proportion of coworkers who use the system.
2. The senior management of this business has been helpful in the use of the system.
3. My supervisor is very supportive of the use of the system for my job.
4. In general, the organization has supported the use of the system.

Image

The degree to which use of an innovation is perceived to enhance one's image or status in one's social system.

1. People in my organization who use the system have more prestige than those who do not.
2. People in my organization who use the system have a high profile.
3. Having the system is a status symbol in my organization

Perceived Behavioral Control

Reflects perceptions of internal and external constraints on behavior and encompasses selfefficacy, resource facilitating conditions, and technology facilitating conditions.

1. I have control over using the system.
2. I have the resources necessary to use the system.
3. I have the knowledge necessary to use the system.
4. Given the resources, opportunities and knowledge it takes to use the system, it would be easy for me to use the system.
5. The system is not compatible with other systems I use.

Facilitating Conditions

Objective factors in the environment that observers agree make an act easy to do, including the provision of computer support.

1. Guidance was available to me in the selection of the system.
2. Specialized instruction concerning the system was available to me.
3. A specific person (or group) is available for assistance with system difficulties.

Compatibility

The degree to which an innovation is perceived as being consistent with existing values, needs, and experiences of potential adopters.

1. Using the system is compatible with all aspects of my work.
2. I think that using the system fits well with the way I like to work.
3. Using the system fits into my work style.

Attitude Toward Behavior

An individual's positive or negative feelings about performing the target behavior.

1. Using the system is a bad/good idea.
2. Using the system is a foolish/wise idea.
3. I dislike/like the idea of using the system.
4. Using the system is unpleasant/pleasant.

Intrinsic Motivation

The perception that users will want to perform an activity for no apparent reinforcement other than the process of performing the activity per se.

1. I find using the system to be enjoyable
2. The actual process of using the system is pleasant.
3. I have fun using the system.

Affect Toward Use

Feelings of joy, elation, or pleasure; or depression, disgust, displeasure, or hate associated by an individual with a particular act.

1. The system makes work more interesting.
2. Working with the system is fun.
3. The system is okay for some jobs, but not the kind of job I want. (R)

Affect

An individual's liking of the behavior.

1. I like working with the system.
2. I look forward to those aspects of my job that require me to use the system.
3. Using the system is frustrating for me. (R)
4. Once I start working on the system, I find it hard to stop.
5. I get bored quickly when using the system. (R)

Anexo: Cuestionarios estandarizados: IBM

Los cuestionarios empleados por IBM y su evolución histórica se detallan en el artículo siguiente:

Lewis, James R. "IBM computer usability satisfaction questionnaires: psychometric evaluation and instructions for use." *International Journal of Human-Computer Interaction* 7.1 (1995): 57-78.

Los ítems específicos del cuestionario se reproducen a continuación.

Post-Study System Usability Questionnaire (PSSUQ)

1. Overall, I am satisfied with how easy it is to use this system.
2. It was simple to use this system.
3. I could effectively complete the tasks and scenarios using this system.
4. I was able to complete the tasks and scenarios quickly using this system.
5. I was able to efficiently complete the tasks and scenarios using this system.
6. I felt comfortable using this system.
7. It was easy to learn to use this system.
8. I believe I could become productive quickly using this system.
9. The system gave error messages that clearly told me how to fix problems.
10. Whenever I made a mistake using the system, I could recover easily and quickly.
11. The information (such as on-line help, on-screen messages and other documentation) provided with this system was clear.
12. It was easy to find the information I needed.
13. The information provided for the system was easy to understand.
14. The information was effective in helping me complete the tasks and scenarios.
15. The organization of information on the system screens was clear.
16. The interface of this system was pleasant.
17. I liked using the interface of this system.
18. This system has all the functions and capabilities I expect it to have.
19. Overall, I am satisfied with this system.

Computer System Usability Questionnaire (CSUQ)

1. Overall, I am satisfied with how easy it is to use this system.
2. It is simple to use this system.
3. I can effectively complete my work using this system.
4. I am able to complete my work quickly using this system.
5. I am able to efficiently complete my work using this system.
6. I feel comfortable using this system.
7. It was easy to learn to use this system.
8. I believe I became productive quickly using this system.
9. The system gives error messages that clearly tell me how to fix problems.
10. Whenever I make a mistake using the system, I recover easily and quickly.

11. The information (such as on-line help, on-screen messages and other documentation) provided with this system is clear.
12. It is easy to find the information I need.
13. The information provided with the system is easy to understand.
14. The information is effective in helping me complete my work.
15. The organization of information on the system screens is clear.
16. The interface of this system is pleasant.
17. I like using the interface of this system.
18. This system has all the functions and capabilities I expect it to have.
19. Overall, I am satisfied with this system.

Procesamiento

Los datos de los dos cuestionarios anteriores se procesan de forma directa para calcular las cuatro variables de aceptabilidad del sistema:

- OVERALL - Media de las preguntas 1 a 19
- SYSUSE - Media de las preguntas 1 a 8
- INFOQUAL - Media de las preguntas 9 a 15
- INTERQUAL - Media de las preguntas 16 a 18.