

Evaluación Heurística

Visualización e Interfaces

Fuente: <https://blog.interactius.com/>

Evaluación Heurística

- Una Evaluación Heurística (EH) es un **método de inspección de la usabilidad** sin usuarios. Este, consiste en examinar la calidad de uso de una interfaz por parte de varios **evaluadores expertos**, a partir del cumplimiento de unos principios reconocidos de usabilidad: **los heurísticos**.
- El objetivo de la EH es medir la **calidad de la interfaz** de cualquier aplicativo en relación a su facilidad para ser aprendido y usado por primera vez.
- Las EH inspeccionan **problemas potenciales** . El evaluador se pone en la piel del usuario real del sistema, intentando **predecir** los errores que podrá encontrarse.

Principios Heurísticos

- **10 Principios Heurísticos de Nielsen**, de su libro “Usability Inspection Methods” (1994).
- **8 reglas de oro** de Ben Schneiderman, de su libro “Designing the User Interface: Strategies for Effective Human-Computer Interaction” (1987).
- **Lista de comprobación de ítems** de Deniese Pierotti (2004), es la que utiliza la empresa Xerox para evaluar la calidad de sus interfaces.

10 Principios Heurísticos de Nielsen



1. **Visibilidad del estado del sistema:** El sistema siempre debe mantener a los usuarios informados sobre lo que ocurre, a través de una retroalimentación apropiada en un tiempo razonable.
2. **Empate entre el sistema y el mundo real:** El sistema debe hablar en el lenguaje del usuario, con palabras, frases y conceptos familiares para él. Utilizar convenciones del mundo real, haciendo que la información aparezca en un orden natural y lógico.
3. **Control y libertad del usuario:** A menudo los usuarios eligen funcionalidades por error y necesitan una “puerta de emergencia” para salir del estado indeseado. Ofrecer soporte para deshacer y rehacer acciones.

10 Principios Heurísticos de Nielsen



4. **Consistencia y estándares:** Los usuarios no deben tener que preguntarse si las diversas palabras, situaciones, o acciones significan las misma cosa. Que se sigan las normas y convenciones de la plataforma sobre la que está implementando el sistema.
5. **Prevención de errores:** Antes que diseñar buenos mensajes de error, es mejor evitar que el problema ocurra.
6. **Reconocimiento mejor que recuerdo:** Minimizar la carga de memoria del usuario haciendo que los objetos, las acciones y las opciones estén visibles. El usuario no debería tener que recordar la información de una parte del diálogo a otra.

10 Principios Heurísticos de Nielsen



7. **Flexibilidad y eficiencia de uso:** Los aceleradores, no vistos por el usuario principiante, mejoran la interacción para el usuario experto de tal manera que el sistema puede servir para usuarios inexpertos y experimentados. Es importante que el sistema permita personalizar acciones frecuentes.
8. **Diseño estético y minimalista:** Los diálogos no deberían contener información irrelevante o que se necesite raramente. Cada unidad extra de información en un diálogo compite con la información importante, disminuyendo su visibilidad relativa.

10 Principios Heurísticos de Nielsen



9. **Ayudar a reconocer, diagnosticar y recuperarse de errores:** Los mensajes de error deben estar expresados en lenguaje llano (sin códigos), indicando con precisión el problema y sugiriendo una solución.
10. **Ayuda y documentación:** Aunque es mejor que se pueda usar el sistema sin documentación, es necesario proveer al usuario de ayuda y documentación. Esta tiene que ser fácil de buscar, centrada en la tareas del usuario, con información de las etapas a realizar y que no sea muy extensa.

8 reglas de oro - Ben Schneiderman



1. **Luchar por la consistencia.** Secuencias de acciones consistentes deberían ser necesarias en situaciones similares; idéntica terminología debe utilizarse en anuncios, menús y pantallas de ayuda, y los comandos consistentes deben ser empleados en todo.
2. **Permite a los usuarios frecuentes utilizar accesos directos.** A medida que la frecuencia de uso aumenta, también lo hacen los deseos del usuario para reducir el número de acciones y aumentar el ritmo de interacción. Acrónimos y abreviaturas, las teclas de función, los comandos ocultos, y macro instalaciones son muy útiles para un usuario experto.

8 reglas de oro - Ben Schneiderman



3. **Ofrece comentarios informativos.** Por cada operador de acción, debe haber algún sistema de retroalimentación. Para acciones frecuentes y de menor uso, la respuesta puede ser modesta, mientras que para los poco frecuentes y las principales acciones, la respuesta debería ser más sustancial.
4. **Diseño de diálogo para producir la clausura.** Acciones secuenciales debe organizarse en grupos con un comienzo, intermedio y final.
5. **Ofrece una manipulación de errores simples.** Si aparece un error, el sistema debería ser capaz de detectar el error y ofrecer de manera sencilla y comprensible una manera para identificar el error.

8 reglas de oro - Ben Schneiderman



6. **Permitir un fácil retroceso de las acciones.** Las unidades de reversibilidad pueden ser una sola acción, una entrada de datos, o un grupo de acciones.
7. **Apoyo interno a un enfoque de control total.** Los usuarios experimentados desean el sentido de que están a cargo del sistema y que el sistema responde a sus acciones. Diseña el sistema para que los usuarios inicien las acciones en lugar de las respuestas.
8. **Reducir la carga de la memoria a corto plazo.** La limitación de recursos humanos de procesamiento de la información en la memoria a corto plazo exige que se muestren los contenidos de manera sencilla.

Conclusiones

- La Evaluación heurística es una técnica crucial y **muy útil** a la hora de encontrar errores de usabilidad, sobre todo en **fases muy tempranas** de un proyecto de diseño web.
- La popularización de los heurísticos de Nielsen ha creado una falsa creencia que da a entender que solamente con cumplir una serie de **principios heurísticos** ya basta para considerar una aplicación como **usable**.
- Las EH en ningún caso **sustituyen** las evaluaciones de usabilidad con usuarios reales -ej. Walkthrough o Cognitive Walkthrough (Recorrido Cognitivo)-, solamente las **complementan**.

Evaluación heurística: ¿Cuándo hacerla?

- En **fases iniciales de un proyecto**, es cuando se adapta mejor y se obtienen mejores resultados. Como no hay material suficientemente firme para efectuar un test con usuarios, se pueden proporcionar maquetas sobre papel o prototipos para detectar los primeros problemas de usabilidad.
- Durante el desarrollo se pueden realizar la EH **sobre primeras versiones**, para localizar y corregir errores a bajo costo.
- Sobre **sistemas en funcionamiento**, a modo de consultorías.

Evaluación heurística: Ventajas

- La EH proporciona un **feedback rápido y relativamente económico**, en comparación con otras técnicas. Ya que además de emplear pocos recursos materiales y humanos, requiere de menor tiempo de preparación y ejecución que otras técnicas.
- Se puede aplicar en **cualquier fase del proyecto**, obteniendo sobre todo buenos resultados en fases tempranas del proyecto.
- Existe la posibilidad de **combinarla con otras metodologías** de test de la usabilidad.

Evaluación heurística: ¿Con quienes?

- Hay dos factores muy importantes, que en función de la naturaleza del proyecto, influirán directamente en la selección de este grupo: el **número de participantes** y su **perfil**.
- Algunos autores sostienen que los profesionales de la usabilidad suelen detectar problemas que no responden al uso real de la aplicación, y es por eso, que plantean que es mejor que la inspección la lleven a cabo diferentes perfiles de evaluadores, tales como **desarrolladores** y **potenciales usuarios**.

Evaluación heurística: ¿Cómo hacerla?

- **Preparación:** conocer el ámbito y el contexto del sistema a evaluar, elegir cuales de los principios heurísticos resultarán pertinentes a evaluar, definir las *subheurísticas*. Seleccionar y formar a los evaluadores (entre 3 y 5 sería lo ideal).
- **Ejecución:** los evaluadores seleccionados inspeccionarán más de una vez la interfaz (durante 1 o 2 horas) en las mismas **condiciones y entorno de trabajo**.
- **Análisis:** todos los evaluadores harán una puesta en común de los resultados y puntuaciones individuales, y se redactará un **informe final** en el que se ordenarán los hallazgos de los más problemáticos a los que menos y se eliminarán los que sean duplicados o similares.

Plantilla EH de Deniese Pierotti (XEROX)

3. User Control and Freedom

Users should be free to select and sequence tasks (when appropriate), rather than having the system do this for them. Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Users should make their own decisions (with clear information) regarding the costs of exiting current work. The system should support undo and redo.

#	Review Checklist	Yes No N/A	Comments
3.1	If setting up windows is a low-frequency task, is it particularly easy to remember?	<input type="radio"/> <input type="radio"/> <input type="radio"/>	
3.2	In systems that use overlapping windows, is it easy for users to rearrange windows on the screen?	<input type="radio"/> <input type="radio"/> <input type="radio"/>	
3.3	In systems that use overlapping windows, is it easy for users to switch between windows?	<input type="radio"/> <input type="radio"/> <input type="radio"/>	
3.4	When a user's task is complete, does the system wait for a signal from the user before processing?	<input type="radio"/> <input type="radio"/> <input type="radio"/>	
3.5	Can users type-ahead in a system with many nested menus?	<input type="radio"/> <input type="radio"/> <input type="radio"/>	
3.6	Are users prompted to confirm commands that have drastic, destructive consequences?	<input type="radio"/> <input type="radio"/> <input type="radio"/>	
3.7	Is there an "undo" function at the level of a single action, a data entry, and a complete group of actions?	<input type="radio"/> <input type="radio"/> <input type="radio"/>	