

10 PRINCIPIOS DE NIELSEN



Principio de visibilidad del estado del sistema

Principio de usabilidad que exige que un sistema mantenga al usuario informado de lo que sucede en todo momento, proporcionando retroalimentación apropiada y oportuna.

Ejemplos:

- WhatsApp: Lo aplica mediante el uso de los checks.
- Uber: Nos muestra la localización del conductor en cada momento.

Contraejemplo:

- Máquinas expendedoras: No muestran si reconocen o no el dinero.

Principio de conciencia entre sistema y mundo real

Un sistema debe usar un lenguaje, conceptos y convenciones que sean familiares y entendibles para el usuario, en lugar de jerga técnica o de la propia aplicación.

Ejemplos:

- Carrito de compras en Amazon: usa el icono de un carrito.
- Notas de iOS: simula un cuaderno de notas.

Contraejemplo:

- Cajeros automáticos antiguos: muestran términos técnicos como 'saldo contable' que no todo el mundo puede entender.

Principio de control y libertad del usuario

La interfaz de un sistema proporciona retroalimentación constante al usuario sobre lo que está ocurriendo en tiempo real.

Ejemplos:

- Google Docs: permite deshacer y rehacer fácilmente.

- Netflix: salir de una reproducción con un toque.

Contraejemplo:

- Puertas giratorias automáticas: no permiten salir hasta que terminen.

Principio de consistencia y estándares

Una interfaz debe comportarse de manera uniforme y seguir las convenciones de la plataforma y del sector para que los usuarios puedan anticipar cómo interactuar con ella, evitando la confusión y facilitando el aprendizaje de nuevas funcionalidades al basarse en experiencias previas.

Ejemplos:

- Microsoft Office: mismo ícono de guardar en todas las apps.
- Semáforos: rojo siempre detenerse, verde avanzar.

Contraejemplo:

- Controles de aire acondicionado en coches: iconos diferentes entre marcas.

Principio de prevención de errores

Se refiere al diseño proactivo de un sistema para evitar que ocurran errores en primer lugar, en lugar de simplemente ayudar a los usuarios a recuperarse de ellos.

Ejemplos:

- Gmail: advierte si mencionas un adjunto sin incluirlo.
- Word: subraya errores ortográficos antes de confirmar.

Contraejemplo:

- Formularios en línea: borran toda la información al fallar el envío.

Principio de reconocimiento antes que recuerdo

Establece que es más fácil para los usuarios identificar elementos en una interfaz si ya los han visto antes que tener que recuperarlos de su memoria.

Ejemplos:

- Google: autocompleta términos en la barra de búsqueda.
- Spotify: muestra playlists recientes en la pantalla principal.

Contraejemplo:

- Microondas básicos: no muestran la potencia elegida.

Principio de flexibilidad y eficiencia de uso

Una interfaz debe ser usable tanto por usuarios novatos como experimentados, ofreciendo opciones y "aceleradores" (como atajos de teclado o macros) para que los usuarios avanzados realicen tareas rápidamente, sin sobrecargar a los novatos.

Ejemplos:

- Excel: menús y atajos de teclado para la misma acción.
- YouTube: adelantar con barra de progreso o la tecla →.

Contraejemplo:

- Mandos a distancia con demasiados botones sin atajos útiles.

Principio de diseño estético y minimalista

Consiste en crear productos y experiencias visualmente agradables mediante la eliminación de elementos innecesarios, el enfoque en la funcionalidad y la claridad de la información.

Ejemplos:

- Apple (iPhone): interfaz limpia con espacio en blanco.
- Google Search: página inicial simple con solo lo necesario.

Contraejemplo:

- Páginas web saturadas de publicidad emergente.

Principio de ayuda a users tras errores

Las interfaces de usuario notifican al usuario sobre un error, informan sobre él de manera clara y comprensible, y le ofrecen una solución o sugieren los pasos a seguir para solucionarlo, en lugar de causar frustración o confusión.

Ejemplos:

- Chrome: mensaje 'Sin conexión' con sugerencia de revisar red.
- Windows: usa papelera antes de borrar definitivamente.

Contraejemplo:

- Errores técnicos en páginas web: 'Error 500', sin explicación.

Principio de ayuda y documentación

Establece que los sistemas de software deben proporcionar documentación o ayuda fácil de encontrar y usar, pero sin que la aplicación necesite un manual para ser utilizada.

Ejemplos:

- Canva: tutoriales rápidos dentro de la interfaz.
- IKEA: manuales visuales con diagramas claros.

Contraejemplo:

- Electrodomésticos baratos: manuales mal traducidos y poco útiles.