

# Interfaces de usuario.

Los diseños deben ser adaptables a diferentes generaciones.

Problemas de un mal diseño:

- Accidentes. Hogareños y lesiones.
- Costo de soporte y entrenamiento. Mayor dificultad de aprendizaje y operación, más se necesita invertir en capacitar a los usuarios y en atender a sus reclamos.
- Menos usuarios.

No poner botones rojos a, por ejemplo “formatear sistema”. No es la mejor opción un botón llamativo para una acción tan radical y sin retorno.

**Usabilidad:** atributo de calidad. Cuan fácil es usar una interfaz.

Cinco atributos para medirlo:

1. qué tan fácil es aprenderlo. Learnability.
2. Qué tan eficiente es. Efficiency.
3. Si no uso la interfaz, qué tan fácil es volver a usarla. Memorability.
4. Cuántos errores cometen los usuarios. Error safe.
5. Qué tan satisfactorio es para mí. Satisfaction.

La satisfacción es lo más importante. Ejemplo: Tinder, app adictiva pero no está diseñado para usarlo por horas y horas (provoca, por ejemplo, tendinitis). Esta es una app usable.

Entonces ¿Para qué hacer productos usables?

- Reduce o elimina costos de soporte.
- Reduce o elimina costos de entrenamiento.
- Aumenta la productividad o eficiencia del usuario.
- Cumple con los estándares de calidad, de haberlos.
- Reduce la carga física y cognitiva o mental.
- Incrementa la satisfacción.

**UX user experience.** Estudia todos los aspectos de la interacción de un usuario con el servicio, producto o software. No se limita a la UI o parte visual solamente. Ejemplo pido un uber con una interfaz hermosa y totalmente claro, pero si el auto no me llega a los cinco minutos o es un auto que esta roto, en malas condiciones, no sirve de nada. UX estudia ambos aspectos.

**Accesibilidad.** Capacidad de un artefacto para ser utilizado por la mayor cantidad de gente, con o sin discapacidad y otras condiciones.

El diseño se adapta a la necesidad del usuario.

**Human-centered design.** Es un enfoque en un proceso de diseño que esta pensado para tomar lo que necesita un usuario dentro del producto.

1. Es iterativo. Empieza entendiendo la necesidad del usuario, se diseñan prototipos de baja resolución y se evalúa con usuarios, iterando hasta lograr un producto mejorado.
2. Involucra al usuario. Tiene que estar involucrado en el proceso. Sino sería authority-based design (diseño basado en la autoridad).

Research → design → evaluate → adapt. (y nuevamente a Research.)

**Design thinking.** Interfaces que generan empatía con el usuario. Plantea cinco fases que pueden no ser secuenciales: Empatiza, definir, idear, prototipar y testear.

1. Investigación con usuarios (UX Research) Cual es la tarea, limitaciones, deseos. Qué quiere hacer la gente y dónde está, entender en contexto. No debemos agregar nuestras percepciones a esto. Entrevistas, encuestas, diarios, etc.
2. Análisis de datos
3. Diseño de prototipos. En papel → luego mockups → de baja calidad.
4. Evaluación o prueba de usabilidad.

## Nielsen's Heuristics (1995)

Principios generales de diseño de interacción que nos permiten encontrar mejoras en un sistema o sitio web.

1. Visibilidad del estado del sistema. Tiene que mantener al usuario informado de lo que está pasando y dar una respuesta en tiempo razonable.
2. Conexión sistema-mundo real
3. Control de usuario y libertad. Soportar el hacer y el deshacer (undo) Los usuarios suelen elegir opciones por error y necesitan una salida delimitada para abandonar ese estado.
4. Consistencia y estándares.
5. Prevención de errores. Un diseño que reviene errores desde el inicio es mejor que unos buenos mensajes de error. Elimine condiciones que conducen errores o verifíquelas y preséntele a los usuarios una confirmación antes de ejecutar la acción. (Los checks al colocar una contraseña nueva)
6. Reconocimiento sobre memoria. Minimice el uso de memoria del usuario haciendo que objetos, acciones y opciones estén visibles. El usuario no debería recordar información de una interfaz en otra. Las instrucciones de uso deberían ser accesibles en el momento necesario (ejemplo el buscado de google que es predilectivo).
7. Flexibilidad y eficiencia de uso.
8. Estética y minimalismo.
9. Ayudar a usuarios a reconocer, diagnosticar y recuperarse de errores. Los mensajes de error deberían mostrarse en lenguaje plano, sin código, indicar el problema y sugerir una solución.
10. Ayuda y documentación. Aunque es mejor que el sistema sea usado sin documentación, puede ser necesario proveer ayuda. Debería ser fácil de buscar y enfocada en la tarea del usuario, listar los pasos concretos y no ser muy extensa.

## Recomendaciones de diseño.

Tipografía amigable con la lectura (Sans), usar máximo cinco colores en la paleta, no usar colores como única forma de informar (ejemplo, es algo meramente cultural y las personas con daltonismo no podrían percibirlo). Usar buen contraste. No usar carruseles.

Organización, el número de oro  $5 \pm 2$  para elementos de grupo, ejemplo menú separados en bloques (categorías), acoplar cosas comunes, resaltar cosas importantes. Reducir cantidad de opciones. Agrandar los elementos principales y debe estar cerca del lugar más cómodo que uno tiene para acceder. Disponer elementos importantes al principio y al final.