



Equipo:

Héctor Ulises Cacho González

Ciro Julián Cervantes Zamora

Dana Lizbeth Castañeda Sánchez

Concepcion Guadalupe Paniagua Gonzalez

Oscar Yael Hernández Rodríguez

Docente:

José Miguel Carrera Pacheco

Materia:

Desarrollo Web Profesional

Actividad:

S - 3. UX/UI profesional y accesibilidad

FECHA:

25 de enero de 2026

UX/UI y Accesibilidad

El desarrollo de interfaces web modernas trasciende la simple funcionalidad técnica; requiere una comprensión profunda de cómo los seres humanos interactúan con los sistemas digitales. Para el proyecto Rapiti, cuyo objetivo es facilitar la comparación de precios a usuarios con diversos niveles de alfabetización digital, la usabilidad no es un lujo, sino un requisito funcional crítico.

Este documento establece el marco teórico que regirá el diseño de la aplicación, basándose en tres pilares fundamentales: las Heurísticas de Jakob Nielsen, los Principios de Navegación de Steve Krug y los estándares internacionales de accesibilidad WCAG 2.1.

LAS 10 HEURÍSTICAS DE USABILIDAD DE JAKOB NIELSEN

Desarrolladas por Jakob Nielsen en los años 90, estas "reglas generales" siguen siendo el estándar de oro para la evaluación de interfaces. A continuación, se detalla cada una y su aplicación en nuestro proyecto.

1. Visibilidad del estado del sistema

El sistema siempre debe mantener a los usuarios informados sobre lo que está sucediendo, a través de retroalimentación adecuada en un tiempo razonable.

- Ejemplo General: Una barra de progreso al subir un archivo o un spinner girando al cargar resultados.
- Aplicación en Rapiti: Cuando el usuario busca "Leche", el botón de búsqueda cambiará de estado y se mostrará un indicador de carga (skeleton loader) antes de pintar las tarjetas de precios, evitando que el usuario piense que la app se trabó.

2. Relación entre el sistema y el mundo real

El sistema debe hablar el lenguaje de los usuarios con palabras, frases y conceptos familiares, en lugar de términos orientados al sistema.

- Ejemplo General: Usar un icono de "Bote de basura" para eliminar, en lugar de "Purgar base de datos".
- Aplicación en Rapiti: Utilizaremos metáforas físicas como "Tienda", "Anaquel" o "Mapa" e iconos universales (carrito, lupa, pin de ubicación) para reducir la curva de aprendizaje.

3. Control y libertad del usuario

Los usuarios eligen funciones del sistema por error y necesitan una "salida de emergencia" claramente marcada para dejar el estado no deseado.

- Ejemplo General: El botón de "Deshacer" (Ctrl+Z) o la capacidad de cerrar un modal emergente haciendo clic fuera de él.
- Aplicación en Rapiti: Si un usuario entra al detalle de un producto, siempre tendrá un botón visible de "Atrás" o una ruta de migas de pan (breadcrumbs) para regresar al listado sin perder su búsqueda.

4. Consistencia y estándares

Los usuarios no deberían tener que preguntarse si diferentes palabras, situaciones o acciones significan lo mismo. Se deben seguir las convenciones de la plataforma.

- Ejemplo General: El logo de la empresa, ubicado arriba a la izquierda, siempre debe llevar al inicio (Home).
- Aplicación en Rapiti: Mantendremos el mismo Header y Footer en todas las pantallas. Los botones de acción primaria siempre serán de color Verde Neón, mientras que los secundarios serán grises.

5. Prevención de errores

Mucho mejor que un buen mensaje de error es un diseño cuidadoso que evite que ocurra el problema en primer lugar.

- Ejemplo General: Sugerencias de búsqueda de Google que corrigen tu ortografía mientras escribes.
- Aplicación en Rapiti: En el buscador, implementaremos autocompletado. Si el usuario escribe "Lche", el sistema sugerirá "¿Quisiste decir: Leche?", evitando una búsqueda vacía.

6. Reconocer antes que recordar

Minimizar la carga de memoria del usuario haciendo visibles los objetos, acciones y opciones. El usuario no debería tener que recordar información de una parte del diálogo a otra.

- Ejemplo General: Un menú de navegación visible en lugar de comandos de terminal ocultos.
- Aplicación en Rapiti: Mostraremos las "Categorías Populares" con iconos en el Home, para que el usuario pueda hacer clic en "Farmacia" sin tener que recordar o escribir la palabra.

7. Flexibilidad y eficiencia de uso

Los aceleradores, no vistos por el usuario novato, a menudo pueden acelerar la interacción para el usuario experto.

- Ejemplo General: Atajos de teclado (Ctrl+C, Ctrl+V) en aplicaciones de escritorio.
- Aplicación en Rapiti: Permitir que al presionar la tecla "Enter" se ejecute la búsqueda, sin obligar al usuario a hacer clic manualmente en el botón de la lupa.

8. Estética y diseño minimalista

Los diálogos no deben contener información que sea irrelevante o rara vez necesaria. Cada unidad extra de información compite con las unidades relevantes y disminuye su visibilidad relativa.

- Ejemplo General: La página de inicio de Google, que es casi totalmente blanca con solo el buscador.
- Aplicación en Rapiti: En las tarjetas de producto, solo mostraremos: Nombre, Precio y Tienda. Ocultaremos datos técnicos irrelevantes (como el SKU interno o fecha de alta) para no saturar visualmente.

9. Ayudar a los usuarios a reconocer, diagnosticar y recuperarse de errores

Los mensajes de error deben expresarse en lenguaje sencillo (sin códigos), indicar con precisión el problema y sugerir constructivamente una solución.

- Ejemplo General: "La contraseña es incorrecta" vs "Error 403: Forbidden".
- Aplicación en Rapiti: Si no hay resultados para una búsqueda, diremos: "No encontramos ese producto cerca de ti. Intenta buscar por categoría", en lugar de dejar la pantalla en blanco.

10. Ayuda y documentación

Aunque es mejor si el sistema se puede usar sin documentación, puede ser necesario proporcionar ayuda y documentación.

- Ejemplo General: Una sección de Preguntas Frecuentes (FAQ) o un tour guiado la primera vez que abres la app.
- Aplicación en Rapiti: Incluiremos tooltips (textos de ayuda) breves en iconos que puedan ser confusos, explicando su función al pasar el mouse.

PRINCIPIOS DE STEVE KRUG: "DON'T MAKE ME THINK"

Steve Krug, en su obra seminal sobre usabilidad web, establece que la primera ley de la usabilidad es: "No me hagas pensar". Esto significa que una página web debe ser evidente, obvia y autoexplicativa.

Principio 1: La Obviedad

Cuando un usuario mira una página web, no debería tener que descifrar qué es o cómo usarla.

- **Aplicación:** En Rapiti, el buscador es el elemento central y más grande de la pantalla de inicio. No hay duda de que la función de la página es "Buscar cosas". Los botones parecen botones (tienen bordes y sombras o colores sólidos) y los enlaces se distinguen del texto normal.

Principio 2: Escaneamos, no leemos

Los usuarios web no leen el contenido palabra por palabra; escanean la página buscando palabras clave o anclas visuales que coincidan con su tarea actual.

- **Aplicación:**
 - Uso amplio de encabezados (H1, H2, H3) para jerarquizar información.
 - Listas con viñetas (bullet points) en lugar de párrafos densos.
 - Uso de iconos junto a las categorías de productos para acelerar el reconocimiento visual.

Principio 3: Omitir palabras innecesarias

Krug sugiere eliminar la mitad de las palabras de cada página y luego eliminar la mitad de lo que queda. El "ruido" textual reduce la usabilidad.

- **Aplicación:** En lugar de poner "Bienvenido a Rapiti, el mejor sistema para buscar precios en tu localidad, por favor ingrese su búsqueda aquí", simplemente pondremos: "¿Qué necesitas hoy?".

CRITERIOS WCAG 2.1 (NIVEL A)

Las Pautas de Accesibilidad para el Contenido Web (WCAG) son estándares técnicos para garantizar que las personas con discapacidades puedan navegar en la web. Para Rapiti, nos enfocamos en el Nivel A (esencial) y AA.

4.1 Contraste (Criterio 1.4.1 y 1.4.3)

El texto y las imágenes de texto deben tener una relación de contraste de color de al menos 4.5:1 respecto al fondo.

- Implementación: Se ha elegido un esquema de "Modo Oscuro".
 - Fondo: Gris Muy Oscuro (#121212)
 - Texto Principal: Blanco Puro (#FFFFFF) → Ratio 15.6:1 (Excelente).
 - Texto Secundario: Gris Claro (#AAAAAA) → Ratio 7.5:1 (Pasa AAA).
 - Acción: Verde Neón (#2EFA2) sobre fondo oscuro → Ratio 11.5:1. Esto garantiza legibilidad incluso para personas con baja visión o daltonismo.

4.2 Navegación y Estructura (Criterio 1.3.1 y 2.4.1)

La información y estructura deben poder ser determinadas por software (lectores de pantalla).

- Uso de HTML Semántico: No usaremos solo <div> para todo. La estructura de Rapiti utiliza etiquetas significativas:
 - <header> para la cabecera.
 - <nav> para el menú de navegación.
 - <main> para el contenido principal.
 - <footer> para el pie de página.
 - <button> para acciones y <a> para enlaces.
- Navegación por Teclado: Se garantizará que el "orden de foco" sea lógico, permitiendo a un usuario navegar todo el sitio usando solo la tecla TAB.

4.3 Texto Alternativo (Criterio 1.1.1)

Todo contenido no textual (imágenes, iconos, gráficos) debe tener una alternativa textual.

- Implementación:
 - Las imágenes de productos incluirán el atributo alt="Foto de cartón de leche Alpura".
 - Los iconos decorativos (flechas, adornos) tendrán aria-hidden="true" para que los lectores de pantalla los ignoren y no confundan al usuario ciego.

- El mapa interactivo tendrá una etiqueta aria-label="Mapa de ubicación de tiendas" y controles accesibles por teclado.

ERRORES COMUNES DE UX Y CÓMO EVITARLOS

Error 1: Falta de Feedback (Retroalimentación)

El usuario hace clic en un botón y no pasa nada visiblemente. Vuelve a hacer clic múltiples veces, causando errores o pagos dobles.

- Solución: Implementar estados de carga (spinners) y deshabilitar el botón inmediatamente después del primer clic (`disabled={true}`) hasta que la acción se complete.

Error 2: Áreas de clic (Touch Targets) demasiado pequeñas

Especialmente en móviles, poner enlaces o botones diminutos hace que sea difícil tocar el correcto con el dedo ("Problema de los dedos gordos").

- Solución: Seguir la guía de iOS y Android de áreas táctiles mínimas de 44x44 píxeles (o 48dp). Daremos suficiente *padding* (relleno) a los botones.

Error 3: Formularios extensos e intrusivos

Pedir demasiados datos antes de permitir usar la aplicación provoca abandono.

- Solución: Permitir la búsqueda y comparación de precios sin necesidad de registrarse. El registro solo se pedirá si el usuario quiere guardar favoritos, y solo pediremos Email y Contraseña.

Error 4: Diseño no responsivo (Horizontal Scroll)

Que el sitio web se vea "cortado" en el celular o requiera hacer scroll horizontal para leer textos.

- Solución: Utilizar diseño fluido con Flexbox y CSS Grid. Las tarjetas de productos se apilarán verticalmente en móviles (1 columna) y se expandirán en escritorio (3 o 4 columnas).