

Memoria de Evaluación y Mejoras – Editor de Texto

Fase 1: Evaluación de la interfaz

a) Heurísticas de Nielsen (5 problemas identificados)

Problema	Heurística violada	Impacto en la experiencia de usuario
No hay retroalimentación inmediata al deshacer cambios	Visibilidad del estado del sistema	El usuario no sabe si el cambio se ha revertido hasta que lo comprueba manualmente.
Botones de operaciones no muestran descripciones fácilmente	Reconocer en lugar de recordar	El usuario debe adivinar qué hace cada botón, aumentando el tiempo para aprender a usar la aplicación.
No hay tutorial inicial ni guía para nuevas funciones	Ayuda y documentación	Usuarios nuevos pueden no descubrir atajos (Ctrl+Z, buscar, reemplazar) o funcionalidades del menú contextual.
Acciones de copiar, cortar y pegar no accesibles desde el teclado	Control y libertad del usuario	Usuarios que prefieren teclado o no usan ratón se ven limitados, afectando la operabilidad.
No se indica el estado de los cambios o de operaciones de guardado	Visibilidad del estado del sistema	El usuario no sabe si el archivo se ha guardado correctamente hasta que aparece el alert; no hay feedback visual constante.

b) Accesibilidad (WCAG 2.2)

Evaluación	Problema / Mejora sugerida
Contraste texto/fondo	El contraste actual de <code>TextArea</code> y <code>Label1</code> es adecuado, pero podría mejorarse usando colores más oscuros para textos de estado.
Operabilidad con teclado	Ahora <code>Ctrl+Z</code> permite deshacer; se podrían añadir atajos para copiar, cortar y pegar y para abrir/guardar.

Visibilidad del foco	El foco en <code>TextField</code> y <code>TextArea</code> funciona, pero no se resalta con borde o cambio de color marcado; mejorar visualmente el foco.
Uso del color	Actualmente, algunos estados se muestran solo con color (rojo/azul) en el texto; se recomienda combinar con iconos o texto para accesibilidad.
Claridad de mensajes	Los mensajes de estado (<code>lblStatus</code>) y tutorial ayudan; alertas de error podrían ser más descriptivas.

Problemas o mejoras posibles:

1. Mejorar contraste y visibilidad del foco en componentes.
 2. Añadir atajos de teclado para todas las acciones.
 3. Evitar depender solo del color para transmitir información.
-

c) Checklist RA4 (resumen)

Punto	Cumple / No cumple
Botones con texto claro y visible	Cumple razonablemente
Retroalimentación de operaciones	Parcialmente cumple (alertas, <code>lblStatus</code>)
Uso de atajos de teclado	Parcialmente cumple (solo Ctrl+Z, Ctrl+F, Ctrl+H)
Accesibilidad del teclado	Parcialmente cumple
Tutorial o ayuda	Cumple (tutorial inicial)
Contraste suficiente	Cumple parcialmente
Iconos y texto combinados	No cumple (solo texto, sin iconos)

Fase 2: Propuestas de mejora

Problema	Solución propuesta	Justificación
Falta de guía para nuevas funciones	Tutorial inicial actualizado y alertas explicativas	Mejora la heurística “Ayuda y documentación” y facilita el reconocimiento de funciones sin recordar.

Acciones de copiar, cortar y pegar limitadas	Añadir menú contextual y atajos de teclado (Ctrl+C, Ctrl+X, Ctrl+V)	Cumple “Control y libertad del usuario” y mejora la operabilidad con teclado.
Estado de cambios y deshacer no claro	Mini-historial de cambios y botón ‘Deshacer’ + Ctrl+Z	Mejora “Visibilidad del estado del sistema”, ya que el usuario puede revertir cambios fácilmente y ver que se aplica.
Mensajes de estado poco claros	lblStatus dinámico con descripción de menús y operaciones	Facilita la heurística “Reconocer en lugar de recordar” y la accesibilidad, dando información textual adicional.
Tutorial inicial incompleto	Incluir todas las funcionalidades implementadas (deshacer, menú contextual, mini-historial)	Ayuda a nuevos usuarios y cumple WCAG 2.2 en claridad de mensajes.

Fase 3: Implementación de mejoras

Mejoras aplicadas:

1. Tutorial inicial actualizado:

- Explica Ctrl+Z/deshacer, menú contextual (copiar/cortar/pegar) y mini-historial.
- Impacto: usuarios nuevos entienden rápidamente el funcionamiento del editor.

2. Mini-historial y deshacer:

- Cada cambio se guarda automáticamente.
- Botón “Deshacer” y Ctrl+Z permiten revertir cambios.
- Impacto: mejora la experiencia al corregir errores y aumenta control del usuario.

3. Menú contextual:

- Botón derecho permite copiar, cortar y pegar.
- Impacto: mejora la accesibilidad y facilita la interacción sin depender del teclado.

Notas técnicas:

- Implementado en `EditorController` con `Stack<String>` para historial.
 - Ctrl+Z y botón ‘Deshacer’ conectados al mismo método.
 - Tutorial mostrado con `Alert` al iniciar.
-

Fase 4: Reevaluación

Las mejoras implementadas han resuelto los problemas iniciales más relevantes:

- **Mini-historial y Ctrl+Z:** ahora los usuarios pueden revertir cambios, cumpliendo “Visibilidad del estado del sistema” y “Control y libertad del usuario”.
- **Menú contextual y tutorial:** facilitan la operabilidad sin ratón y el aprendizaje de la interfaz, mejorando “Reconocer en lugar de recordar” y criterios WCAG de claridad y operabilidad.
- La experiencia de usuario es más fluida, intuitiva y accesible para nuevos usuarios y usuarios de teclado.