



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA EN SOFTWARE
ISWD633

ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
CONSTRUCCIÓN Y EVOLUCIÓN DE SOFTWARE



ESCUELA
POLITÉCNICA
NACIONAL

Proyecto Segundo Bimestre
Transcriptor a Braille vs 2.0

EQUIPO 3:

Sara Guayasamin

Roberth Jácome

Danna Morales

Salma Morales

Profesor:

Evelyn Marcela Mosquera Espinosa

Fecha de entrega:

30/01/2026



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA EN SOFTWARE
ISWD633

INDICE

1. Objetivo General para esta Iteración.....	3
2. Información General del Cambio.....	3
3. Descripción del Cambio.....	3
4. Justificación del Cambio.....	4
5. Análisis de Impacto.....	4
6. Elementos de Configuración Afectados.....	5
7. Riesgos Asociados al Cambio.....	5
8. Implementación del Cambio	5
9. Plan de Pruebas del Cambio	6



Análisis de Cambios

Requisito 3: Transcribir textos de Braille a español

1. Objetivo General para esta Iteración

Identificar y describir las modificaciones necesarias en el código actual para cumplir con los nuevos requisitos de la segunda iteración, los cuales son: Traducción de Braille a español y Generación de impresión en espejo (Para facilitar escritura manual).

2. Información General del Cambio

Campo	Descripción
ID del Cambio	AC-01
Nombre del Cambio	Implementación de transcripción de Braille a Español
Tipo de Cambio	Nuevo requerimiento funcional
Requisito Relacionado	RF-03 – Transcripción de Braille a Español
Fecha	Enero 2026
Autor(es)	Equipo de Desarrollo
Versión	1.0

3. Descripción del Cambio

En la primera iteración del sistema se implementó la funcionalidad de transcripción de texto desde español hacia Braille, permitiendo generar textos en sistema braille, incluyendo letras, números, vocales acentuadas y signos básicos.

En la segunda iteración se requiere incorporar la funcionalidad inversa, permitiendo la transcripción de textos escritos en sistema Braille hacia texto en español donde se implementará, con las teclas q, a, z, w, s, x, un teclado braille que representa la regleta para escribir.



Este cambio implica la incorporación de nuevos módulos de procesamiento que permitan interpretar combinaciones de puntos braille y transformarlas en caracteres del alfabeto español, números y signos básicos de puntuación.

4. Justificación del Cambio

La implementación de la transcripción de Braille a Español permite ampliar la funcionalidad del sistema, convirtiéndolo en una herramienta bidireccional para la transcripción.

Este cambio es necesario para:

- Facilitar la comprensión de textos en braille por parte de personas sin discapacidad visual.
- Apoyar procesos educativos relacionados con el aprendizaje del sistema braille.
- Permitir la validación de textos braille generados por el sistema.
- Contribuir a la inclusión social mediante herramientas tecnológicas accesibles.

Además, fortalece el objetivo del proyecto de fomentar el conocimiento del sistema de lectura y escritura braille, promoviendo así entornos más inclusivos.

5. Análisis de Impacto

Área	Impacto	Descripción
Requisitos	Alto	Se incorpora un nuevo requerimiento funcional de transcripción inversa.
Código Fuente	Alto	Se implementan nuevos métodos de conversión braille a texto.
Interfaz de Usuario	Medio	Se agrega opción para seleccionar tipo de transcripción. Se agrega teclado para la escritura en Braille.
Casos de Prueba	Alto	Se deben crear nuevos casos de prueba y validar precisión de traducción.
Documentación	Alto	Se deben actualizar el manual técnico y el manual de usuario.



6. Elementos de Configuración Afectados

Los siguientes elementos del sistema deben ser actualizados:

- Código fuente relacionado a la transcripción.
- Diccionarios o tablas de equivalencia Braille – Español.
- Interfaz de usuario relacionada a la transcripción.
- Casos de prueba funcionales.
- Documentación técnica del sistema.
- Manual de usuario.

7. Riesgos Asociados al Cambio

Riesgo	Nivel	Mitigación
Traducciones incorrectas de símbolos braille	Alto	Validar con tablas oficiales de braille.
Errores en caracteres especiales	Medio	Implementar pruebas unitarias específicas.
Incompatibilidad con traducción existente Español → Braille	Bajo	Mantener módulos separados.

8. Implementación del Cambio

Para cumplir con el requisito Transcribir textos de Braille a español se debe de invertir la lógica actual que solo funciona en una dirección.

8.1. Cambios en la lógica del negocio:

Creación de un Mapa Inverso: El código tiene un Map. Por lo que se necesita crear un Map para buscar el carácter en español a partir del símbolo Braille.

Nuevo Método: Se debe procesar la cadena símbolo por símbolo (cada carácter Unicode de Braille).



Manejo de Prefijos: Debe detectar el signo numeral (:) para convertir los siguientes símbolos en números, y el signo de mayúscula (:) para la siguiente letra.

Lógica de Estado: Se debe de implementar una pequeña "máquina de estados" que sepa si el símbolo actual debe interpretarse como letra, número o mayúscula según el prefijo anterior.

8.2.Cambios en la API Nuevo Endpoint:

Crear un endpoint POST /traducir-inverso. El cual recibirá la cadena de símbolos Braille y llamará al nuevo método del traductor.

8.3.Cambios en la Interfaz:

Selector de Modo: Añadir un botón o toggle para cambiar entre "Español → Braille" y "Braille → Español". Habilitar Entrada: El *textArea* de Braille (que actualmente es *readOnly*) debe permitir entrada de texto si se selecciona el modo inverso.

Checkbox para activar Teclado Braille: Se debe implementar un checkbox para probar el teclado braille, que indique cuando está activa esta función. Además de poner un texto informativo sobre las teclas que actuarán como regleta. “Presiona las teclas Q/A/Z/W/S/X (puntos 1-6) para escribir en braille”

9. Plan de Pruebas del Cambio

Se implementarán pruebas para validar la correcta transcripción de:

- Letras del alfabeto
- Vocales acentuadas
- Números
- Palabras y frases completas

Criterios de aceptación:

- La conversión braille a español debe tener una precisión del 100% en los casos definidos.
- El sistema debe procesar textos sin generar errores de ejecución.
- La interfaz debe mostrar correctamente los resultados.