



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 Primer Bimestre
Transcriptor a Braille vs 1.0

EQUIPO 3:

Sara Guayasamin
Roberth Jácome
Danna Morales
Salma Morales

Profesor:

Evelyn Marcela Mosquera Espinosa

Fecha de entrega:

25/11/2025



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

Casos de Prueba

Caso de Prueba 1

ID del Caso de Prueba	Prueba CP-001
Título	Conversión básica de texto español a braille mediante endpoint REST
Descripción	Verificar que el sistema convierte correctamente texto en español a su representación en braille cuando se envía una petición POST al endpoint /traducir. El sistema debe procesar el texto y devolver la respuesta en formato JSON con el texto traducido.
Precondiciones	<ol style="list-style-type: none">1. El servidor está ejecutándose en http://localhost:80812. El endpoint /traducir está disponible y responde3. La base de datos o diccionario de traducción braille está cargado4. No hay errores en los logs del servidor
Pasos	<ol style="list-style-type: none">1. Abrir cliente HTTP (Postman, Insomnia, curl o navegador con extensión REST)2. Configurar método POST hacia http://localhost:8081/traducir3. Establecer header Content-Type: text/plain4. En el body de la petición, escribir el texto: "hola mundo"5. Enviar la petición6. Observar la respuesta HTTP
Datos de Prueba	Entrada: <ul style="list-style-type: none">- Texto: "hola mundo"- Content-Type: text/plain- Método: POST- URL: http://localhost:8081/traducir



Código de la Prueba CP-001:

```
@Test
@DisplayName("Test 1: Conversión de letras básicas en minúsculas")
void testLetrasBasicas() {
    String resultado = translator.textToBraille("hola");
    String esperado = "⠠⠏⠗⠠⠇⠠⠁⠠⠎⠠";
    assertEquals(esperado, resultado, "Las letras 'hola' deben convertirse correctamente a Braille");
}
```



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

Caso de Prueba 2

ID del Caso de Prueba	CP-002
Título	Validación de entrada vacía o nula en endpoint de traducción
Descripción	Verificar que el sistema maneja correctamente casos de entrada vacía, nula o con solo espacios en blanco sin generar errores internos del servidor. El sistema debe responder de manera controlada y predecible ante datos de entrada inválidos o vacíos.
Precondiciones	1. El servidor está ejecutándose en http://localhost:8081 2. El endpoint /traducir está disponible 3. El sistema tiene implementada validación de entrada 4. No hay solicitudes pendientes que puedan interferir
Pasos	1. Abrir cliente HTTP (Postman, curl, etc.) 2. Configurar método POST hacia http://localhost:8081/traducir 3. Establecer header Content-Type: text/plain 4. Prueba A: Enviar body vacío (cadena "") 5. Observar respuesta 6. Prueba B: Enviar body con solo espacios (" ") 7. Observar respuesta 8. Verificar que el servidor no se detiene ni genera error 500
Datos de Prueba	Caso A: - Texto: "" (cadena vacía) - Content-Type: text/plain - Método: POST Caso B: - Texto: " " (3 espacios) - Content-Type: text/plain - Método: POST
Resultado Esperado	Código de estado: 200 OK o 400 Bad Request (según especificación) Content-Type: application/json Body de respuesta: - Opción 1: Cadena vacía "" - Opción 2: Mensaje de error controlado como {"error": "Entrada vacía"} Comportamiento: El servidor NO debe devolver error 500 ni crash Logs: Sin excepciones no controladas
Resultado Obtenido	(A completar durante la ejecución) Caso A (vacío): - Código HTTP: ____ - Respuesta: ____ Caso B (espacios): - Código HTTP: ____ - Respuesta: ____ - Estado: <input type="checkbox"/> Aprobado <input type="checkbox"/> Fallido



```
@Test
@DisplayName("Test 2: Conversión de vocales acentuadas")
void testVocalesAcentuadas() {
    String resultado = translator.textToBraille(text: "café");
    String esperado = "⠠⠉⠠⠋⠠⠑⠠⠋⠠⠋";
    assertEquals(esperado, resultado, message: "La palabra 'café' con acento debe convertirse correctamente");
}
```

ID del Caso de Prueba	CP-003
Título	Conversión de texto con caracteres especiales, acentos, números y signos de puntuación
Descripción	Verificar que el sistema traduce correctamente texto español que incluye vocales acentuadas (á, é, í, ó, ú), la letra ñ, signos de puntuación (¡, ¿, comas, puntos), números y caracteres especiales. El sistema debe mantener la codificación UTF-8 y no perder ni corromper caracteres durante la traducción.
Precondiciones	<ol style="list-style-type: none"> 1. El servidor está ejecutándose en http://localhost:8081 2. El endpoint /traducir acepta codificación UTF-8 3. El diccionario de traducción incluye caracteres acentuados, ñ, números y puntuación 4. La configuración del servidor soporta charset UTF-8
Pasos	<ol style="list-style-type: none"> 1. Abrir cliente HTTP con soporte UTF-8 2. Configurar método POST hacia http://localhost:8081/traducir 3. Establecer headers: <ul style="list-style-type: none"> - Content-Type: text/plain; charset=UTF-8 - Accept-Charset: UTF-8 4. En el body, escribir el texto de prueba con caracteres especiales 5. Enviar la petición 6. Verificar que la respuesta mantiene la integridad de caracteres 7. Revisar que no aparecen caracteres corruptos (¿, ?, etc.)
Datos de Prueba	<p>Entrada:</p> <ul style="list-style-type: none"> - Texto: "¡Hola, señor! Año 2025. ¿Cómo estás?" - Content-Type: text/plain; charset=UTF-8 - Método: POST - URL: http://localhost:8081/traducir <p>Caracteres a probar:</p> <ul style="list-style-type: none"> - Signos de apertura: ¡, ¿ - Vocales acentuadas: ó, ñ, á - Números: 2025 - Puntuación: comas, puntos, interrogación, exclamación



Código de la Prueba CP-003:

Caso de Prueba 4

ID del Caso de Prueba	CP-004
Título	Conversión de texto mixto (letras, números y espacios).
Descripción	Verificar que el sistema traduce correctamente un texto que mezcla letras minúsculas, números y espacios. El sistema debe aplicar las reglas de braille para letras y números, incluyendo el indicador numérico antes de secuencias de dígitos. También debe respetar los espacios y mantener la secuencia completa sin omitir caracteres.
Precondiciones	<ol style="list-style-type: none"> 1. El servidor está ejecutándose en: <code>http://localhost:8081</code> 2. El endpoint <code>/traducir</code> se encuentra disponible 3. El diccionario de traducción incluye: <ul style="list-style-type: none"> • Letras minúsculas a braille



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

	<ul style="list-style-type: none">• Indicador numérico ∴• Dígitos del 0 al 9 <p>4. La traducción debe usar codificación UTF-8</p> <p>5. El método interno textToBraille() está operativo y accesible</p>
Pasos	<p>1. Abrir el cliente HTTP o ejecutar el método interno textToBraille()</p> <p>2. Preparar la entrada: "hola 123 mundo"</p> <p>3. Enviar la solicitud (HTTP POST o llamada directa al método)</p> <p>4. Verificar que:</p> <ul style="list-style-type: none">• Las letras se traduzcan correctamente• El indicador numérico ∴ aparece antes de la secuencia 123• Los espacios se preservan• No se omiten ni alteran caracteres <p>5. Comparar la respuesta con el valor esperado</p> <p>6. Validar que el resultado esté en formato UTF-8</p>
Datos de Prueba	<p>Entrada: Texto: "hola 123 mundo"</p> <p>Caracteres a probar:</p> <ul style="list-style-type: none">• Letras minúsculas: h, o, l, a, m, u, n, d, o• Números: 1, 2, 3• Indicador numérico: ∴• Espacios: " " (2 espacios en total)
Resultado Esperado	<p>1. Código de estado:</p> <ul style="list-style-type: none">• HTTP 200 OK (si se prueba por API)• Ejecución sin errores (si se prueba por método directo) <p>2. Traducción:</p> <ul style="list-style-type: none">• Letras traducidas correctamente• Secuencia numérica precedida por ∴• Los números convertidos en: 1→ˆ, 2→˙, 3→¨• Espacios preservados <p>3. Encoding: UTF-8 correcto</p> <p>4. No deben aparecer caracteres corruptos, vacíos o reemplazados</p>
Resultado Obtenido	<p>(A completar durante la ejecución)</p> <ul style="list-style-type: none">• Código HTTP / Estado: ____• Respuesta / Traducción: ____• ¿Letras correctas? <input type="checkbox"/> Sí <input type="checkbox"/> No• ¿Números correctos? <input type="checkbox"/> Sí <input type="checkbox"/> No• ¿Espacios preservados? <input type="checkbox"/> Sí <input type="checkbox"/> No• ¿Caracteres corruptos? <input type="checkbox"/> Sí <input type="checkbox"/> No• Tiempo: ____ ms• Estado: <input type="checkbox"/> Aprobado <input type="checkbox"/> Fallido



```
@Test
@DisplayName("Test 4: Conversión de texto mixto (letras, números y espacios)")
void testTextoMixto() {
    String resultado = translator.textToBraille(text: "hola 123 mundo");
    String esperado = "⠁⠊⠋⠗⠎ ⠠⠏⠑⠗⠇⠆ ⠠⠍⠓⠗⠝⠔";
    assertEquals(esperado, resultado, message: "El texto mixto debe manejar letras, números y espacios co");
}
```

ID del Caso de Prueba	CP-005
Título	Conversión de signos de puntuación.
Descripción	Validar que el sistema realice correctamente la traducción de un texto que contiene signos de puntuación, específicamente la apertura de interrogación “¿” y el cierre de interrogación “?”. El sistema debe mapear correctamente estos símbolos a sus equivalentes en braille y no alterar la secuencia de letras o el orden de los caracteres.
Precondiciones	<ol style="list-style-type: none"> 1. El servidor está ejecutándose en: http://localhost:8081 2. El endpoint /traducir está disponible en modo UTF-8 3. El diccionario del sistema incluye equivalentes braille para: <ul style="list-style-type: none"> ○ ¿ → ⠏. ○ ? → ⠕. 4. La función interna textToBraille() está operativa 5. El sistema reconoce y procesa signos de puntuación iniciales y finales
Pasos	<ol style="list-style-type: none"> 1. Preparar el texto de entrada: "¿hola?" 2. Realizar una solicitud POST al endpoint /traducir o invocar directamente el método translator.textToBraille(). 3. Verificar que el sistema: <ul style="list-style-type: none"> Traduce “¿” como ⠏. Traduce cada letra: h, o, l, a Traduce “?” como ⠕. 4. Confirmar que el orden de los caracteres no cambia 5. Verificar que no se pierden caracteres ni se generan símbolos corruptos 6. Comparar la salida con el valor esperado
Datos de Prueba	Entrada: Texto: "¿hola?" Caracteres a probar: <ul style="list-style-type: none"> • Signo “¿” → debe traducirse a ⠏. • Letras: h, o, l, a



Código de la Prueba CP-005:

Caso de Prueba 6

ID del Caso de Prueba	CP-006
Título	Manejo de entrada vacía o nula.
Descripción	Verificar que el sistema maneja correctamente entradas inválidas o sin contenido, incluyendo cadenas vacías (""), valores nulos (null) y cadenas compuestas únicamente por espacios. El sistema debe retornar una cadena vacía en todos los casos sin generar errores, excepciones o caracteres inesperados.
Precondiciones	<ol style="list-style-type: none"> 1. El servidor está ejecutándose correctamente en http://localhost:8081 (si se prueba por API). 2. El endpoint <code>/traducir</code> acepta peticiones en UTF-8.



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

	<p>3. El método interno <code>textToBraille()</code> está disponible y no debe lanzar excepciones ante entradas nulas.</p> <p>4. El sistema incluye validaciones de sanitización: trimming, verificación de nulidad y chequeo de longitud.</p>
Pasos	<p>1. Enviar o procesar una cadena vacía ""</p> <p>2. Enviar o procesar una entrada nula null</p> <p>3. Enviar o procesar una cadena con solo espacios " "</p> <p>4. Observar la salida generada por el sistema</p> <p>5. Comprobar que en los tres casos el resultado sea exactamente una cadena vacía</p> <p>6. Verificar que no existan errores, excepciones o rastros de caracteres de control</p> <p>7. Asegurar que el encoding UTF-8 se mantiene</p> <p>8. Comparar el resultado con lo esperado</p>
Datos de Prueba	<p>Entradas:</p> <ul style="list-style-type: none">• "" (cadena vacía)• null (valor nulo)• " " (tres espacios)
Resultado Esperado	<ul style="list-style-type: none">• Salida exacta: "" en las tres pruebas• No debe generarse:<ul style="list-style-type: none">○ Excepciones○ Símbolos corruptos (☒)○ Espacios residuales○ Traducción accidental• La función debe aplicar correctamente limpieza de entrada (trim + null-handling)• Flujo de ejecución estable y controlado
Resultado Obtenido	<p>(A completar durante la ejecución)</p> <ul style="list-style-type: none">• Código HTTP / Estado: ____• Resultado para "": ____• Resultado para null: ____• Resultado para " ": ____• ¿Se generó alguna excepción? <input type="checkbox"/> Sí <input type="checkbox"/> No• ¿El retorno fue vacío en los tres casos? <input type="checkbox"/> Sí <input type="checkbox"/> No• Tiempo de respuesta: ____ ms• Estado: <input type="checkbox"/> Aprobado <input type="checkbox"/> Fallido

Código de la Prueba CP-006:

```
@Test
@DisplayName("Test 6: Manejo de entrada vacía o nula")
void testEntradaVacíaONula() {
    assertEquals(expected: "", translator.textToBraille(text: ""), message: "Una cadena vacía debe retornar va
    assertEquals(expected: "", translator.textToBraille(text: null), message: "Una entrada nula debe retornar
    assertEquals(expected: "", translator.textToBraille(text: " "), message: "Una cadena con solo espacios d
}
```



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

Casos de Prueba Exitosos:

Test Runner for Java

- ✓ Test 1: Conversión de letras básicas en minúsculas `$(symbol-class) BrailleTranslatorTest < $(symbol-namespace) <Default Package> < $(project) braille-app`
- ✓ Test 5: Conversión de signos de puntuación `$(symbol-class) BrailleTranslatorTest < $(symbol-namespace) <Default Package> < $(project) braille-app`
- ✓ Test 6: Manejo de entrada vacía o nula `$(symbol-class) BrailleTranslatorTest < $(symbol-namespace) <Default Package> < $(project) braille-app`
- ✓ Test 2: Conversión de vocales acentuadas `$(symbol-class) BrailleTranslatorTest < $(symbol-namespace) <Default Package> < $(project) braille-app`
- ✓ Test 3: Conversión de números con prefijo numérico `$(symbol-class) BrailleTranslatorTest < $(symbol-namespace) <Default Package> < $(project) braille-app`
- ✓ Test 4: Conversión de texto mixto `$(symbol-class) BrailleTranslatorTest < $(symbol-namespace) <Default Package> < $(project) braille-app`
- 🔗 `BrailleTranslatorTest $(symbol-namespace) <Default Package> < $(project) braille-app`