



ICCD533 CONSTRUCCIÓN Y EVOLUCIÓN DE SOFTWARE

INTEGRANTES

Julio Arrobo
Ángel Anguaya
Juan Cofre
Jhordy Parra

FECHA:

23/11/2025

Contenido

1	INTRODUCCIÓN	3
1.1	Objetivo del documento.....	3
1.2	Alcance	3
1.3	Relación con documentos del proyecto	3
1.4	Objetivo de las pruebas.....	3
2	PLAN DE PRUEBAS	4
2.1	Objetivos de pruebas.....	4
2.2	Alcance del plan.....	4
2.3	Estrategia de pruebas	5
2.4	Entorno de pruebas	5
2.5	Criterios	5
2.6	Matriz de trazabilidad	6
3	CASOS DE PRUEBA	6
3.1	Casos de prueba funcionales (RF01)	6
3.1.1	CP01 – Conversión de letras y palabras simples.....	6
3.1.2	CP02 – Vocales acentuadas	7
3.1.3	CP03 – Letras especiales (ñ, ü)	7
3.1.4	CP04 – Números y texto mixto.....	7
3.2	Casos de prueba funcionales (RF02)	8
3.2.1	CP05 – Generación de señalética simple	8
3.2.2	CP06 – Generación de señalética con números	8
3.3	Casos de error / validaciones (RNF04 – Fiabilidad).....	9
3.3.1	CP07 – Entrada vacía.....	9
3.3.2	CP08 – Caracteres no soportados.....	9
3.3.3	CP09 – Texto extenso	10
4	RESULTADOS DE EJECUCIÓN	10



ICCD533 CONSTRUCCIÓN Y EVOLUCIÓN DE SOFTWARE

4.1	Resultados iniciales.....	10
4.2	Análisis de fallos.....	13
4.3	Correcciones aplicadas	13
4.4	Re-ejecución	14
4.5	Resultados finales	14
5	CONCLUSIONES	14



1 INTRODUCCIÓN

1.1 Objetivo del documento

El presente documento tiene como objetivo describir el Plan de Pruebas, los Casos de Prueba y los Resultados de ejecución correspondientes a la Iteración 1 del proyecto *Traductor Braille*. Su propósito es asegurar que las funcionalidades implementadas cumplen con los requisitos definidos en el *Documento de Requisitos SRS*, especialmente los requisitos funcionales RF01 (transcribir texto en español a Braille) y RF02 (generar señalética Braille).

1.2 Alcance

Este documento abarca únicamente las pruebas asociadas a la **primera iteración del sistema**. Dentro de este alcance se evalúa el comportamiento del traductor en las siguientes funciones:

- Ingreso y captura de texto en español.
- Traducción de caracteres alfabéticos, vocales acentuadas, letras especiales (ñ, ü), signos básicos y números al sistema Braille.
- Visualización de la salida Braille en la interfaz.
- Generación de señalética Braille básica a partir de un texto ingresado.

Las pruebas no incluyen funcionalidades futuras como conversión Braille → texto, exportación avanzada o módulos adicionales.

1.3 Relación con documentos del proyecto

Este documento se encuentra directamente relacionado con los siguientes entregables del proyecto:

- **Documento SRS de Requisitos**, del cual provienen los requisitos funcionales RF01 y RF02 que guían todo el proceso de pruebas.
- **Casos de uso CU01 y CU02**, que describen el flujo funcional del usuario al ingresar texto y solicitar la traducción a Braille, sirviendo de base para diseñar los escenarios de prueba.
- **Documento de Arquitectura**, que define la estructura del sistema mediante el patrón Modelo–Vista–Controlador (MVC) y presenta los diagramas de componentes, clases y secuencia utilizados para comprender el comportamiento interno durante las pruebas.
- **Ambiente de Desarrollo**, que establece las herramientas utilizadas (Java, JavaFX, Maven y GitHub Flow) y garantiza que el entorno de ejecución de pruebas sea consistente y reproducible.

1.4 Objetivo de las pruebas



ICCD533 CONSTRUCCIÓN Y EVOLUCIÓN DE SOFTWARE

El objetivo de las pruebas de la Iteración 1 es verificar que el sistema cumple correctamente con los requisitos funcionales establecidos, asegurando que:

- El usuario pueda ingresar texto en español y solicitar su conversión (CU01, CU02).
- El sistema traduzca de manera correcta el texto a Braille según la normativa oficial.
- Los símbolos Braille generados se representen adecuadamente en la vista.
- El sistema pueda generar señalética Braille simple a partir del texto ingresado.
- La aplicación funcione de forma estable bajo el entorno definido, manejando adecuadamente casos válidos y entradas erróneas.

Estas pruebas garantizan la calidad, fiabilidad y funcionalidad del sistema antes de avanzar a la siguiente iteración del proyecto.

2 PLAN DE PRUEBAS

2.1 Objetivos de pruebas

El objetivo de las pruebas es verificar que las funcionalidades implementadas en la Iteración 1 cumplen correctamente con los requisitos funcionales **RF01** y **RF02** definidos en el documento SRS, asegurando que:

- El sistema transcriba texto en español a Braille de manera correcta, incluyendo abecedario, vocales acentuadas, letras especiales, signos básicos y números.
- Se genere señalética Braille simple a partir del texto ingresado por el usuario.
- El sistema se comporte de manera estable, manejando entradas válidas y entradas con errores sin fallos en la ejecución.
- Los flujos definidos en los casos de uso **CU01** e **CU02** se cumplan en su totalidad.

Estas pruebas permiten validar la calidad del software, identificar fallos tempranos y garantizar que el comportamiento observado coincide con la funcionalidad esperada para esta primera iteración.

2.2 Alcance del plan

El presente plan de pruebas cubre únicamente la funcionalidad desarrollada para la Iteración 1, la cual comprende:

- Traducción de texto en español a Braille (RF01).
- Manejo de caracteres del alfabeto español.
- Vocales acentuadas: á, é, í, ó, ú.
- Letras especiales: ñ, ü.
- Signos básicos: coma, punto, signos de interrogación y exclamación, entre otros.
- Traducción de números de uno o varios dígitos (con el signo de número).
- Generación de señalética Braille básica (RF02).



ICCD533 CONSTRUCCIÓN Y EVOLUCIÓN DE SOFTWARE

- Validación de entradas vacías o caracteres no soportados

2.3 Estrategia de pruebas

Para asegurar una validación completa, se aplicará la siguiente estrategia:

Tipos de pruebas

- **Pruebas funcionales de caja negra:** validación de entradas y salidas sin considerar la implementación interna.
- **Pruebas unitarias (opcional):** verificación de métodos clave del modelo, como el diccionario Braille y la lógica de traducción.

Niveles de prueba

- **Pruebas de unidad:** en clases del Modelo definidas en la arquitectura MVC (TraductorBraille, DiccionarioBraille).
- **Pruebas de sistema:** validación del flujo completo descrito en CU01 y CU02, desde la entrada del usuario hasta la salida en pantalla.

Técnicas empleadas

- **Comparación con la tabla Braille oficial,** según el SRS (series, numeración y signos).
- **Pruebas directas sobre la interfaz gráfica** (JavaFX) para validar la interacción del usuario.
- **Pruebas de valores límite,** como cadenas vacías, caracteres no soportados y textos mixtos.

2.4 Entorno de pruebas

Las pruebas se ejecutarán bajo el entorno definido en el documento *03_Ambiente de Desarrollo*:

Software

- **Lenguaje:** Java (OpenJDK 25.0.1)
- **Interfaz gráfica:** JavaFX
- **Gestor de dependencias:** Maven
- **IDE:** Visual Studio Code
- **Control de versiones:** GitHub – flujo GitHub Flow

Hardware

- Sistema operativo: Windows
- Procesador y memoria: especificaciones estándar para aplicaciones JavaFX

Este entorno garantiza reproducibilidad y consistencia en la ejecución de los casos de prueba.

2.5 Criterios

Criterios de inicio



ICCD533 CONSTRUCCIÓN Y EVOLUCIÓN DE SOFTWARE

- El módulo de traducción (Modelo) está implementado y compilado sin errores.
- La interfaz gráfica permite ingresar texto y mostrar la salida Braille.
- Las ramas *feature/traductor* y *feature/interfaz* han sido integradas en *develop* siguiendo GitHub Flow.
- El entorno JavaFX y Maven está configurado correctamente.

Criterios de finalización

- Todos los casos de prueba definidos para la Iteración 1 han sido ejecutados.
- En caso de presentarse fallos críticos, estos deben ser identificados, documentados y corregidos antes de finalizar la iteración.
- Si algún caso de prueba falla, debe realizarse su re-ejecución hasta obtener un resultado satisfactorio.
- Se debe disponer de evidencia (capturas y resultados) de la ejecución de cada caso de prueba, independientemente de si la prueba fue exitosa desde el inicio o requirió ajustes.

2.6 Matriz de trazabilidad

Requisito (SRS)	Descripción	Caso de uso	Casos de prueba asociados
RF01	Transcribir texto en español a Braille	CU01, CU02	CP01, CP02, CP03, CP04, CP09
RF02	Generar señalética Braille	CU01	CP05, CP06
RNF04	Fiabilidad (manejo de errores)		CP07, CP08, CP09

3 CASOS DE PRUEBA

A continuación, se describen los casos de prueba definidos para la Iteración 1 del proyecto *Traductor Braille*. Cada caso de prueba se ha diseñado con base en los requisitos funcionales RF01, RF02 y el requisito no funcional RNF04 – Fiabilidad, definidos en el documento de requisitos.

3.1 Casos de prueba funcionales (RF01)

3.1.1 CP01 – Conversión de letras y palabras simples

- **Requisito asociado:** RF01
- **Casos de uso asociados:** CU01, CU02
- **Precondiciones:**
 - La aplicación está en ejecución.
 - La interfaz muestra el campo de entrada de texto y el área de salida en Braille.
- **Datos de entrada:**
 - a, b, c
 - casa, perro, braille



ICCD533 CONSTRUCCIÓN Y EVOLUCIÓN DE SOFTWARE

- **Pasos:**

1. Abrir la aplicación Traductor Braille.
2. Ingresar una letra o palabra simple en el campo de texto.
3. Presionar el botón de “Traducir” o la acción equivalente.
4. Observar el resultado en la sección de salida Braille.

- **Resultado esperado:**

- Cada letra y palabra se traduce correctamente a su representación Braille, respetando el mapeo estándar del alfabeto español.

3.1.2 CP02 – Vocales acentuadas

- **Requisito asociado:** RF01

- **Casos de uso asociados:** CU01, CU02

- **Precondiciones:**

- Aplicación abierta y operativa.

- **Datos de entrada:**

- á, é, í, ó, ú
 - Palabras: camión, rápido, corazón

- **Pasos:**

1. Ingresar cada vocal acentuada y posteriormente las palabras indicadas.
2. Solicitar la traducción.
3. Verificar el resultado mostrado en Braille.

- **Resultado esperado:**

- Cada vocal acentuada y las palabras que las contienen se traducen correctamente según la normativa Braille para el español.

3.1.3 CP03 – Letras especiales (ñ, ü)

- **Requisito asociado:** RF01

- **Casos de uso asociados:** CU01, CU02

- **Precondiciones:**

- Aplicación en ejecución.

- **Datos de entrada:**

- niño, año, pingüino, vergüenza

- **Pasos:**

1. Ingresar las palabras de prueba en el campo de texto.
2. Ejecutar la acción de traducción.
3. Observar la salida en Braille.

- **Resultado esperado:**

- Los caracteres ñ y ü se representan con el patrón Braille correcto en todas las palabras de prueba.

3.1.4 CP04 – Números y texto mixto

- **Requisito asociado:** RF01

- **Casos de uso asociados:** CU01, CU02



ICCD533 CONSTRUCCIÓN Y EVOLUCIÓN DE SOFTWARE

- **Precondiciones:**
 - Aplicación iniciada.
- **Datos de entrada:**
 - Números: 1, 9, 12, 345
 - Texto mixto: Piso 3, Sala 101
- **Pasos:**
 1. Ingresar cada número y texto mixto en el campo de entrada.
 2. Solicitar la traducción a Braille.
 3. Revisar la salida generada.
- **Resultado esperado:**
 - Para números, el sistema antepone el **signo de número** y luego los patrones Braille de cada dígito.
 - En textos mixtos, primero se traduce el texto y luego, tras un espacio, se muestra el signo de número seguido de los dígitos correspondientes.

3.2 Casos de prueba funcionales (RF02)

3.2.1 CP05 – Generación de señalética simple

- **Requisito asociado:** RF02
- **Casos de uso asociados:** CU01 (ingresar texto), CU02 (traducir)
- **Precondiciones:**
 - La aplicación está en ejecución.
 - La funcionalidad de señalética está disponible en la interfaz.
- **Datos de entrada:**
 - Baño mujeres
- **Pasos:**
 1. Ingresar el texto Baño mujeres en el campo de texto.
 2. Seleccionar la opción para generar señalética Braille (o el modo correspondiente).
 3. Solicitar la traducción.
 4. Observar la representación de la señalética en la salida.
- **Resultado esperado:**
 - Se genera la salida Braille correcta para el texto ingresado, en un formato adecuado para ser usado como señalética (disposición clara y completa del texto en Braille).

3.2.2 CP06 – Generación de señalética con números

- **Requisito asociado:** RF02
- **Casos de uso asociados:** CU01, CU02
- **Precondiciones:**
 - Aplicación operativa.
- **Datos de entrada:**
 - Piso 1
 - Consultorio 3



ICCD533 CONSTRUCCIÓN Y EVOLUCIÓN DE SOFTWARE

- **Pasos:**

1. Ingresar cada uno de los textos de prueba en el campo de entrada.
2. Activar la opción de generación de señalética.
3. Solicitar la traducción.
4. Revisar el resultado en la sección de salida Braille.

- **Resultado esperado:**

- El texto se traduce correctamente y los números se representan con el signo de número y sus dígitos correspondientes.
- El resultado mantiene un formato coherente para ser utilizado en placas o rótulos Braille.

3.3 Casos de error / validaciones (RNF04 – Fiabilidad)

3.3.1 CP07 – Entrada vacía

- **Requisito asociado:** RNF04 – Fiabilidad

- **Casos de uso asociados:** CU01

- **Precondiciones:**

- Aplicación en ejecución.

- **Datos de entrada:**

- Cadena vacía:

- **Pasos:**

1. Dejar el campo de texto vacío.
2. Presionar el botón de “Traducir” o la opción correspondiente.
3. Observar el comportamiento de la aplicación.

- **Resultado esperado:**

- El sistema no se bloquea ni se cierra inesperadamente.
- Se muestra un mensaje de advertencia o información indicando que no hay texto para traducir.

3.3.2 CP08 – Caracteres no soportados

- **Requisito asociado:** RNF04 – Fiabilidad

- **Casos de uso asociados:** CU01, CU02

- **Precondiciones:**

- Aplicación iniciada y estable.

- **Datos de entrada:**

- Hola 😊

- Texto @#% con símbolos

- **Pasos:**

1. Ingresar los textos de prueba que contienen caracteres no soportados.
2. Solicitar la traducción.
3. Observar el resultado en la salida y el comportamiento del sistema.

- **Resultado esperado:**

- El sistema no se bloquea.



ICCD533 CONSTRUCCIÓN Y EVOLUCIÓN DE SOFTWARE

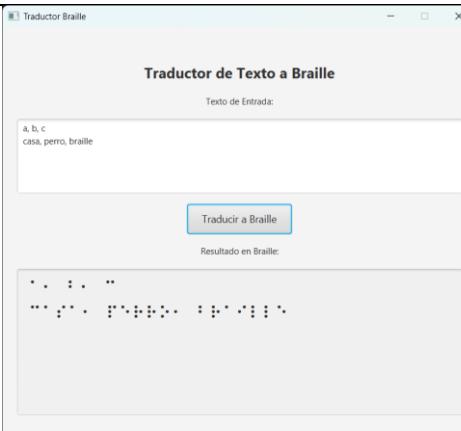
- Los caracteres no soportados se ignoran, se reemplazan o se notifica al usuario de manera clara, según lo defina el diseño, sin interrumpir la ejecución.

3.3.3 CP09 – Texto extenso

- **Requisito asociado:** RNF04 – Fiabilidad
- **Casos de uso asociados:** CU01, CU02
- **Precondiciones:**
 - Aplicación funcionando correctamente.
- **Datos de entrada:**
 - Un párrafo largo de varias líneas en español (texto continuo).
- **Pasos:**
 1. Ingresar el párrafo de texto extenso en el campo de entrada.
 2. Solicitar la traducción a Braille.
 3. Revisar el tiempo de respuesta y el resultado presentado.
- **Resultado esperado:**
 - La aplicación procesa el texto extenso sin congelarse ni cerrarse.
 - El resultado Braille se muestra completo o dividido de forma coherente, según el diseño de la interfaz.

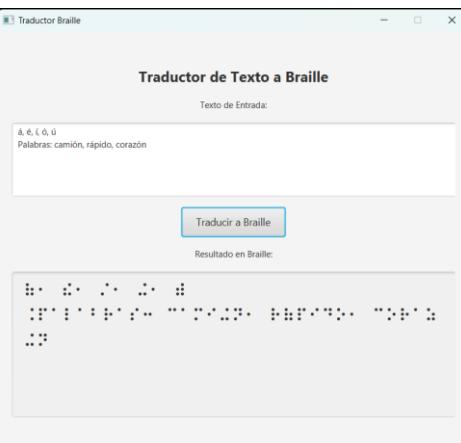
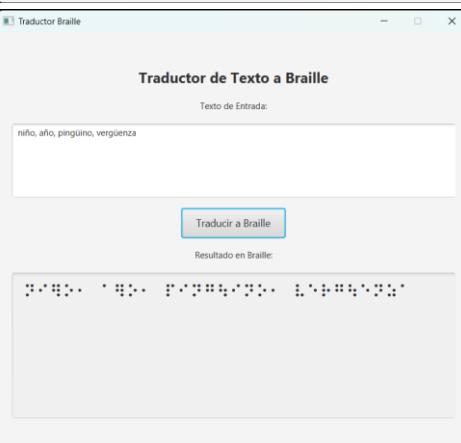
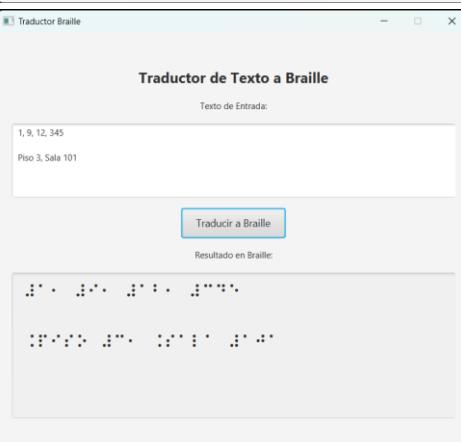
4 RESULTADOS DE EJECUCIÓN

4.1 Resultados iniciales

ID, CP	Resultado inicial (OK/FALLÓ)	Evidencia (captura)	Observación
CP01	OK		La traducción de letras y palabras simples coincide con el Braille oficial.

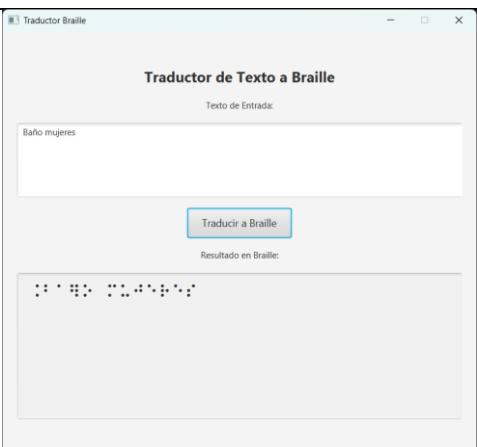
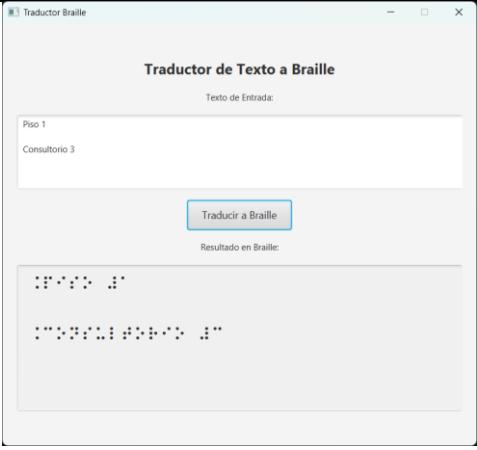


ICCD533 CONSTRUCCIÓN Y EVOLUCIÓN DE SOFTWARE

CP02	OK		Vocales acentuadas y palabras con acento son traducidas correctamente a Braille.
CP03	OK		ñ y ü se traducen correctamente en todas las palabras
CP04	OK		Los números y textos mixtos se traducen correctamente, incluyendo el signo de número.



ICCD533 CONSTRUCCIÓN Y EVOLUCIÓN DE SOFTWARE

CP05	OK		La señalética para “Baño mujeres” se genera correctamente en Braille.
CP06	OK		La señalética con texto + número se genera correctamente y usa bien el signo de número.
CP07	OK		El sistema detecta correctamente la entrada vacía y muestra un mensaje informativo sin fallar.



ICCD533 CONSTRUCCIÓN Y EVOLUCIÓN DE SOFTWARE

CP08	OK		El sistema no soporta emoji ni símbolos especiales, pero los maneja sin fallar, cumpliendo RNF04.
CS09	OK		El texto extenso es traducido completamente sin errores y sin afectar la estabilidad del sistema.

4.2 Análisis de fallos

Durante la ejecución de los casos de prueba CP01–CP09 no se identificaron fallos críticos ni comportamientos inesperados. El sistema respondió adecuadamente ante entradas válidas y no válidas, cumpliendo con los requisitos funcionales **RF01** y **RF02**, así como con el requisito no funcional **RNF04 – Fiabilidad**.

Las funcionalidades principales (traducción de texto a Braille, manejo de acentos, caracteres especiales, números y señalética) operaron sin generar errores de traducción, interrupciones en la ejecución o fallos en la interfaz.

En el caso de caracteres no soportados (CP08), el sistema permitió visualizar dichos caracteres sin afectar el procesamiento, demostrando robustez en el manejo de excepciones.

4.3 Correcciones aplicadas

No fue necesario aplicar correcciones durante esta iteración, debido a que todos los casos de prueba fueron exitosos en la primera ejecución.

Las funcionalidades implementadas se comportaron conforme al diseño descrito en la arquitectura MVC del proyecto y al mapeo Braille definido en el documento SRS. El sistema mostró estabilidad, completitud en la traducción y manejo adecuado de situaciones excepcionales, por lo que no se requirieron ajustes adicionales.



4.4 Re-ejecución

Dado que todos los casos de prueba (CP01–CP09) tuvieron un resultado **OK** en la ejecución inicial, no fue necesaria la re-ejecución de ningún caso.

Para efectos de validación, se verificó el comportamiento del sistema con entradas repetidas y variadas, confirmando que los resultados eran consistentes y reproducibles, lo que refuerza la estabilidad del sistema para la Iteración 1.

4.5 Resultados finales

Los resultados finales demuestran que **todos los casos de prueba fueron exitosos**, validando completamente los requisitos establecidos para la Iteración 1:

- **RF01:** Transcribir texto en español a Braille
- **RF02:** Generar señalética Braille
- **RNF04:** Manejar entradas no válidas o caracteres no soportados sin fallos

El sistema mostró un comportamiento estable, preciso y coherente con lo esperado, incluyendo:

- Traducción correcta del abecedario, acentos, caracteres especiales y números.
- Manejo adecuado de señalética con y sin números.
- Estabilidad ante textos extensos.
- Robustez al procesar caracteres no soportados.

Por lo tanto, se concluye que la Iteración 1 del Traductor Braille está **completamente validada**, cumpliendo con los objetivos planteados para esta fase del proyecto.

5 CONCLUSIONES

Las pruebas realizadas en la Iteración 1 del proyecto *Traductor Braille* permitieron comprobar que las funcionalidades implementadas cumplen correctamente con los requisitos establecidos en el SRS. Los casos de prueba funcionales demostraron que el sistema traduce de forma adecuada letras, vocales acentuadas, caracteres especiales, números, texto mixto y señalética Braille, validando completamente los requisitos **RF01** y **RF02**.

Asimismo, las pruebas de fiabilidad confirmaron que la aplicación responde de manera estable ante entradas vacías, caracteres no soportados y textos extensos, cumpliendo el requisito **RNF04**. Durante la ejecución no se identificaron fallos ni se requirieron correcciones, evidenciando un comportamiento consistente y una correcta implementación bajo la arquitectura MVC.

En conjunto, los resultados muestran que el sistema es funcional, estable y apto para avanzar hacia las siguientes iteraciones del proyecto.