



## ICCD633 CONSTRUCCIÓN Y EVOLUCIÓN DE SOFTWARE

*Versión 1.1.0*

### 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 .....	6
2.6 Matriz de trazabilidad .....	6
3 CASOS DE PRUEBA .....	6
3.1 Casos de prueba funcionales (RF01) .....	7
3.1.1 CP01 – Conversión de letras y palabras simples.....	7
3.1.2 CP02 – Vocales acentuadas.....	7
3.1.3 CP03 – Letras especiales (ñ, ü) .....	8
3.1.4 CP04 – Números y texto mixto .....	8
3.1.5 CP11 – Conversión de texto con mayúsculas.....	9
3.2 Casos de prueba funcionales (RF02) .....	9
3.2.1 CP05 – Generación de señalética simple.....	9
3.2.2 CP06 – Generación de señalética con números.....	10
3.3 Casos de error / validaciones (RNF04 – Fiabilidad).....	10
3.3.1 CP07 – Entrada vacía .....	11
3.3.2 CP08 – Caracteres no soportados .....	11



## **ICCD633 CONSTRUCCIÓN Y EVOLUCIÓN DE SOFTWARE**

*Versión 1.1.0*

3.3.3	CP09 – Texto extenso.....	11
<b>5</b>	<b>RESULTADOS DE EJECUCIÓN.....</b>	<b>13</b>
5.1	Resultados iniciales.....	13
5.2	Análisis de fallos.....	17
5.3	Correcciones que deberían aplicarse.....	17
5.4	Re-ejecución .....	17
5.5	Resultados finales .....	17
<b>6</b>	<b>CONCLUSIONES .....</b>	<b>17</b>



## ICCD633 CONSTRUCCIÓN Y EVOLUCIÓN DE SOFTWARE

*Versión 1.1.0*

### 1 INTRODUCCIÓN

#### 1.1 Objetivo del documento

Este documento tiene como propósito detallar el Plan de Pruebas, los Casos de Prueba y los Resultados obtenidos durante la ejecución de la Iteración 1 del proyecto Traductor Braille. Su objetivo principal es garantizar que las funcionalidades desarrolladas cumplen con los requisitos establecidos en el Documento de Requisitos SRS, específicamente los requisitos funcionales RF01 (convertir texto en español a Braille) y RF02 (generar señalética en Braille), los cuales contemplan la creación de un archivo PDF con la señalética generada.

#### 1.2 Alcance

Este documento se limita a las pruebas correspondientes a la primera iteración del sistema. Dentro de este alcance se evalúa el funcionamiento del traductor en las siguientes áreas:

- Ingreso y captura de texto en español.
- Conversión de caracteres alfabéticos, vocales con tilde, letras especiales (ñ, ü), signos básicos y números al sistema Braille.
- Visualización del resultado en Braille dentro de la interfaz.
- Generación de señalética Braille en pantalla.
- Creación de un archivo PDF que contiene la señalética generada (CU3).

Quedan fuera de estas pruebas las funcionalidades futuras, como la conversión de Braille a texto, módulos avanzados de edición o nuevas versiones del generador de documentos.

#### 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.
- **Casos de uso CU01, CU02 y CU3**, que describen el flujo funcional del ingreso de texto, traducción a Braille y generación de señalética en documento PDF.
- **Documento de Arquitectura**, que define la estructura del sistema mediante el patrón Modelo–Vista–Controlador (MVC) y describe el funcionamiento del servicio encargado de generar el PDF.
- **Documento de Ambiente de Desarrollo**, que especifica las herramientas utilizadas (Java, JavaFX, Maven, librerías iTextPDF y GitHub Flow), asegurando que las pruebas sean reproducibles en un entorno controlado.

#### 1.4 Objetivo de las pruebas



## ICCD633 CONSTRUCCIÓN Y EVOLUCIÓN DE SOFTWARE

Versión 1.1.0

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 interfaz.
- El sistema genere señalética Braille y permita descargar un documento PDF con dicha señalética (CU3).
- La aplicación funcione de forma estable bajo el entorno definido, manejando adecuadamente entradas válidas, casos erróneos y funciones adicionales como el botón “Limpiar”.

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

## 2 PLAN DE PRUEBAS

### 2.1 Objetivos de pruebas

El propósito de estas pruebas es confirmar que las funcionalidades desarrolladas en la Iteración 1 cumplen con los requisitos funcionales RF01 y RF02 establecidos en el SRS, asegurando lo siguiente:

- El sistema transcribe correctamente texto en español a Braille, incluyendo abecedario, vocales acentuadas, caracteres especiales (ñ, ü), signos básicos, números y texto en mayúsculas.
- Se genera señalética Braille a partir del texto ingresado y se permite descargar un archivo PDF con dicha señalética, conforme al caso de uso CU3.
- El sistema mantiene estabilidad, gestionando adecuadamente entradas válidas, no soportadas y cadenas vacías sin errores en la ejecución.
- La aplicación responde adecuadamente a funciones de interfaz como el botón “Limpiar”.
- Los flujos definidos en los casos de uso CU01, CU02 y CU3 se cumplen en su totalidad.

Estas pruebas contribuyen a validar la calidad del software, detectar errores tempranos y garantizar que el comportamiento observado coincide con lo esperado 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.



## ICCD633 CONSTRUCCIÓN Y EVOLUCIÓN DE SOFTWARE

Versión 1.1.0

- Traducción de números de uno o varios dígitos con el signo de número.
- Traducción de texto ingresado completamente en mayúsculas.
- Generación de señalética Braille básica (RF02).
- Generación y descarga de un documento PDF con la señalética Braille (CU3).
- Validación de entradas vacías, caracteres no soportados y funcionamiento del botón “Limpiar”.

### 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 examinar la implementación interna.

#### Niveles de prueba

- **Pruebas de unidad:** En clases del Modelo (TraductorBraille, DiccionarioBraille).
- **Pruebas de sistema:** Validación del flujo completo descrito en CU01, CU02 y CU3, desde la entrada del usuario hasta la visualización y generación del PDF.

#### Técnicas empleadas

- Comparación con la tabla Braille oficial indicada en el SRS.
- Pruebas directas sobre la interfaz JavaFX para validar la interacción del usuario.
- Pruebas de valores límite como:
  - cadenas vacías
  - caracteres no soportados
  - texto extenso
  - **texto completamente en mayúsculas**
- **Verificación del archivo PDF generado**, comprobando su existencia, formato y contenido correcto.

#### Referencia de resultados esperados

Los resultados esperados utilizados en los casos de prueba se basan en la *Signografía Básica del Braille Español* establecida por la Comisión Braille Española (ONCE), que define oficialmente la representación de letras, signos, números, vocales acentuadas y prefijos de mayúsculas en el sistema Braille.

### 2.4 Entorno de pruebas

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

#### Software

- Lenguaje: Java (OpenJDK 25.0.1)
- Interfaz gráfica: JavaFX



## ICCD633 CONSTRUCCIÓN Y EVOLUCIÓN DE SOFTWARE

*Versión 1.1.0*

- Gestor de dependencias: Maven
- IDE: Visual Studio Code
- Librerías: iTextPDF (kernel, layout, io)
- Control de versiones: GitHub – 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

- El módulo de traducción está implementado y compilado sin errores.
- La interfaz gráfica permite ingresar texto, traducirlo, limpiar campos y descargar el PDF.
- Las ramas feature relacionadas con la traducción, interfaz y generación de PDF han sido integradas en develop según GitHub Flow.
- JavaFX, Maven y sus dependencias (incluyendo iTextPDF) están correctamente configurados.

#### 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.
- Si algún caso de prueba falla, debe realizarse su re-ejecución hasta obtener un resultado satisfactorio.
- Se debe disponer de evidencia (capturas, registros y/o archivos PDF generados) de cada caso de prueba ejecutado.

### 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, CP11
RF02	Generar señalética Braille	CU01, CU02, CU03	CP05, CP06
RNF04	Fiabilidad (manejo de errores)		CP07, CP08, CP09, CP10

### 3 CASOS DE PRUEBA



## ICCD633 CONSTRUCCIÓN Y EVOLUCIÓN DE SOFTWARE

Versión 1.1.0

Se han definido pruebas basadas en los requisitos funcionales **RF01** (transcripción de texto a Braille) y **RF02** (generación de señalética Braille en PDF), además del requisito no funcional **RNF04** (fiabilidad y manejo de errores). Los resultados esperados siguen la **Signografía Básica del Sistema Braille Español (ONCE)**, asegurando la correcta representación de mayúsculas, números, acentos, caracteres especiales y convenciones oficiales.

### 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
- **Pasos:**
  - a. Abrir la aplicación Traductor Braille.
  - b. Ingresar una letra o palabra simple en el campo de texto.
  - c. Presionar el botón “Traducir a Braille”.
  - d. Observar el resultado en la sección de resultado en Braille.
- **Resultado esperado:**
  - a → ·
  - b → :
  - c → ..
  - casa → .. · ; ..
  - perro → .. · ; ..
  - braille → .. · ; ..

#### 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
  - camión, rápido, corazón
- **Pasos:**
  - a. Ingresar cada vocal acentuada y posteriormente las palabras indicadas.
  - b. Presionar “Traducir a Braille”.



**ICCPD633 CONSTRUCCIÓN Y EVOLUCIÓN DE SOFTWARE**

Versión 1.1.0

- c. 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.
    - á → ☿
    - é → ☻
    - í → .̄
    - ó → ..̄
    - ú → ☽
    - camión → ☺ ☺ ☺ ☺ ☺ ☺
    - rápido → ☺ ☺ ☺ ☺ ☺
    - corazón → ☺ ☺ ☺ ☺ ☺

### 3.1.3 CP03 – Letras especiales (ñ, ü)



### 3.1.4 CP04 – Números y texto mixto

- **Requisito asociado:** RF01
  - **Casos de uso asociados:** CU01, CU02
  - **Precondiciones:**
    - Aplicación iniciada.
  - **Datos de entrada:**
    - Números: 1, 9, 12, 345
    - Texto mixto: Piso 3, Sala 101
  - **Pasos:**
    - a. Ingresar cada número y texto mixto en el campo de entrada.
    - b. Presionar “Traducir a Braille”.



# **ICCD633 CONSTRUCCIÓN Y EVOLUCIÓN DE SOFTWARE**

Versión 1.1.0

- c. Revisar la salida generada.

- **Resultado esperado:**

- $1 \rightarrow \begin{array}{c} \bullet \\ \bullet \end{array}$
  - $9 \rightarrow \begin{array}{c} \bullet \\ \bullet \\ \bullet \end{array}$
  - $12 \rightarrow \begin{array}{c} \bullet \\ \bullet \\ \bullet \end{array} :$
  - $345 \rightarrow \begin{array}{c} \bullet \\ \bullet \\ \bullet \end{array} :\bullet :\bullet$
  - Piso 3  $\rightarrow \begin{array}{c} \bullet \\ \bullet \\ \bullet \end{array} :\bullet :\bullet :\bullet$
  - Sala 101  $\rightarrow \begin{array}{c} \bullet \\ \bullet \\ \bullet \end{array} :\bullet :\bullet :\bullet :\bullet :\bullet :\bullet$

### 3.1.5 CP11 – Conversión de texto con mayúsculas

- **Requisito asociado:** RF01
  - **Casos de uso asociados:** CU01, CU02
  - **Precondiciones:**
    - Aplicación en ejecución.
  - **Datos de entrada:**
    - HOLA
    - Hola
    - hola
  - **Pasos:**
    - a. Ingresar el texto de prueba con distintas combinaciones de mayúsculas.
    - b. Presionar “Traducir a Braille”.
    - c. Observar la salida.
  - **Resultado esperado (ONCE):**
    - HOLA → : ; : ; : ; :
    - Hola → ; ; : ; : ; :
    - hola → ; ; ; ; : ; :

### 3.2 Casos de prueba funcionales (RF02)

### 3.2.1 CP05 – Generación de señalética simple

- **Requisito asociado:** RF02
  - **Casos de uso asociados:** CU01, CU02, CU03
  - **Precondiciones:**
    - La aplicación está en ejecución.
    - La funcionalidad de generación de PDF está disponible (botón “**Descargar PDF**” visible).
  - **Datos de entrada:**
    - Baño mujeres
  - **Pasos:**
    - a. Ingresar el texto “Baño mujeres”, (B mayúscula, resto minúsculas).
    - b. Presionar “Traducir a Braille”.
    - c. Verificar la salida Braille.
    - d. Presionar “Descargar PDF”.



## ICCD633 CONSTRUCCIÓN Y EVOLUCIÓN DE SOFTWARE

Versión 1.1.0

- e. Abrir el PDF generado y revisar el contenido.

- **Resultado esperado:**

- Salida Braille:

**Baño mujeres** → :

El PDF descargado debe contener la misma secuencia **texto legible arriba y Braille debajo**, sin adornos, para que sea claro y accesible.

Cada señal en su propio bloque (idealmente una por página si es señalética grande).

### 3.2.2 CP06 – Generación de señalética con números

- **Requisito asociado:** RF02
- **Casos de uso asociados:** CU01, CU02
- **Precondiciones:**
  - La aplicación está en ejecución.
  - La funcionalidad de generación de PDF está disponible (botón “**Descargar PDF**” visible).
- **Datos de entrada:**
  - Piso 1
  - Consultorio 3
- **Pasos:**
  - a. Ingresar Piso 1 en el campo de texto.
  - b. Presionar “**Traducir a Braille**” y observar la salida.
  - c. Repetir con Consultorio 3.
  - d. Presionar “**Descargar PDF**”.
  - e. Revisar en el PDF que el contenido Braille coincida.
- **Resultado esperado:**
  - **Piso 1** → :P:·:·:·:·:  
(Indicador de mayúscula : antes de “P”, y .: para números seguido de · para el dígito 1).
  - **Consultorio 3** → :C:·:·:·:·:  
(Indicador de mayúscula : antes de “C”, y .: seguido de “” para el dígito 3).
  - Texto en español en la parte superior (centrado o alineado).
  - Braille justo debajo en la misma línea, sin adornos.
  - Cada señal en su propio bloque (idealmente una por página si es señalética grande).

### 3.3 Casos de error / validaciones (RNF04 – Fiabilidad)



## ICCD633 CONSTRUCCIÓN Y EVOLUCIÓN DE SOFTWARE

Versión 1.1.0

### 3.3.1 CP07 – Entrada vacía

- **Requisito asociado:** RNF04
- **Casos de uso asociados:** CU01
- **Precondiciones:**
  - Aplicación en ejecución.
- **Datos de entrada:**
  - Cadena vacía:
- **Pasos:**
  - a. Dejar el campo de texto vacío.
  - b. Presionar el botón de “Traducir” o la opción correspondiente.
  - c. Observar el comportamiento de la aplicación.
- **Resultado esperado:**
  - No se genera ninguna salida Braille.
  - Se muestra un mensaje de advertencia del tipo: “Por favor ingrese texto para traducir”.
  - La aplicación no se bloquea ni se cierra inesperadamente.

### 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:**
  - a. Ingresar Hola 😊 en el campo de texto.
  - b. Presionar “Traducir a Braille”.
  - c. Repetir el procedimiento con Texto @#% con símbolos.
  - d. Observar la salida y el comportamiento del sistema.
- **Resultado esperado:**
  - Para Hola 😊 se traduce únicamente el texto soportado:  
Hola → :·:·:· ·  
(se ignora el emoji 😊).
  - Para Texto @#% con símbolos se ignoran @#% y se traduce el resto:  
Texto con símbolos →  
:·:·:·:· ·:·:·:·:· ·:·:·:·:·
  - En ningún caso la aplicación debe bloquearse o cerrarse.

### 3.3.3 CP09 – Texto extenso

- **Requisito asociado:** RNF04 – Fiabilidad
- **Casos de uso asociados:** CU01, CU02



**ICCD633 CONSTRUCCIÓN Y EVOLUCIÓN DE SOFTWARE**

Versión 1.1.0



#### 4 3.3.4 CP10 – Funcionalidad del botón “Limpiar”

- **Requisito asociado:** RNF04 – Fiabilidad
  - **Casos de uso asociados:** CU01
  - **Precondiciones:**
    - La aplicación está en ejecución.
    - Existe texto previamente ingresado y una salida Braille generada.
  - **Datos de entrada:**
    - Cualquier texto válido, por ejemplo: Hola mundo.
  - **Pasos:**
    - a. Ingresar el texto de prueba en el campo de entrada.
    - b. Presionar “Traducir a Braille” y verificar que se genere una salida.
    - c. Presionar el botón “Limpiar”.
  - **Resultado esperado:**
    - El campo de texto queda vacío.
    - El área de “Resultado en Braille” se limpia por completo.
    - No se producen errores ni cierres inesperados.

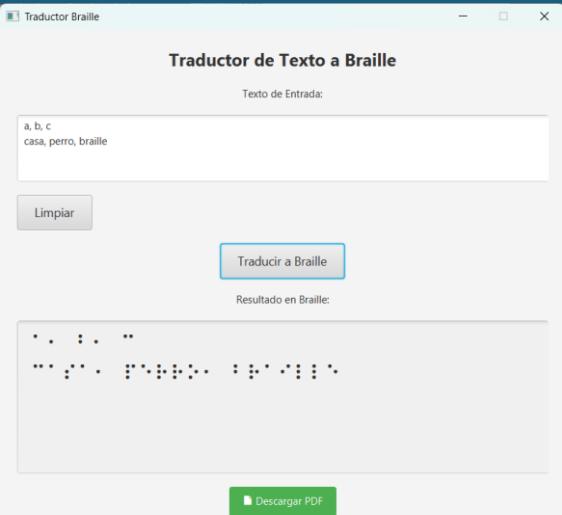
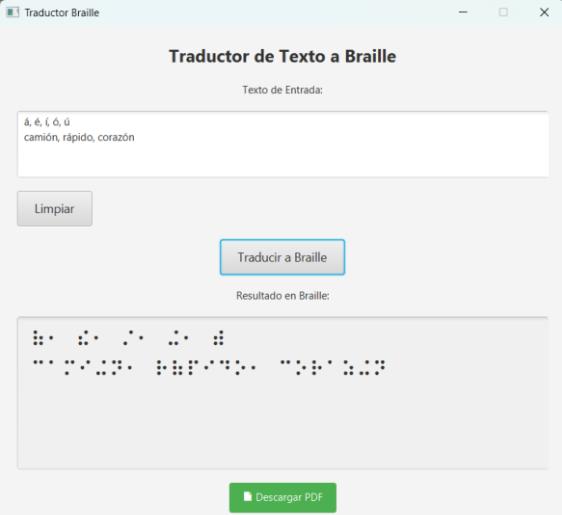


## ICCD633 CONSTRUCCIÓN Y EVOLUCIÓN DE SOFTWARE

Versión 1.1.0

### 5 RESULTADOS DE EJECUCIÓN

#### 5.1 Resultados iniciales

ID, CP	Resultado inicial (OK/FALLÓ)	Evidencia (captura)	Observación
CP01	OK		La traducción de letras y palabras simples coincide con la signografía Braille oficial (ONCE). No se presentan errores en la conversión.
CP02	OK		Las vocales acentuadas y las palabras con caracteres especiales se traducen correctamente según la normativa Braille española. No se presentan errores.



## ICCD633 CONSTRUCCIÓN Y EVOLUCIÓN DE SOFTWARE

Versión 1.1.0

CP03	OK		Los números y textos mixtos se traducen correctamente al Braille español, incluyendo el uso del indicador numérico (⠼). No se presentan errores
CP04	OK		El PDF generado contiene el texto legible arriba y la secuencia Braille correcta debajo, cumpliendo el requisito RF02. Diseño limpio y sin errores.



## ICCD633 CONSTRUCCIÓN Y EVOLUCIÓN DE SOFTWARE

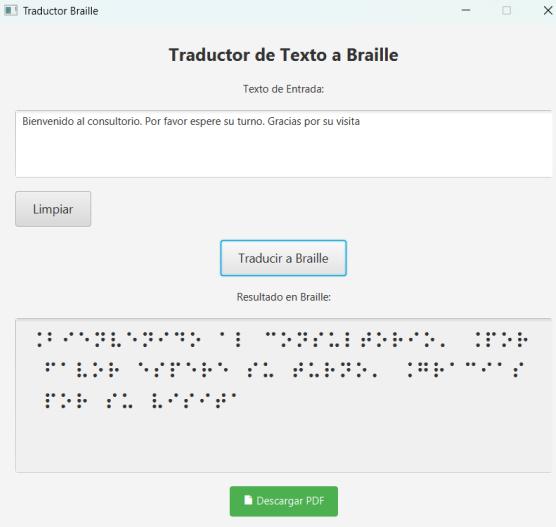
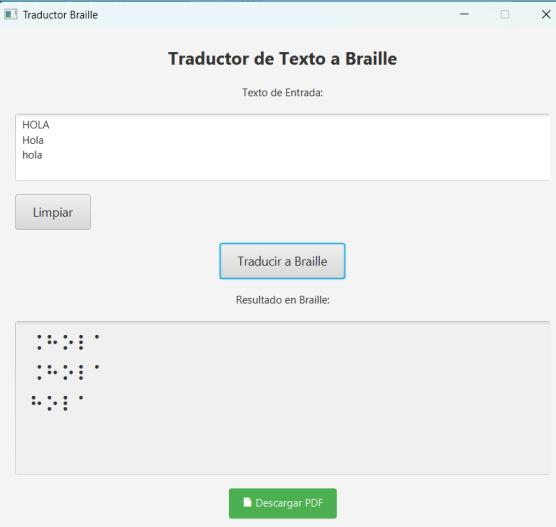
Versión 1.1.0

CP06	OK		Las señales con números se traducen correctamente, usando mayúsculas y prefijo numérico. El PDF coincide con la salida Braille y mantiene el formato limpio y accesible. Cumple RF02.
CP07	OK		Se muestra el mensaje de advertencia correcto y la aplicación permanece estable sin generar salida Braille.
CP08	OK		La aplicación procesa el texto soportado y omite correctamente los caracteres no reconocidos (emoji y símbolos @#%). Las palabras válidas se traducen sin errores a Braille, mientras que los caracteres no soportados no provocan fallos ni cierres inesperados.



## ICCD633 CONSTRUCCIÓN Y EVOLUCIÓN DE SOFTWARE

Versión 1.1.0

CS09	OK		El texto extenso se tradujo completamente a Braille, la aplicación no se bloqueó y el resultado se muestra de forma coherente en la interfaz.
CS10	OK		El botón “Limpiar” elimina correctamente el texto de entrada y la salida Braille, sin errores ni cierres inesperados.
CS11	FALLÓ		La salida Braille no refleja correctamente las reglas para mayúsculas: <ul style="list-style-type: none"><li>Para “HOLA” debería mostrar doble indicador de mayúsculas, pero solo aparece uno.</li><li>Para “Hola” y “hola” la conversión parece correcta.</li></ul>



## 5.2 Análisis de fallos

De los once casos de prueba ejecutados, **solo se identificó un fallo**: el sistema **no aplica el signo de mayúsculas en la traducción a Braille**, por lo que las entradas “HOLA”, “Hola” y “hola” generan la misma salida. Este comportamiento incumple lo establecido en el estándar ONCE y afecta directamente al requisito **RF01**.

## 5.3 Correcciones que deberían aplicarse.

La única corrección necesaria es **implementar correctamente el manejo de mayúsculas en la traducción a Braille**, conforme a la signografía oficial ONCE. En concreto, se debe:

- Aplicar el **prefijo de mayúscula (:) cuando una letra esté en mayúscula**.
- Aplicar **doble prefijo (:) (:) cuando toda la palabra esté escrita en mayúsculas**.
- Mantener la conversión normal cuando el texto esté en minúsculas.

Con esta mejora, la aplicación cumplirá completamente el requisito **RF01** y garantizará una traducción fiel a los estándares del sistema Braille.

## 5.4 Re-ejecución

A la espera de la corrección.

## 5.5 Resultados finales

De los once casos de prueba ejecutados, **diez fueron aprobados** y uno presentó fallo.

La aplicación demuestra un funcionamiento estable, fiable y consistente en la mayoría de funcionalidades evaluadas: traducción de texto estándar, acentos, caracteres especiales, números, señalética y manejo de la interfaz.

El único incumplimiento identificado corresponde al **manejo de mayúsculas**, que requiere la incorporación del prefijo Braille correspondiente para cumplir completamente con el estándar ONCE.

En general, el sistema es funcional y se considera **apto**, quedando pendiente únicamente la corrección del soporte de mayúsculas para alcanzar el 100% de conformidad con los requisitos establecidos.

## 6 CONCLUSIONES

De los once casos de prueba ejecutados, **diez fueron exitosos y uno falló**, lo que evidencia que la aplicación cumple de forma correcta y fiable con casi todos los requisitos definidos (RF01, RF02 y RNF04). Las funciones de traducción de texto, números, acentos, caracteres especiales, señalética y manejo de la interfaz (entrada vacía, caracteres no soportados, texto extenso y botón “Limpiar”) se comportaron según lo esperado y sin afectar la estabilidad del sistema.

El **único fallo identificado** corresponde al manejo de mayúsculas (CP11), donde la aplicación no aplica el prefijo Braille de mayúscula ni el doble prefijo para palabras totalmente en mayúsculas, incumpliendo parcialmente el requisito **RF01** y el estándar ONCE. Este defecto no compromete



ESCUELA  
POLÍTÉCNICA  
NACIONAL



## ICCD633 CONSTRUCCIÓN Y EVOLUCIÓN DE SOFTWARE

*Versión 1.1.0*

la estabilidad general del sistema, pero sí la fidelidad de la traducción. Una vez corregido este punto, la aplicación podrá considerarse **plenamente alineada** con los requisitos funcionales y con la normativa oficial de signografía Braille.