Manual de Pruebas — Braille Express

**Fecha**: 27/06/2025

**Autor**: Ylia Jamile Ochoa Gutierrez

**Versión**: 1.0

# 1. Introducción

Este manual describe el proceso de verificación y validación del sistema Braille Express, que incluye la aplicación móvil Android, la base de datos en Firebase Realtime Database y el módulo físico basado en Arduino Nano con servomotores. Está dirigido al equipo de control de calidad (QA), desarrolladores y personal técnico, con el fin de garantizar que la solución cumpla con los requisitos funcionales, sea estable, segura y accesible para personas con discapacidad visual.

# 2. Objetivos

* Verificar que las funcionalidades de traducción, historial, configuración y conexión física funcionen correctamente.
* Detectar errores antes de la entrega final o su uso en el entorno educativo.
* Asegurar la usabilidad y accesibilidad del sistema.
* Validar que los datos se guarden y sincronicen correctamente por usuario.

# 3. Tipos de Pruebas Aplicadas

**3.1 Pruebas Unitarias**

* Validan clases Java como MapaBrailleModel, BrailleTraductorModel y los ViewModels.
* Herramientas: JUnit, Mockito.

**3.2 Pruebas de Integración**

* Verifican la interacción entre Firebase y la app (ej. guardar historial, obtener configuración).
* Ejemplo: Traducción → Envío a Arduino → Registro en Firebase.

**3.3 Pruebas Funcionales**

* Validan funciones como: traducción de texto, selección de tema, conexión Bluetooth, y recuperación del historial.
* Herramientas: TestLab (Firebase), Espresso (Android).

**3.5 Pruebas Manuales (Caja Negra)**

* Pruebas desde el punto de vista del usuario final (estudiante o docente).
* Simulan el uso en dispositivos reales y evalúan respuesta visual y física.

# 4. Ambiente de Pruebas

|  |  |
| --- | --- |
| Componente | Versión / Condición |
| Android Studio | 7.0 o superior |
| Dispositivo | Android físico con Bluetooth |
| Firebase | Proyecto con Realtime Database |
| Arduino IDE | 1.8+ (con librería Servo + SoftwareSerial) |
| Hardware | Arduino Nano + módulo HC-06 + 6 servos |

# 5. Casos de Prueba

**5.1 Traducción de Texto a Braille**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Descripción | Entrada | Resultado Esperado | Estado |
| TC01 | Traducción simple | "hola" | Muestra braille ⠓⠕⠇⠁ en pantalla y hardware | Aceptado |
| TC02 | Traducción con signos y mayúsculas | "¡Hola!" | Muestra signos y representación en Braille | Aceptado |

**5.2 Conexión con Arduino vía Bluetooth**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Descripción | Entrada | Resultado Esperado | Estado |
| TC03 | Activar Bluetooth | Toggle activado | Conexión establecida con HC-06 | Aceptado |
| TC04 | Enviar letra traducida | Letra "a" | Servo 1 se mueve según mapa de Braille | Aceptado |

**5.3 Historial y Configuración**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Descripción | Entrada | Resultado Esperado | Estado |
| TC05 | Guardar traducción | "luz" | Se guarda con fecha y color elegido | Aceptado |
| TC06 | Cargar configuración | ID de usuario | Carga tema y color desde Firebase | Aceptado |

# 6. Resultados

Todas las pruebas funcionales fueron exitosas durante los ciclos de prueba del Sprint 1. Las funcionalidades críticas (traducción, historial, personalización y Bluetooth) se comportan conforme a lo esperado, tanto a nivel de interfaz como en hardware físico.

# 7. Criterios de Aceptación

* El sistema debe traducir texto digital a Braille correctamente (visual + físico).
* La configuración debe ser persistente por usuario.
* El historial debe almacenarse correctamente y permitir acciones como copiar y eliminar.
* La conexión Bluetooth debe ser estable y funcional.
* La app debe responder sin errores en dispositivos con Android 7 o superior.

# 8. Herramientas Utilizadas

* **Android Studio:** Emulador + pruebas unitarias/instrumentadas.
* **Firebase Test Lab:** Pruebas automáticas en distintos dispositivos.
* **JUnit / Mockito:** Validación de lógica de negocio.
* **Hardware físico:** Arduino Nano + servos + módulo Bluetooth.
* **Espresso:** Automatización de pruebas UI.

# 9. Recomendaciones Futuras

* Incluir pruebas automatizadas con Espresso para flujos completos.
* Ejecutar pruebas de rendimiento en múltiples dispositivos Android.
* Probar uso del sistema por usuarios con baja visión o sin visión para validar accesibilidad.
* Simular desconexiones de Bluetooth para validar comportamiento.
* Agregar pruebas E2E conectando app, Firebase y Arduino en tiempo real.

# 10. Contacto QA

Para informes de errores o coordinación de nuevas pruebas: [qaBrailleExpress@gmail.com](mailto:qaBrailleExpress@gmail.com)