Backlog del Producto – Braille Express

**Fecha:**01/04/2025

**Autor:** Ylia Jamile Ochoa Gutierrez

**Versión:** 1.0

# 1. Introducción

Este documento detalla el backlog del producto para **Braille Express**, una aplicación que permitirá la conversión de texto a Braille y su representación en un dispositivo físico mediante servomotores controlados por un Arduino Uno. El backlog está estructurado en **épicas**, **historias de usuario** y sus respectivos **criterios de aceptación**.

# 2. Diseño de Épicas e Historias de Usuario

Para la correcta estructuración del backlog, cada épica y su respectiva historia de usuario estarán diseñadas bajo los siguientes principios:

* **Épicas**: Representan grandes bloques funcionales del sistema, agrupando historias de usuario relacionadas.
* **Historias de Usuario**: Son descripciones cortas y centradas en el usuario sobre una funcionalidad específica del producto.
* **Criterios de Aceptación**: Reglas que deben cumplirse para que la historia de usuario sea considerada completada.
* **Estimación**: Cada historia de usuario incluye una estimación de esfuerzo en puntos de historia.
* **Definición de "Hecho"**: Se especifica cuándo una historia de usuario se considera completada con éxito.

El diseño y priorización de las épicas y las historias de usuario se revisarán periódicamente para adaptarse a nuevas necesidades del negocio y de los usuarios.

# 3. Épicas e Historias de Usuario

## Épica 1: Gestión de Usuarios y Sesiones

### Historia de Usuario 1.1: Registro de usuarios

Como usuario, quiero registrarme en la aplicación utilizando mi DNI para asegurar la autenticidad de mi cuenta.

**Criterios de Aceptación:**

* El sistema debe validar el formato del DNI como usuario.
* Se debe almacenar la información de manera segura.

**Estimación:** 5 puntos

**Definición de "Hecho":**

* El usuario puede registrarse exitosamente.
* La información se almacena de forma segura en la base de datos.

### Historia de Usuario 1.2: Inicio y cierre de sesión en la nube

Como usuario, quiero iniciar y cerrar sesión en cualquier dispositivo móvil con sincronización en la nube para acceder a mis datos de manera segura.

**Criterios de Aceptación:**

* La aplicación debe permitir autenticación desde múltiples dispositivos.
* La información debe estar sincronizada en la nube.

**Estimación:** 8 puntos

**Definición de "Hecho":**

* El usuario puede iniciar y cerrar sesión sin problemas.
* Los datos del usuario se mantienen sincronizados en la nube.

## Épica 2: Ingreso y Traducción de Texto a Braille

### Historia de Usuario 2.1: Ingreso manual de texto

Como usuario con discapacidad visual, quiero ingresar texto manualmente en la aplicación para que pueda ser traducido a Braille.

**Criterios de Aceptación:**

* La aplicación debe permitir el ingreso de caracteres alfanuméricos y signos de puntuación básicos.
* Debe validar que el texto ingresado sea compatible con la conversión a Braille.

**Estimación:** 5 puntos

**Definición de "Hecho":**

* El usuario puede ingresar texto manualmente.
* La conversión a Braille es precisa.

### Historia de Usuario 2.2: Ingreso de texto por voz

Como usuario con discapacidad visual, quiero ingresar texto mediante comandos de voz para convertirlo a Braille.

**Criterios de Aceptación:**

* La aplicación debe transcribir la voz a texto utilizando una API de reconocimiento de voz.
* El usuario podrá editar el texto antes de su conversión.

**Estimación:** 8 puntos

**Definición de "Hecho":**

* La transcripción de voz a texto es precisa.
* El usuario puede editar y convertir el texto sin problemas.

### Historia de Usuario 2.2: Traducción automáticamente el texto ingresado

Como usuario, quiero que el texto ingresado en la aplicación se traduzca automáticamente a Braille sin necesidad de acciones adicionales, para mejorar la fluidez y rapidez en la conversión.

**Criterios de Aceptación:**

* La aplicación debe detectar automáticamente cuando el usuario finaliza la escritura.
* Debe iniciar la conversión a Braille sin necesidad de presionar un botón adicional.
* El usuario debe poder visualizar la traducción en un área específica de la interfaz.
* La traducción debe ser precisa y compatible con el sistema Braille.

**Estimación:** 5 puntos

**Definición de "Hecho":**

* La traducción se activa automáticamente al finalizar la escritura.
* El texto traducido se muestra correctamente en la interfaz de usuario.
* No se requiere acción adicional por parte del usuario para iniciar la traducción.
* La conversión es precisa y se valida con pruebas funcionales.

## Épica 3: Conectividad con el Dispositivo Braille

### Historia de Usuario 3.1: Conexión Bluetooth

Como usuario, quiero que la aplicación se conecte al dispositivo Braille mediante Bluetooth para transmitir los caracteres.

**Criterios de Aceptación:**

* La aplicación debe detectar dispositivos Bluetooth disponibles.
* Debe establecer y mostrar el estado de la conexión (conectado/desconectado).

**Estimación:** 10 puntos

**Definición de "Hecho":**

* La aplicación se conecta correctamente al dispositivo Bluetooth.
* El estado de la conexión es visible y actualizable.

### Historia de Usuario 3.2: Envío de caracteres a Arduino

Como usuario, quiero que la aplicación envíe los caracteres al Arduino para que los represente en Braille.

**Criterios de Aceptación:**

* La aplicación debe transmitir los caracteres al dispositivo electrónico.
* El Arduino debe activar los servomotores correspondientes para representar cada carácter en Braille.

**Estimación:** 10 puntos

**Definición de "Hecho":**

* Los caracteres se transmiten correctamente y el Arduino los representa en Braille.

### Historia de Usuario 3.3: Reinicio manual de bluetooth

Como usuario, quiero poder reiniciar la conexión Bluetooth manualmente en caso de errores de comunicación.

**Criterios de Aceptación:**

* La aplicación debe permitir restablecer la conexión manualmente.
* Debe notificar cuando la conexión se restablezca correctamente.

**Estimación:** 5 puntos

**Definición de "Hecho":**

* La opción de reinicio manual funciona correctamente.
* La aplicación notifica al usuario el estado del reinicio.

## Épica 4: Interfaz de Usuario y Notificaciones

### Historia de Usuario 4.1: Notificaciones de acciones y errores

Como usuario, quiero recibir notificaciones cuando haya errores o confirmaciones de acciones dentro de la aplicación.

**Criterios de Aceptación:**

* Se deben mostrar mensajes emergentes para errores como "Bluetooth no disponible".
* Deben existir confirmaciones para acciones como "Texto enviado correctamente".

**Estimación:** 3 puntos

**Definición de "Hecho":**

* Los mensajes emergentes se muestran correctamente en la interfaz.
* Se pueden diferenciar entre errores y confirmaciones exitosas.

### Historia de Usuario 4.2: Notificación de actualizaciones

Como usuario, quiero recibir notificaciones cuando haya actualizaciones disponibles para la aplicación.

**Criterios de Aceptación:**

* La aplicación debe verificar actualizaciones en la nube.
* Debe notificar a los usuarios cuando haya una nueva versión disponible.

**Estimación:** 3 puntos

**Definición de "Hecho":**

* La aplicación verifica actualizaciones correctamente.
* Los usuarios reciben notificaciones de nuevas versiones.

# 4. Priorización del Backlog

|  |  |  |  |
| --- | --- | --- | --- |
| Prioridad | Historia de Usuario | Estado | Estimación (Puntos) |
| Alta | Registro seguro | Pendiente | 5 |
| Alta | Inicio y cierre de sesión en la nube | Pendiente | 8 |
| Alta | Ingreso manual de texto | Pendiente | 5 |
| Media | Ingreso de texto por voz | Pendiente | 8 |
| Alta | Traducción automática del texto ingresado | Pendiente | 5 |
| Alta | Conexión Bluetooth | Pendiente | 10 |
| Alta | Envío de caracteres a Arduino | Pendiente | 10 |
| Media | Reinicio manual de Bluetooth | Pendiente | 5 |
| Media | Notificaciones de acciones y errores | Pendiente | 3 |
| Baja | Notificaciones de actualizaciones | Pendiente | 3 |