**WP ID: WP.16 - Diseño de Software**

### **1. Descripción:**

Este documento proporciona información textual y gráfica sobre la estructura del software desarrollado para transcribir textos a Braille y viceversa, detallando tanto la arquitectura de alto nivel como el diseño detallado de los componentes.

### **2. Diseño de software arquitectónico o de alto nivel:**

#### **2.1 Descripción de la Estructura General del Software:**

El software está diseñado con una arquitectura en capas que incluye:

* **Capa de Presentación:** Interfaz de usuario implementada en HTML, CSS y JavaScript, que permite a los usuarios interactuar con las funciones de traducción. También incluye opciones adicionales como copiar texto, borrar texto, cambiar al modo claro, y realizar zoom in y zoom out.
* **Capa Lógica:** Implementación en Python de la lógica de negocio, incluyendo las funciones para validar texto, realizar traducciones entre texto y Braille, y manejar errores.
* **Capa de Datos:** Manejo de diccionarios y almacenamiento de configuraciones, como los mapeos entre caracteres de texto y Braille.

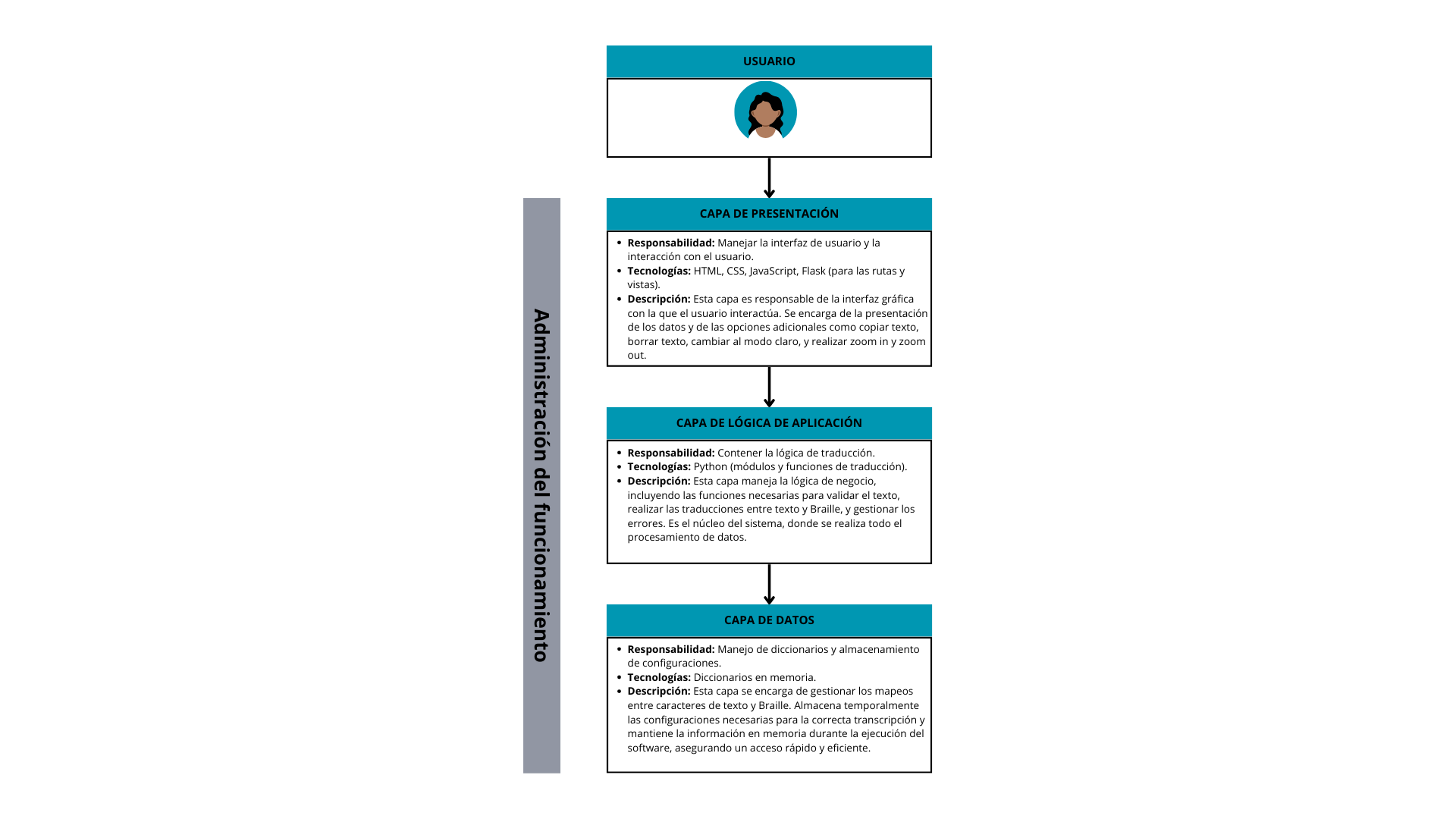


Figura 1 Arquitectura de la Aplicación

#### **2.2 Componentes de Software Requeridos:**

* **Módulo de Interfaz de Usuario:** Controla la visualización y los elementos interactivos, como botones de traducción, selección de dirección de traducción, y botones de descarga de archivos (PDF, PNG). Además, gestiona las funcionalidades adicionales de copiar texto, borrar texto, cambiar al modo claro, y zoom.
* **Módulo de Lógica de Traducción:** Contiene las funciones is\_valid\_text, text\_to\_braille, braille\_to\_text, y handle\_translation para procesar las entradas del usuario y devolver los resultados de la traducción.
* **Módulo de Exportación:** Permite la exportación de resultados en formatos PDF y PNG.
* **Módulo de Validación:** Valida las entradas del usuario y notifica sobre cualquier carácter no permitido o errores en la dirección de traducción seleccionada.

#### **2.3 Relación entre Componentes de Software:**

Los componentes se comunican a través de llamadas a funciones y el intercambio de datos. La interfaz de usuario recopila la entrada del usuario, la envía al módulo de lógica de traducción, y muestra el resultado en pantalla. El módulo de exportación se activa una vez que la traducción ha sido realizada, permitiendo al usuario descargar el resultado. Las funcionalidades adicionales como copiar texto, borrar texto, cambiar al modo claro, y zoom también están integradas en la interfaz.

#### **2.4 Consideraciones Específicas:**

* **Características de Rendimiento:** El software debe responder rápidamente a las solicitudes de traducción, generación de archivos y activación de las opciones adicionales (copiar, borrar, modo claro, zoom).
* **Interfaces de Hardware, Software y Humanas:** La interfaz está diseñada para ser compatible con diferentes navegadores, y para ser intuitiva y fácil de usar, incluyendo todas las opciones adicionales mencionadas.
* **Características de Seguridad:** Asegurar que los archivos generados no contengan errores.
* **Requisitos de Diseño de Bases de Datos:** Uso de diccionarios en memoria para el mapeo de caracteres de texto a Braille, eliminando la necesidad de una base de datos persistente.
* **Atributos de Manejo y Recuperación de Errores:** La interfaz notifica al usuario sobre cualquier entrada no válida o errores durante el proceso de traducción.

### **3. Diseño de software detallado o de bajo nivel:**

#### **3.1 Diseño Detallado:**

Cada componente del software está diseñado con cuidado para facilitar su construcción y prueba dentro del entorno de programación:

* **Prototipo de Interfaz de Usuario:** Incluye botones, campos de entrada de texto, y áreas de resultados, diseñados para facilitar la interacción del usuario con la aplicación. Las opciones de copiar texto, borrar texto, modo claro, y zoom también están implementadas.



Figura 2 Interfaz

* **Diagrama de Flujo de Traducción:** Describe el proceso de traducción desde la entrada del usuario hasta la salida en Braille o texto, con consideraciones para caracteres especiales y números.

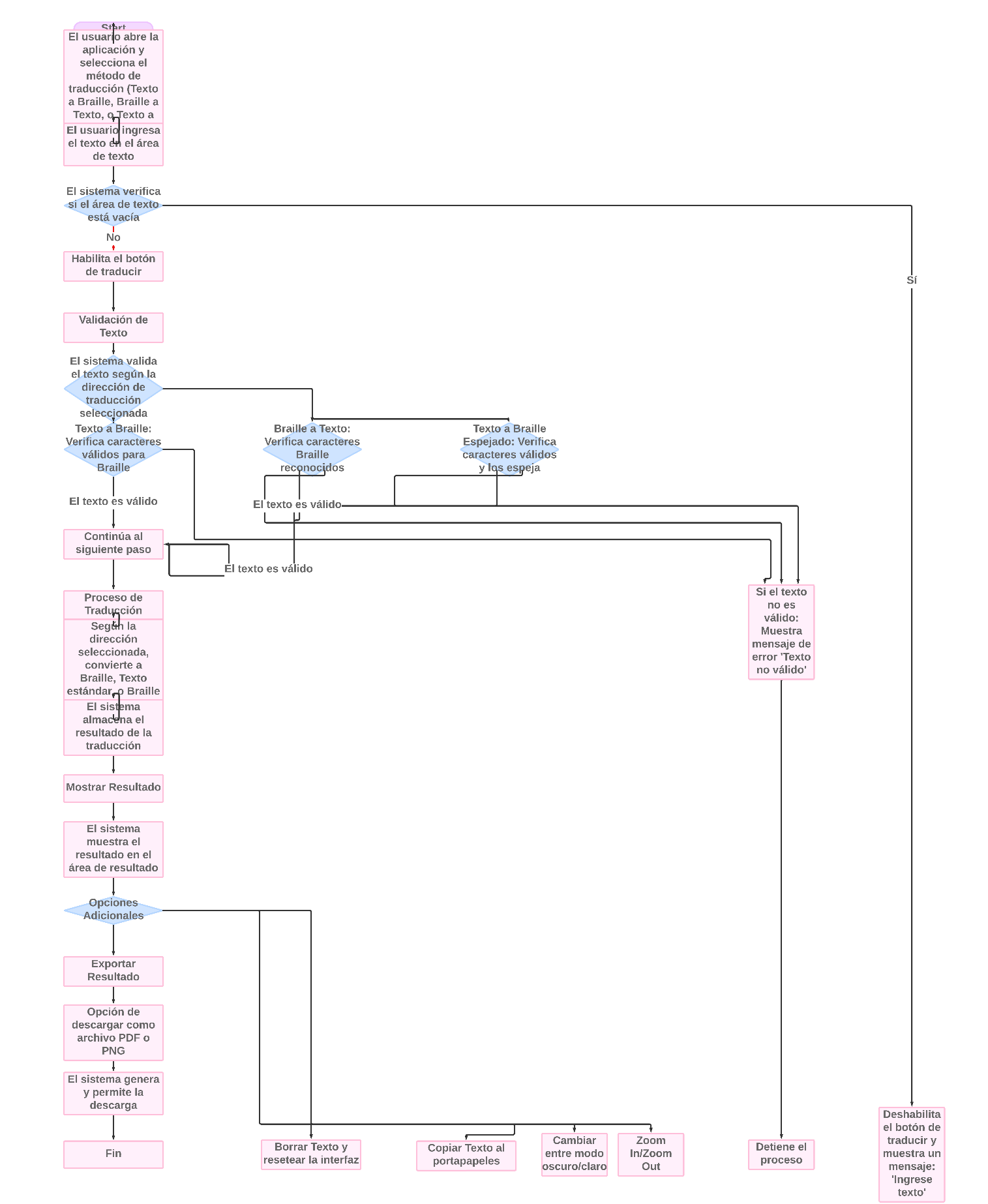


Figura 3 Diagrama de Flujo de Traducción

#### **3.2 Formato de Datos de Entrada/Salida:**

* **Entrada:**
  + Texto en formato estándar (alfabético, numérico, con signos básicos).
  + Texto en formato Braille.
* **Salida:**
  + Texto transcrito en Braille.
  + Texto transcrito de Braille a texto estándar.
  + Archivos de resultados en formatos PDF y PNG.



Figura 4 Ejemplo de traducción de Texto a Braille

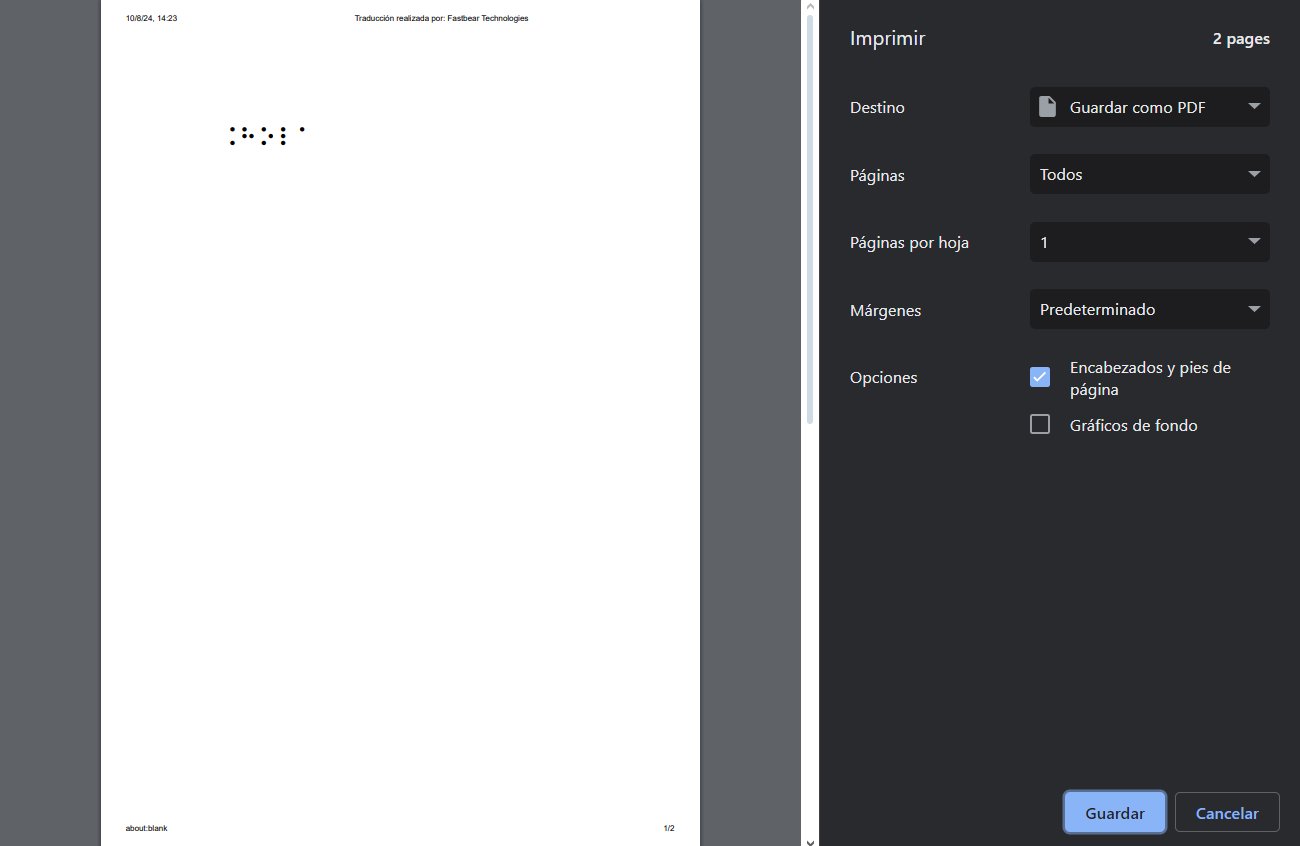


Figura 5 Descarga de traducción de Texto a Braille

#### **3.3 Especificaciones de Almacenamiento de Datos:**

El software utiliza diccionarios en memoria para almacenar los mapeos de caracteres de texto a Braille, permitiendo un acceso rápido y eficiente durante el proceso de traducción.

#### **3.4 Convenciones de Nombres de Datos:**

* **Entrada de Texto:** input\_text
* **Salida de Braille:** output\_braille
* **Archivo Exportado:** exported\_file

#### **3.5 Formato de las Estructuras de Datos Requeridas:**

* **Estructura de Texto:** Una cadena de texto que representa la entrada del usuario, la cual es validada y procesada por los módulos de traducción.
* **Estructura de Braille:** Una cadena de texto en formato Braille, generada como salida de las funciones de traducción.

#### **3.6 Campos de Datos y Propósito de Cada Elemento de Datos:**

* **Campo char\_id:** Identificador único de cada carácter en el diccionario.
* **Campo braille\_symbol:** Representación Braille asociada a un carácter específico.
* **Campo user\_preference:** Configuraciones guardadas del usuario para personalizar la experiencia de transcripción.

#### **3.7 Especificaciones de la Estructura del Programa:**

El programa está organizado en módulos y funciones, donde cada módulo maneja un aspecto específico de la aplicación. El modularidad del código permite futuras expansiones y facilita el mantenimiento. Las opciones adicionales (copiar, borrar, modo claro, zoom) están integradas de manera que no interfieran con las funciones principales, pero sí mejoren la experiencia del usuario.

### **Estados Aplicables:**

El diseño del software ha sido verificado, validado y establecido como base para la implementación final.