ESCUELA POLITÉCNICA NACIONAL

ISWD652 CALIDAD DE SOFTWARE

INTEGRANTES:	Eddy Arias Fredviner Bailón Max Carrión Milton Pástor Jonathan Poaquiza

FECHA:	05-06-2024
TEMA:	Documento de diseño

Descripción General:

El proyecto "BrailleTech" es una aplicación web diseñada para traducir texto entre español y Braille, generar señalética en Braille a partir de texto en español y crear impresiones en espejo para escritura manual.

Listado de Herramientas y Tecnologías de Desarrollo:

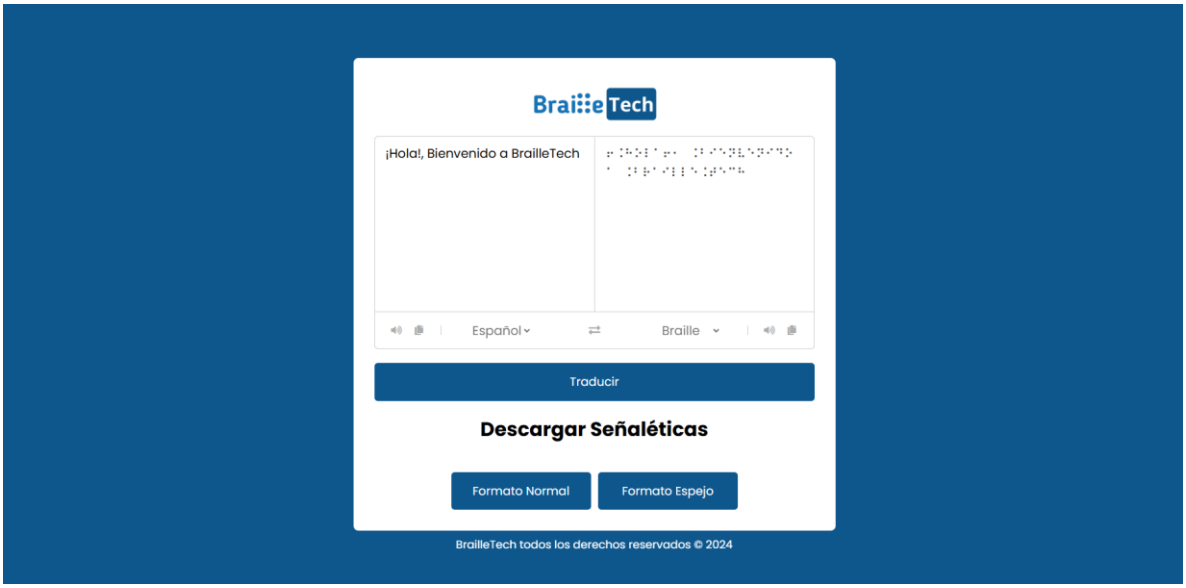
1. **Node.js y NPM:**
   - Utilizados para la gestión de dependencias y la ejecución de scripts de desarrollo.
   - Versión recomendada: Node.js 14.x y NPM 6.x.
2. **Webpack:**
   - Herramienta de empaquetado de módulos que compila y agrupa los recursos del proyecto.
   - Versión utilizada: 5.91.0.
3. **Jest:**
   - Framework de pruebas unitarias que asegura la calidad del código mediante pruebas automáticas.
   - Versión utilizada: 29.7.0.
4. **HTML5 y CSS3:**
   - HTML5 para la estructura de la interfaz de usuario.
   - CSS3 para los estilos y el diseño visual.
5. **JavaScript (ES6+):**

- Lenguaje de programación utilizado para implementar la lógica de la aplicación y la manipulación del DOM.

6. **html2canvas y jsPDF:**
   - html2canvas: Biblioteca que captura contenido de HTML y lo convierte en una imagen.
   - jsPDF: Biblioteca que genera archivos PDF a partir de contenido HTML.
   - Versiones utilizadas: html2canvas 1.4.1 y jsPDF 2.5.1.
7. **Babel:**
   - Transpilador que convierte el código ES6+ a una versión compatible con navegadores más antiguos.
   - Versión utilizada: @babel/core 7.24.6 y @babel/preset-env 7.24.6.

Estructura y Diseño del Producto

Diseño de Interfaz de Usuario para BrailleTech



Componentes de la Interfaz

1. **Encabezado:**
   - **Logo:** Un logotipo de la aplicación "BrailleTech" ubicado en la parte superior central de la página.
   - **Título de la página:** "Traductor de Braille".

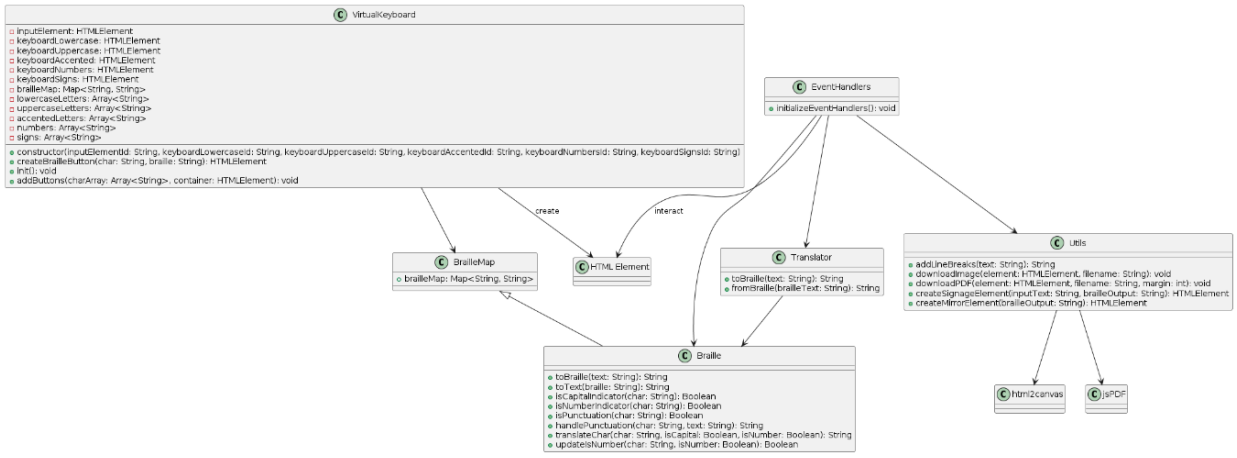
2. **Área de Traducción:**
    - **Caja de Texto de Entrada:**
        - Ubicada a la izquierda.
        - Etiquetada como "Español".
        - Área de texto donde el usuario puede ingresar el texto en español.
    - **Caja de Texto de Salida:**
        - Ubicada a la derecha.
        - Etiquetada como "Braille".
        - Área de texto donde se muestra el texto traducido a Braille.
        - Es de solo lectura.
3. **Controles de Traducción:**
    - **Botón de Intercambio de Idiomas:**
        - Ubicado entre las cajas de texto de entrada y salida.
        - Representado por un ícono de intercambio (flechas bidireccionales).
    - **Botones de Sonido:**
        - Ubicados al lado de cada caja de texto (entrada y salida).
        - Representados por un ícono de altavoz.
        - Permiten escuchar el contenido de la caja de texto correspondiente.
    - **Botones de Copiar:**
        - Ubicados al lado de cada caja de texto (entrada y salida).
        - Representados por un ícono de copiar.
        - Permiten copiar el contenido de la caja de texto correspondiente al portapapeles.
4. **Botón de Traducción:**
    - Ubicado debajo de las cajas de texto de entrada y salida.

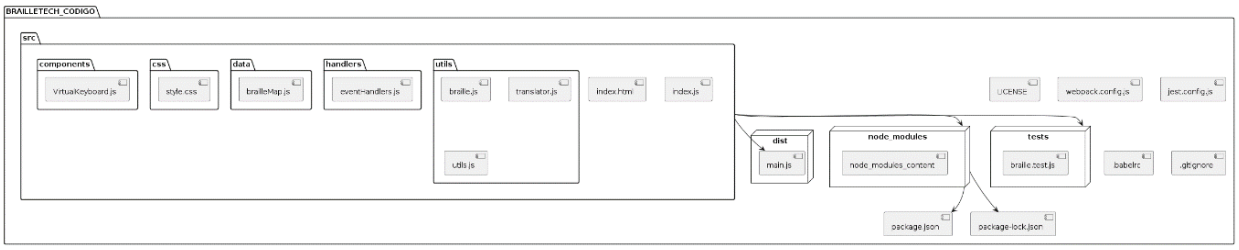
- Etiquetado como "Traducir".
- Ejecuta la traducción del texto ingresado.

5. **Sección de Descarga de Señaléticas:**
   - **Título de la Sección:** "Descargar Señaléticas".
   - **Botones de Descarga:**
     - **Formato Normal:** Botón para descargar la señalética en formato normal.
     - **Formato Espejo:** Botón para descargar la señalética en formato espejo.
6. **Pie de Página:**
   - Contiene la nota de derechos reservados: "BrailleTech todos los derechos reservados © 2024".

Modelo Arquitectónico

La estructura del proyecto está organizada en varios directorios y archivos clave:





Componentes Principales

Frontend

- **index.html:** Archivo HTML principal que define la estructura de la interfaz de usuario.
- **style.css:** Archivo de estilos CSS para la aplicación.
- **index.js:** Archivo JavaScript que inicializa la aplicación y maneja la lógica de la interfaz de usuario.
- **eventHandlers.js:** Archivo que contiene los manejadores de eventos para la interacción con el usuario.

Backend

- **braille.js:** Módulo que proporciona las funciones de conversión entre texto en español y Braille.
- **brailleMap.js:** Mapa de caracteres que define la relación entre los caracteres en español y sus equivalentes en Braille.
- **translator.js:** Módulo que utiliza braille.js para convertir texto a Braille y viceversa.
- **utils.js:** Módulo con funciones utilitarias, como la generación de señalética y la creación de elementos en espejo.

Configuración y Herramientas

- **package.json:** Archivo de configuración de NPM que define las dependencias del proyecto y los scripts disponibles.
- **webpack.config.js:** Configuración de Webpack para la compilación y el empaquetado del proyecto.
- **jest.config.js:** Configuración de Jest para la ejecución de pruebas unitarias.

Arquitectura del Sistema

El proyecto sigue una arquitectura de capas, donde cada capa tiene responsabilidades específicas. A continuación, se describe cada capa:

Capa de Presentación

- **Objetivo:** Proveer una interfaz de usuario interactiva.

- **Componentes:** index.html, style.css, index.js, eventHandlers.js.
- **Responsabilidades:** Manejar la entrada del usuario, actualizar la interfaz de acuerdo con las acciones del usuario, y enviar/recibir datos de la capa de lógica de negocio.

Capa de Lógica de Negocio

- **Objetivo:** Implementar la lógica de negocio del traductor de Braille.
- **Componentes:** braille.js, translator.js.
- **Responsabilidades:** Convertir texto entre español y Braille utilizando el mapa de caracteres, manejar reglas específicas de conversión, y proporcionar funciones para la generación de señalética y formatos en espejo.

Capa de Datos

- **Objetivo:** Gestionar la estructura de datos necesaria para las conversiones.
- **Componentes:** brailleMap.js.
- **Responsabilidades:** Definir y mantener la relación entre los caracteres en español y sus equivalentes en Braille.

Dependencias y Herramientas

- **Node.js y NPM:** Utilizados para la gestión de dependencias y la ejecución de scripts.
- **Webpack:** Herramienta de empaquetado de módulos utilizada para compilar y agrupar los recursos del proyecto.
- **Jest:** Framework de pruebas unitarias utilizado para garantizar la calidad del código.
- **html2canvas y jsPDF:** Bibliotecas utilizadas para generar imágenes y PDFs a partir de elementos HTML.

Flujo de Datos

1. **Entrada del Usuario:** El usuario ingresa texto en el área de texto en index.html.
2. **Interacción del Usuario:** El usuario presiona el botón "Traducir".
3. **Procesamiento de Datos:** eventHandlers.js envía el texto a las funciones de conversión en translator.js.
4. **Conversión:** translator.js utiliza braille.js y brailleMap.js para convertir el texto y devolver el resultado.

5. **Actualización de la UI:** eventHandlers.js actualiza el área de texto de salida con el resultado de la conversión.

Patrones de Diseño

- **Modularidad:** Cada función y su lógica correspondiente están encapsuladas en módulos específicos (braille.js, brailleMap.js, translator.js, utils.js).
- **Separación de Responsabilidades:** La lógica de presentación y la lógica de negocio están claramente separadas en diferentes capas.

REFERENCIAS

[1] M. Pástor, E. Arias, M. Carrion, J. Poaquiza y F. Bailón, «BrailleTech,» [En línea]. Available: https://github.com/MaxCar31/BrailleTech.