

INTEGRANTES:

BrailleTech

FECHA:

06-06-2024

TEMA:

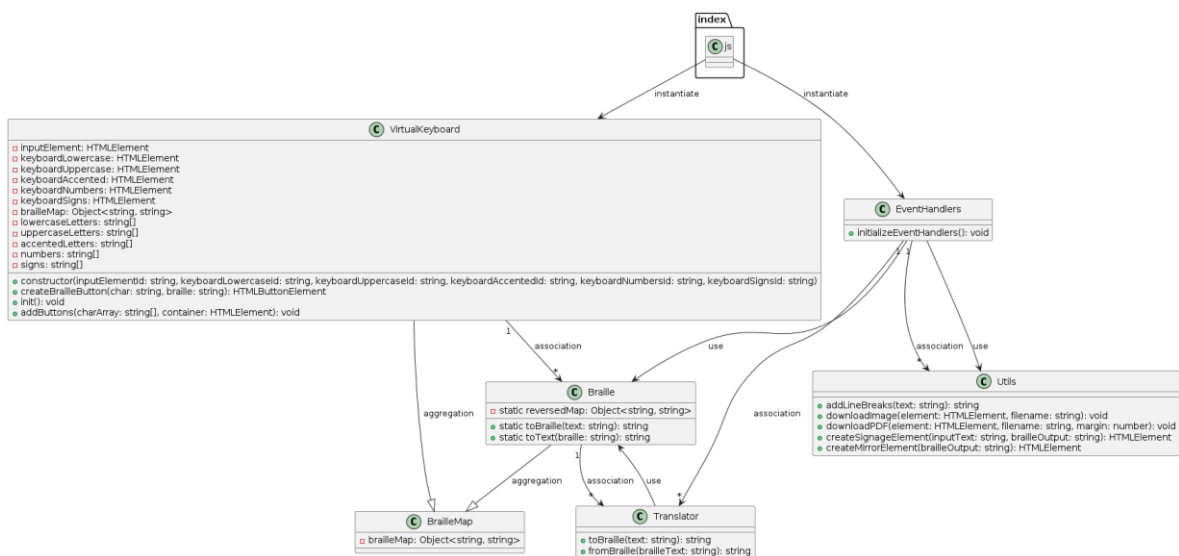
Desarrollo del producto versión 1.0.0

1.- Descripción General:

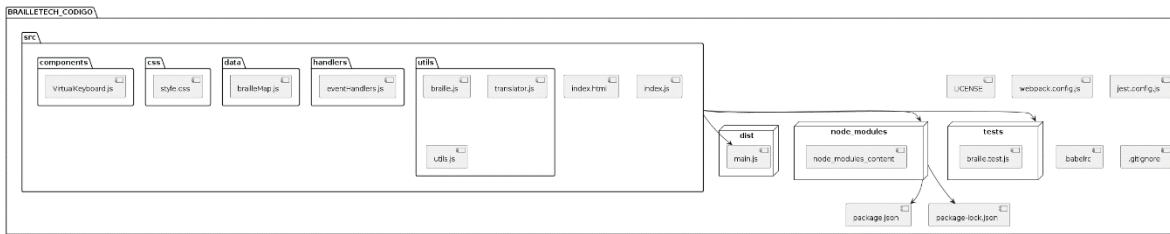
El proyecto "BrailleTech" es una aplicación web diseñada para traducir texto entre español y Braille, generar señalética en Braille a partir de texto en español y crear impresiones en espejo para escritura manual.

2.2.- Modelo Arquitectónico

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



ISWD652 CALIDAD DE SOFTWARE



3. Componentes Principales

3.1. 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.

3.2. 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.

3.3. 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

ISWD652 CALIDAD DE SOFTWARE

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

4.1. 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.

4.2. 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.

4.3. 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.

ISWD652 CALIDAD DE SOFTWARE

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.

De ser necesario revisar:

Manual de Instalación:

[BrailleTech/BrailleTech_Documentacion/Documentacion - Construccion/Manual de Instalacion - BrailleTech.pdf at main · MaxCar31/BrailleTech \(github.com\)](#)

Documentación del código generado a través de JSDocs:

[BrailleTech/BrailleTech_Documentacion/JSDoc at main · MaxCar31/BrailleTech \(github.com\)](#)

Manual de Usuario:

[BrailleTech/BrailleTech_Documentacion/Documentacion - Construccion/Manual de Usuario - BrailleTech.pdf at main · MaxCar31/BrailleTech \(github.com\)](#)