

ISWD652 CALIDAD DE SOFTWARE

Pruebas Unitarias

Se han creado algunas pruebas unitarias en las cuales probaremos los métodos de nuestras 3 clases principales las cuales son *brailleDictionary*, *traductor*, *validator*; :

Para **brailleDictionaryTest**:

getLetraBraille

- **Propósito:** Verificar que el método `getLetraBraille` devuelve el Braille correcto para un carácter dado.
- **Casos de prueba:**
 - Retorna el Braille "1" para la letra "a".
 - Retorna el Braille "12456" para la letra "ñ".
 - Verifica que el método es insensible a mayúsculas retornando "1" para "A".
 - Prueba un caso para una letra no incluida en el diccionario (debería retornar null según el propósito de la prueba).

getSigno

- **Propósito:** Verificar la conversión de signos de puntuación a Braille.
- **Casos de prueba:**
 - Retorna "3" para el punto ".".
 - Retorna "235" para el signo de exclamación "!".
 - Prueba un signo que no está en el diccionario (debería retornar null).

getNumeroBraille

- **Propósito:** Verificar la conversión de números a Braille en diferentes modos (computarizado y no computarizado).
- **Casos de prueba:**
 - Retorna "1" para el número 1 en modo no computarizado.
 - Retorna "16" para el número 1 en modo computarizado.
 - Prueba un número fuera del diccionario (debería retornar null).

getLetraEspañol

- **Propósito:** Verificar la conversión inversa de Braille a letras.
- **Casos de prueba:**
 - Retorna "a" para el Braille "1".
 - Retorna "ñ" para el Braille "12456".
 - Prueba un código Braille que no existe (debería retornar null).

getInvertSigno

- **Propósito:** Verificar la conversión inversa de Braille a signos.
- **Casos de prueba:**
 - Retorna "." para el Braille "3".
 - Retorna "@" para el Braille "5".

ISWD652 CALIDAD DE SOFTWARE

- Prueba un código Braille no incluido en el diccionario (debería retornar null).

getInvertNumeroBraille

- **Propósito:** Verificar la conversión inversa de Braille a números.
- **Casos de prueba:**
 - Retorna 1 para el Braille "1".
 - Retorna 1 para el Braille "16" en modo computarizado.
 - Prueba un código Braille no existente (debería retornar null).

Al ejecutar todas estas pruebas podemos ver que conseguimos que todas fueron aprobadas con éxito:

```
PS C:\Users\shander\Desktop\BrailleX\Version3-Final\implementation> npm test

> braillex@1.0.0 test
> mocha

BrailleDictionary
  getLetraBraille
    ✓ debería retornar "1" para la letra "a"
    ✓ debería retornar "12456" para la letra "ñ"
    ✓ debería retornar "1" para la letra "A" (mayúscula)
    ✓ debería retornar null para una letra que no está en el diccionario
  getSigno
    ✓ debería retornar "3" para el punto "."
    ✓ debería retornar "235" para el signo de exclamación "!"
    ✓ debería retornar null para un signo que no está en el diccionario
  getNumeroBraille
    ✓ debería retornar "1" para el número 1 (no computarizado)
    ✓ debería retornar "16" para el número 1 (computarizado)
    ✓ debería retornar null para un número que no está en el diccionario
  getLetraEspañol
    ✓ debería retornar "a" para el Braille "1"
    ✓ debería retornar "ñ" para el Braille "12456"
    ✓ debería retornar null para un Braille que no está en el diccionario
  getInvertSigno
    ✓ debería retornar "." para el Braille "3"
    ✓ debería retornar "@" para el Braille "5"
    ✓ debería retornar null para un Braille que no está en el diccionario
  getInvertNumeroBraille
    ✓ debería retornar 1 para el Braille "1"
    ✓ debería retornar 1 para el Braille "16" (computarizado)
    ✓ debería retornar null para un Braille que no está en el diccionario
```



- **Propósito:** Verificar que el método `traducirLineaABraille` traduce correctamente una línea completa de texto en español a Braille.

ISWD652 CALIDAD DE SOFTWARE

➤ Casos de prueba:

- Traduce la línea "hola" a Braille y espera "125 135 123 1".

obtenerBrailleDeCaracter

- **Propósito:** Verificar que el método obtenerBrailleDeCaracter obtiene correctamente el código Braille de un carácter específico.

➤ Casos de prueba:

- Obtiene el código Braille de la letra 'h' en el texto 'hola' en la posición 0 y espera "125".
- Obtiene el código Braille del signo de puntuación '.' en el texto 'hola.' en la posición 4 y espera "3".
- Obtiene el código Braille del espacio en el texto 'hola mundo' en la posición 4 y espera "" (vacío).

traducirLineaAEspanol

- **Propósito:** Verificar que el método traducirLineaAEspanol convierte correctamente una línea de Braille de vuelta a español.

➤ Casos de prueba:

- Traduce la línea de Braille "125 135 123 1" a español y espera "hola".

unicodeToBraille

- **Propósito:** Verificar que el método unicodeToBraille convierte correctamente Braille en Unicode a código Braille.

➤ Casos de prueba:

- Convierte el Braille Unicode "⠠⠠⠠⠠⠠⠠" a código Braille y espera "125 135 123 1".

Al ejecutar todas estas pruebas podemos ver que conseguimos que todas fueron aprobadas con éxito:

```
Traductor
  traducirEspanolABraille
    ✓debería traducir texto en español a Braille
    ✓debería traducir texto en español a Braille con mayúsculas
    ✓debería traducir texto en español a Braille con números
  traducirEspanolABrailleInverso
    ✓debería traducir texto en español a Braille invertido
  traducirBrailleAEspanol
    ✓debería traducir texto en Braille a español
    ✓debería traducir texto en Braille con mayúsculas a español
    ✓debería traducir texto en Braille con números a español
  convertirTextoABraille
    ✓debería convertir texto a Braille sin invertir
    ✓debería convertir texto a Braille invertido
  reordenarPuntosBraille
    ✓debería reordenar los puntos Braille
  traducirLineaABraille
    ✓debería traducir una línea de texto en español a Braille
  obtenerBrailleDeCaracter
    ✓debería obtener el código Braille de una letra
    ✓debería obtener el código Braille de un signo de puntuación
    ✓debería obtener el código Braille de un espacio
  traducirLineaAEspanol
    ✓debería traducir una línea de Braille a español
  unicodeToBraille
    ✓debería convertir Braille Unicode a código Braille
```

ISWD652 CALIDAD DE SOFTWARE

Para validateTest

isValidEmail

- **Propósito:** Verificar que el método isValidEmail valide correctamente las direcciones de correo electrónico.
- **Casos de prueba:**
 - **Prueba positiva:** Retorna true para la dirección de correo "test@example.com".
 - **Prueba negativa:** Retorna false para una dirección de correo inválida "invalid-email".

isValidURL

- **Propósito:** Verificar que el método isValidURL valide correctamente las URLs.
- **Casos de prueba:**
 - **Prueba positiva:** Retorna true para la URL "http://www.example.com".
 - **Prueba negativa:** Retorna false para una URL inválida "invalid-url".

isValidTag

- **Propósito:** Verificar que el método isValidTag valide correctamente las etiquetas.
- **Casos de prueba:**
 - **Prueba positiva:** Retorna true para una etiqueta válida "#validTag".
 - **Prueba negativa:** Retorna false para una etiqueta inválida "invalidTag".

isLetter

- **Propósito:** Verificar que el método isLetter identifique correctamente si un carácter es una letra.
- **Casos de prueba:**
 - **Prueba positiva:** Retorna true para el carácter 'a'.
 - **Prueba negativa:** Retorna false para el carácter '1'.

isNumber

- **Propósito:** Verificar que el método isNumber identifique correctamente si un carácter es un número.
- **Casos de prueba:**
 - **Prueba positiva:** Retorna true para el carácter '1'.

ISWD652 CALIDAD DE SOFTWARE

- **Prueba negativa:** Retorna false para el carácter 'a'.

isSpace

- **Propósito:** Verificar que el método isSpace identifique correctamente si un carácter es un espacio.
- **Casos de prueba:**
 - **Prueba positiva:** Retorna true para el carácter ' '.
 - **Prueba negativa:** Retorna false para el carácter 'a'.

isUpperCase

- **Propósito:** Verificar que el método isUpperCase identifique correctamente si una palabra está en mayúsculas.
- **Casos de prueba:**
 - **Prueba positiva:** Retorna true para la palabra "HELLO".
 - **Prueba negativa:** Retorna false para la palabra "Hello".

isDecimalPoint

- **Propósito:** Verificar que el método isDecimalPoint identifique correctamente si un punto está en la posición correcta de un número decimal.
- **Casos de prueba:**
 - **Prueba positiva:** Retorna true para un punto decimal en la posición 1 en el número "3.14".
 - **Prueba negativa:** Retorna false para un punto en una posición incorrecta (0 y 3) en el número "3.14".

Al ejecutar todas estas pruebas podemos ver que conseguimos que todas fueron aprobadas con éxito:



ISWD652 CALIDAD DE SOFTWARE

```
Validator
  isValidEmail
    ✓ debería retornar true para un correo válido
    ✓ debería retornar false para un correo inválido
  isValidURL
    ✓ debería retornar true para una URL válida
    ✓ debería retornar false para una URL inválida
  isValidTag
    ✓ debería retornar true para una etiqueta válida
    ✓ debería retornar false para una etiqueta inválida
  isLetter
    ✓ debería retornar true para un carácter que es una letra
    ✓ debería retornar false para un carácter que no es una letra
  isNumber
    ✓ debería retornar true para un carácter que es un número
    ✓ debería retornar false para un carácter que no es un número
  isSpace
    ✓ debería retornar true para un carácter que es un espacio
    ✓ debería retornar false para un carácter que no es un espacio
  isUpperCase
    ✓ debería retornar true para una palabra en mayúsculas
    ✓ debería retornar false para una palabra que no está en mayúsculas
  isDecimalPoint
    ✓ debería retornar true para un punto decimal en un número
    ✓ debería retornar false para un punto que no es decimal
```

51 passing (174ms)

PS C:\Users\shander\Desktop\BrailleX\Version3-Final\implementation> |