**HISTORIAS DE USUARIO**

|  |  |  |
| --- | --- | --- |
| **Nro:** HU-001 | **Título: Traducción de números arábigos occidentales a Braille** | **Prioridad:** Alta  **Estimación:** 7 horas |
| **Historia de usuario:**  Como usuario interesado en el lenguaje braille, quiero ingresar una cantidad numérica del 1 al 8000 a lenguaje braille para producir a bajo costo señalética o rotulación braille. | | |
| **Sabré que he terminado cuando se cumplan todos los criterios de aceptación a continuación:**   1. **Precisión de traducción:** Dada una cantidad numérica válida, cuando el usuario solicita la traducción entonces la traducción mostrada por la aplicación debe corresponder al valor numérico ingresado. 2. **Tolerancia a errores de traducción:** Dado una entrada inválida, como una cantidad numérica mayor a 8000, cuando el usuario ingresa la cantidad numérica entonces la aplicación debe mostrar un mensaje de error. | | |
| **Tareas de implementación:**   1. Crear un diccionario numérico a braille (1 hora). 2. Diseñar el algoritmo de traducción de números a braille (1 hora). 3. Diseñar la interfaz de usuario (1 hora). 4. Implementar la interfaz de usuario (2 horas). 5. Codificar un método para validar si el número ingresado este dentro del rango permitido (1/2 hora). 6. Codificar el despliegue de un mensaje de error en caso de que el número ingresado sea inválido (1/2 hora). 7. Codificar un método que implemente el algoritmo de traducción utilizando el diccionario (1/2 hora). 8. Probar con datos de prueba válidos e inválidos (1/2 hora). | | |

|  |  |  |
| --- | --- | --- |
| **Nro:** HU-002 | **Título: Traducción del alfabeto español a Braille** | **Prioridad:** Alta  **Estimación:** 6 horas |
| **Historia de usuario:**  Como usuario interesado en el lenguaje braille, quiero ingresar una cadena de caracteres alfabéticos en español con o sin espacios en blanco a braille para producir a bajo costo señalética o rotulación braille. | | |
| **Sabré que he terminado cuando se cumplan todos los criterios de aceptación a continuación:**   1. **Precisión de traducción:** Dada una cadena de caracteres alfabéticos con o sin espacios en blanco válida, cuando el usuario solicita la traducción entonces la traducción mostrada por la aplicación debe ser gramaticalmente correcta y corresponder a la cadena de texto ingresada. 2. **Tolerancia a errores de traducción:** Dado una entrada inválida, como una cadena de caracteres alfabéticos no españoles con o sin espacios en blanco, cuando el usuario ingresa la cadena entonces la aplicación debe mostrar un mensaje de error. | | |
| **Tareas de implementación:**   1. Crear un diccionario del alfabeto español a braille (2 horas). 2. Diseñar el algoritmo de traducción del alfabeto español a braille (1 hora). 3. Codificar un método para validar si la cadena de caracteres alfabéticos con o sin espacios en blanco ingresada sea correcta (1/2 hora). 4. Codificar el despliegue de un mensaje de error en caso de que la cadena sea inválida (1/2 hora). 5. Codificar un método que implemente el algoritmo de traducción utilizando el diccionario (1 hora). 6. Probar con datos de prueba válidos e inválidos (1 hora). | | |

|  |  |  |
| --- | --- | --- |
| **Nro:** HU-003 | **Título: Traducción de vocales acentuadas en español a Braille** | **Prioridad:** Alta  **Estimación:** 3 horas |
| **Historia de usuario:**  Como usuario interesado en el lenguaje braille, quiero ingresar un caracter; “á”, “é”, “í”, “ó”, “ú”, “ü”, en cadenas de texto en español a braille para producir a bajo costo señalética o rotulación braille. | | |
| **Sabré que he terminado cuando se cumplan todos los criterios de aceptación a continuación:**   1. **Precisión de traducción:** Dado un caracter acentuado en una cadena de texto, cuando el usuario solicita la traducción entonces la traducción mostrada por la aplicación debe ser gramaticalmente correcta y corresponder a la cadena de texto ingresada. 2. **Tolerancia a errores de traducción:** Dado una entrada inválida, como un caracter à, cuando el usuario ingresa el caracter entonces la aplicación debe mostrar un mensaje de error. | | |
| **Tareas de implementación:**   1. Actualizar el diccionario del alfabeto español con las vocales acentuadas a braille (1/2 hora). 2. Actualizar el algoritmo de traducción del alfabeto español a braille (1/2 hora). 3. Codificar un método para validar si un caracter dentro de una cadena de texto es correcta (1/2 hora). 4. Codificar el despliegue de un mensaje de error en caso de que el caracter sea inválido (1/2 hora). 5. Actualizar el método que implementa el algoritmo de traducción utilizando el diccionario (1/2 hora). 6. Probar con datos de prueba válidos e inválidos (1/2 hora). | | |

|  |  |  |
| --- | --- | --- |
| **Nro:** HU-004 | **Título: Traducción de números decimales a Braille** | **Prioridad:** Alta  **Estimación:** 3 horas |
| **Historia de usuario:**  Como usuario interesado en el lenguaje Braille, quiero ingresar una cantidad numérica decimal utilizando el caracter “.” o el caracter “,” como separador decimal para producir a bajo costo señalética o rotulación braille. | | |
| **Sabré que he terminado cuando se cumplan todos los criterios de aceptación a continuación:**   1. **Precisión de traducción:** Dada una cantidad numérica decimal con el caracter “.” o el caracter “,”, cuando el usuario solicita la traducción entonces la traducción mostrada por la aplicación debe corresponder al valor numérico decimal ingresado. 2. **Tolerancia a errores de traducción:** Dado una entrada inválida, como una cantidad numérica decimal con el caracter “;”, cuando el usuario ingresa la cantidad numérica decimal entonces la aplicación debe mostrar un mensaje de error. | | |
| **Tareas de implementación:**   1. Actualizar el diccionario numérico con los caracteres “.” y “,” a braille (1/2 hora). 2. Actualizar el algoritmo de traducción de números a braille (1/2 hora). 3. Codificar un método para validar si una cantidad numérica decimal tiene el caracter “.” ó el caracter “,” (1/2 hora). 4. Codificar el despliegue de un mensaje de error en caso de que la cantidad numérica decimal sea inválida (1/2 hora). 5. Actualizar el método que implementa el algoritmo de traducción utilizando el diccionario (1/2 hora). 6. Probar con datos de prueba válidos e inválidos (1/2 hora). | | |

|  |  |  |
| --- | --- | --- |
| **Nro:** HU-005 | **Título: Traducción de números en Braille a números arábigos occidentales** | **Prioridad:** Alta  **Estimación:** 3 horas |
| **Historia de usuario:**  Como usuario interesado en el lenguaje braille, quiero ingresar una cantidad numérica del “⠼⠈” al “⠼⠓⠚⠚⠚” a números arábigos occidentales para producir a bajo costo señalética o rotulación braille. | | |
| **Sabré que he terminado cuando se cumplan todos los criterios de aceptación:**   1. **Precisión de traducción:** Dada una cantidad numérica válida, cuando el usuario solicita la traducción entonces la traducción mostrada por la aplicación debe corresponder al valor numérico ingresado. 2. **Tolerancia a errores de traducción:** Dado una entrada inválida, como una cantidad numérica mayor a “⠼⠓⠚⠚⠚”, cuando el usuario ingresa la cantidad numérica entonces la aplicación debe mostrar un mensaje de error. | | |
| **Tareas de implementación:**   1. Diseñar el algoritmo de traducción de números braille a números arábigos occidentales (1 hora). 2. Codificar un método para validar si el número ingresado este dentro del rango permitido (1/2 hora). 3. Codificar el despliegue de un mensaje de error en caso de que el número ingresado sea inválido (1/2 hora). 4. Codificar un método que implemente el algoritmo de traducción utilizando el diccionario (1/2 hora). 5. Probar con datos de prueba válidos e inválidos (1/2 hora). | | |

|  |  |  |
| --- | --- | --- |
| **Nro:** HU-006 | **Título: Traducción del alfabeto Braille a español** | **Prioridad:** Alta  **Estimación:** 4 horas |
| **Historia de usuario:**  Como usuario interesado en el lenguaje braille, quiero ingresar una cadena de caracteres alfabéticos en braille con o sin espacios en blanco a español para producir a bajo costo señalética o rotulación braille. | | |
| **Sabré que he terminado cuando se cumplan todos los criterios de aceptación:**   1. **Precisión de la traducción:** Dada una cadena de caracteres alfabéticos braille con o sin espacios en blanco válida, cuando el usuario solicita la traducción entonces la traducción mostrada por la aplicación debe ser gramaticalmente correcta y corresponder a la cadena de texto ingresada. 2. **Tolerancia a errores de traducción:** Dado una entrada inválida, como una cadena de caracteres alfabéticos en español, cuando el usuario ingresa la cadena entonces la aplicación debe mostrar un mensaje de error. | | |
| **Tareas de implementación:**   1. Diseñar el algoritmo de traducción del alfabeto braille a español (1 hora). 2. Codificar un método para validar si la cadena de caracteres alfabéticos con o sin espacios en blanco ingresada sea correcta (1/2 hora). 3. Codificar el despliegue de un mensaje de error en caso de que la cadena sea inválida (1/2 hora). 4. Codificar un método que implemente el algoritmo de traducción utilizando el diccionario (1 hora). 5. Probar con datos de prueba válidos e inválidos (1 hora). | | |

|  |  |  |
| --- | --- | --- |
| **Nro:** HU-007 | **Título: Traducción de vocales acentuadas en Braille a español** | **Prioridad:** Alta  **Estimación:** 3 horas |
| **Historia de usuario:**  Como usuario interesado en el lenguaje braille, quiero ingresar un caracter; “⠷”, “⠮”, “⠌”, “⠬”, “⠾”, “⠳”, en cadenas de texto braille a español para producir a bajo costo señalética o rotulación braille. | | |
| **Sabré que he terminado cuando se cumplan todos los criterios de aceptación:**   1. **Precisión de la traducción:** Dado un caracter; “⠷”, “⠮”, “⠌”, “⠬”, “⠾”, “⠳”, en una cadena de texto, cuando el usuario solicita la traducción entonces la traducción mostrada por la aplicación debe ser gramaticalmente correcta y corresponder a la cadena de texto ingresada. 2. **Tolerancia a errores de traducción:** Dado una entrada inválida, como un caracter à, cuando el usuario ingresa el caracter entonces la aplicación debe mostrar un mensaje de error. | | |
| **Tareas de implementación:**   1. Actualizar el algoritmo de traducción del alfabeto braille a español (1/2 hora). 2. Codificar un método para validar si un caracter dentro de una cadena de texto es correcta (1/2 hora). 3. Codificar el despliegue de un mensaje de error en caso de que el caracter sea inválido (1/2 hora). 4. Actualizar el método que implementa el algoritmo de traducción utilizando el diccionario (1/2 hora). 5. Probar con datos de prueba válidos e inválidos (1/2 hora). | | |

|  |  |  |
| --- | --- | --- |
| **Nro:** HU-008 | **Título: Traducción de números decimales en Braille a español** | **Prioridad:** Alta  **Estimación:** 3 horas |
| **Historia de usuario:**  Como usuario interesado en el lenguaje Braille, quiero ingresar una cantidad numérica decimal utilizando el caracter “⠄” o el caracter “⠂” como separador decimal para producir a bajo costo señalética o rotulación braille. | | |
| **Sabré que he terminado cuando se cumplan todos los criterios de aceptación:**   1. **Precisión de la traducción:** Dada una cantidad numérica decimal con el caracter “⠄” o el caracter “⠂”, cuando el usuario solicita la traducción entonces la traducción mostrada por la aplicación debe corresponder al valor numérico decimal ingresado. 2. **Tolerancia a errores de traducción:** Dado una entrada inválida, como una cantidad numérica decimal con el caracter “.”, cuando el usuario ingresa la cantidad numérica decimal entonces la aplicación debe mostrar un mensaje de error. | | |
| **Tareas de implementación:**   1. Actualizar el algoritmo de traducción de números a braille (1 hora). 2. Codificar un método para validar si una cantidad numérica decimal tiene el caracter “.” ó el caracter “,” (1/2 hora). 3. Codificar el despliegue de un mensaje de error en caso de que la cantidad numérica decimal sea inválida (1/2 hora). 4. Actualizar el método que implementa el algoritmo de traducción utilizando el diccionario (1/2 hora). 5. Probar con datos de prueba válidos e inválidos (1/2 hora). | | |

|  |  |  |
| --- | --- | --- |
| **Nro:** HU-009 | **Título: Traducción de números arábigos occidentales a Braille en espejo** | **Prioridad:** Alta  **Estimación:** 4 horas |
| **Historia de usuario:**  Como usuario interesado en el lenguaje braille, quiero ingresar una cantidad numérica del 1 al 8000 a lenguaje braille en espejo para producir a bajo costo señalética o rotulación braille. | | |
| **Sabré que he terminado cuando se cumplan todos los criterios de aceptación a continuación:**   1. **Precisión de traducción:** Dada una cantidad numérica válida, cuando el usuario solicita la traducción entonces la traducción mostrada por la aplicación debe corresponder al valor numérico ingresado. 2. **Tolerancia a errores de traducción:** Dado una entrada inválida, como una cantidad numérica mayor a 8000, cuando el usuario ingresa la cantidad numérica entonces la aplicación debe mostrar un mensaje de error. | | |
| **Tareas de implementación:**   1. Crear un diccionario numérico a braille en espejo (1 hora). 2. Diseñar el algoritmo de traducción de números a braille en espejo (1 hora). 3. Codificar un método para validar si el número ingresado este dentro del rango permitido (1/2 hora). 4. Codificar el despliegue de un mensaje de error en caso de que el número ingresado sea inválido (1/2 hora). 5. Codificar un método que implemente el algoritmo de traducción utilizando el diccionario (1/2 hora). 6. Probar con datos de prueba válidos e inválidos (1/2 hora). | | |

|  |  |  |
| --- | --- | --- |
| **Nro:** HU-010 | **Título: Traducción del alfabeto español a Braille en espejo** | **Prioridad:** Alta  **Estimación:** 6 horas |
| **Historia de usuario:**  Como usuario interesado en el lenguaje braille, quiero ingresar una cadena de caracteres alfabéticos en español con o sin espacios en blanco a braille en espejo para producir a bajo costo señalética o rotulación braille. | | |
| **Sabré que he terminado cuando se cumplan todos los criterios de aceptación a continuación:**   1. **Precisión de traducción:** Dada una cadena de caracteres alfabéticos con o sin espacios en blanco válida, cuando el usuario solicita la traducción entonces la traducción mostrada por la aplicación debe ser gramaticalmente correcta y corresponder a la cadena de texto ingresada. 2. **Tolerancia a errores de traducción:** Dado una entrada inválida, como una cadena de caracteres alfabéticos no españoles con o sin espacios en blanco, cuando el usuario ingresa la cadena entonces la aplicación debe mostrar un mensaje de error. | | |
| **Tareas de implementación:**   1. Crear un diccionario del alfabeto español a braille en espejo (2 horas). 2. Diseñar el algoritmo de traducción del alfabeto español a braille en espejo (1 hora). 3. Codificar un método para validar si la cadena de caracteres alfabéticos con o sin espacios en blanco ingresada sea correcta (1/2 hora). 4. Codificar el despliegue de un mensaje de error en caso de que la cadena sea inválida (1/2 hora). 5. Codificar un método que implemente el algoritmo de traducción utilizando el diccionario (1 hora). 6. Probar con datos de prueba válidos e inválidos (1 hora). | | |

|  |  |  |
| --- | --- | --- |
| **Nro:** HU-011 | **Título: Traducción de vocales acentuadas en español a Braille en espejo** | **Prioridad:** Alta  **Estimación:** 3 horas |
| **Historia de usuario:**  Como usuario interesado en el lenguaje braille, quiero ingresar un caracter; “á”, “é”, “í”, “ó”, “ú”, “ü”, en cadenas de texto en español a braille en espejo para producir a bajo costo señalética o rotulación braille. | | |
| **Sabré que he terminado cuando se cumplan todos los criterios de aceptación a continuación:**   1. **Precisión de traducción:** Dado un caracter acentuado en una cadena de texto, cuando el usuario solicita la traducción entonces la traducción mostrada por la aplicación debe ser gramaticalmente correcta y corresponder a la cadena de texto ingresada. 2. **Tolerancia a errores de traducción:** Dado una entrada inválida, como un caracter à, cuando el usuario ingresa el caracter entonces la aplicación debe mostrar un mensaje de error. | | |
| **Tareas de implementación:**   1. Actualizar el diccionario del alfabeto español con las vocales acentuadas a braille en espejo (1/2 hora). 2. Actualizar el algoritmo de traducción del alfabeto español a braille en espejo (1/2 hora). 3. Codificar un método para validar si un caracter dentro de una cadena de texto es correcta (1/2 hora). 4. Codificar el despliegue de un mensaje de error en caso de que el caracter sea inválido (1/2 hora). 5. Actualizar el método que implementa el algoritmo de traducción utilizando el diccionario (1/2 hora). 6. Probar con datos de prueba válidos e inválidos (1/2 hora). | | |

|  |  |  |
| --- | --- | --- |
| **Nro:** HU-012 | **Título: Traducción de números decimales a Braille en espejo** | **Prioridad:** Alta  **Estimación:** 3 horas |
| **Historia de usuario:**  Como usuario interesado en el lenguaje Braille, quiero ingresar una cantidad numérica decimal utilizando el caracter “.” o el caracter “,” como separador decimal para producir a bajo costo señalética o rotulación braille. | | |
| **Sabré que he terminado cuando se cumplan todos los criterios de aceptación a continuación:**   1. **Precisión de traducción:** Dada una cantidad numérica decimal con el caracter “.” o el caracter “,”, cuando el usuario solicita la traducción entonces la traducción mostrada por la aplicación debe corresponder al valor numérico decimal ingresado. 2. **Tolerancia a errores de traducción:** Dado una entrada inválida, como una cantidad numérica decimal con el caracter “;”, cuando el usuario ingresa la cantidad numérica decimal entonces la aplicación debe mostrar un mensaje de error. | | |
| **Tareas de implementación:**   1. Actualizar el diccionario numérico con los caracteres “.” y “,” a braille en espejo (1/2 hora). 2. Actualizar el algoritmo de traducción de números a braille en espejo (1/2 hora). 3. Codificar un método para validar si una cantidad numérica decimal tiene el caracter “.” ó el caracter “,” (1/2 hora). 4. Codificar el despliegue de un mensaje de error en caso de que la cantidad numérica decimal sea inválida (1/2 hora). 5. Actualizar el método que implementa el algoritmo de traducción utilizando el diccionario (1/2 hora). 6. Probar con datos de prueba válidos e inválidos (1/2 hora). | | |

|  |  |  |
| --- | --- | --- |
| **Nro:** HU-013 | **Título: Integrar teclado español en pantalla** | **Prioridad:** Baja  **Estimación:** 5 horas |
| **Historia de usuario:**  Como usuario interesado en el lenguaje Braille, quiero acceder a un teclado español con caracteres alfabéticos, numéricos y especiales, en pantalla para facilitar la producción a bajo costo de señalética o rotulación braille. | | |
| **Sabré que he terminado cuando se cumplan todos los criterios de aceptación a continuación:**   1. **Disponibilidad:** Dada la pantalla principal, cuando un usuario ingresa al traductor entonces el teclado debe aparecer y permanecer hasta cerrar el traductor. 2. **Facilidad de uso:** Dada la pantalla principal, cuando un usuario ingresa al traductor entonces el teclado debe estar bien etiquetado y seccionado. 3. **Consistencia:** Dada la pantalla principal, cuando un usuario ingresa información al traductor entonces la tecla seleccionada debe imprimir el caracter etiquetado. | | |
| **Tareas de implementación:**   1. Diseñar el algoritmo que permita imprimir el teclado español en pantalla (2 hora). 2. Codificar el método que implemente algoritmo del teclado español en pantalla (2 horas). 3. Probar con traducciones e ingresos válidos e inválidos (1 hora). | | |

|  |  |  |
| --- | --- | --- |
| **Nro:** HU-014 | **Título: Integrar teclado Braille en pantalla** | **Prioridad:** Alta  **Estimación:** 5 horas |
| **Historia de usuario:**  Como usuario interesado en el lenguaje Braille, quiero acceder a un teclado braille con caracteres alfabéticos, numéricos y especiales, en pantalla para producir a bajo costo señalética o rotulación braille. | | |
| **Sabré que he terminado cuando se cumplan todos los criterios de aceptación a continuación:**   1. **Disponibilidad:** Dada la pantalla principal, cuando un usuario ingresa al traductor entonces el teclado debe aparecer y permanecer hasta cerrar el traductor. 2. **Facilidad de uso:** Dada la pantalla principal, cuando un usuario ingresa al traductor entonces el teclado debe estar bien etiquetado y seccionado. 3. **Consistencia:** Dada la pantalla principal, cuando un usuario ingresa información al traductor entonces la tecla seleccionada debe imprimir el caracter etiquetado. | | |
| **Tareas de implementación:**   1. Diseñar el algoritmo que permita imprimir el teclado braille en pantalla (2 hora). 2. Codificar el método que implemente algoritmo del teclado braille en pantalla (2 horas). 3. Probar con traducciones e ingresos válidos e inválidos (1 hora). | | |