

Documento de Requisitos TE-K

Elaborado por Squad Code

Integrantes:

Kevin Joel Cóndor Quispe

Rafaela Nicole Robalino Oquendo

Jackson Alexander Ramón Ramón

Jean Pierre Villacis Rivera

1 Introducción

El propósito del siguiente documento es describir el funcionamiento que se espera conseguir del traductor de kichwa mediante la creación de historias de usuario con criterios de aceptación.

2 Historias de usuario

En este apartado se describirán las historias de usuario que dirigirán el desarrollo del software. De igual manera, para cada historia se crearán los criterios de aceptación para establecer las condiciones que deberán ser satisfechas por el traductor y así lograr la aceptación de los clientes.

2.1 Historia de usuario 1

No: 1	Título: Interfaz gráfica simple.
Historia de usuario: Como usuario del sistema quiero tener una interfaz gráfica con elementos simples para permitir la traducción de cantidades numéricas a kichwa y comunicar la importancia de este idioma en Ecuador.	
Criterios de aceptación: <ol style="list-style-type: none">1. Comprobar que el programa se desarrolle correctamente en diferentes sistemas operativos como Windows, Linux y Mac OS. (Portabilidad: Adaptabilidad)2. Asegurarse de que la interfaz del programa es lo suficiente simple que permita su uso de manera intuitiva sin la necesidad de tener un manual de respaldo. (Usabilidad: Estética de la Interfaz de Usuario)	

2.2 Historia de usuario 2

No: 2	Título: Traducir únicamente números.
Historia de usuario: Como usuario del sistema quiero poder ingresar únicamente números al traductor para evitar la generación de resultados erróneos o poco entendibles.	
Criterios de aceptación: <ol style="list-style-type: none">1. Comprobar que el traductor no interpreta el ingreso de otro tipo de caracteres que no sean números enteros. (Usabilidad: Protección Contra Errores de Usuario)2. Asegurarse de que el traductor esté preparado para el caso en el que se quiera traducir sin haber ingresado un número antes. (Fiabilidad: Tolerancia a Fallos)3. Asegurarse de que las traducciones se realicen de manera rápida (tiempo menor a 1 segundo) y sin errores. (Eficiencia de Rendimiento: Comportamiento Temporal)	

2.3 Historia de usuario 3

No: 3	Título: Borrar el contenido de las traducciones.
Historia de usuario: Como usuario del sistema quiero contar con la función de borrado para mantener la interfaz del programa limpia de textos y agilizar las traducciones.	
Criterios de aceptación: <ol style="list-style-type: none">1. Comprobar que las áreas de texto de la interfaz del traductor queden en blanco al usar la función de borrado. (Adecuación Funcional: Completitud Funcional)2. Comprobar que los resultados borrados de las traducciones no se acumulen y se presenten como parte del resultado de nuevas traducciones. (Adecuación Funcional: Correctitud Funcional)3. Asegurarse de que el traductor funcione correctamente al momento de usar la función de borrar cuando no exista ningún contenido en las áreas de texto del traductor. (Fiabilidad: Tolerancia a Fallos)	

2.4 Historia de usuario 4

No: 4	Título: Controlar el límite de traducción.
Historia de usuario: Como usuario del sistema quiero que se me comunique si un número ingresado pasa el límite aceptado (menor o igual a 9999) por el traductor para poder discriminar rápidamente los números que se vayan a traducir.	
Criterios de aceptación: <ol style="list-style-type: none">1. Comprobar que el programa notifique que un número pasa el límite aceptado al momento de querer traducirlo. (Fiabilidad: Tolerancia a Fallos)2. Asegurarse de que el traductor muestre las notificaciones de manera rápida (tiempo menor a 1 segundo) y sin errores. (Eficiencia de Rendimiento: Comportamiento Temporal)	

2.5 Historia de usuario 5

No: 5	Título: Comentar el código fuente.
Historia de usuario: Como usuario técnico del sistema quiero tener comentarios en el código del programa para poder entender el propósito de las líneas o bloques de código y así facilitar futuros procesos de mantenimiento.	
Criterios de aceptación: <ol style="list-style-type: none">1. Comprobar que el código del programa esté comentado al momento de abrir el proyecto dentro del entorno de desarrollo integrado (IDE). (Adecuación Funcional: Completitud Funcional)2. Asegurarse de que los comentarios existentes o que se vayan a agregar estén correctamente ubicados en el código. (Adecuación Funcional: Completitud Funcional)3. Asegurarse de que los comentarios existentes o que se vayan a agregar muestren información relevante que facilite la comprensión a los desarrolladores. (Adecuación Funcional: Pertinencia Funcional)	