

UNIVERSIDAD CATÓLICA SANTO TORIBIO DE MOGROVEJO
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA DE SISTEMAS Y COMPUTACIÓN



**TUTOR INTELIGENTE PARA APOYAR EN EL APRENDIZAJE DE LA ESCRITURA
DEL IDIOMA QUECHUA A PROFESORES HISPANOABLANTES EN UN COLEGIO DE
INCAHUASI**

**TESIS PARA OPTAR EL TÍTULO DE
INGENIERO DE SISTEMAS Y COMPUTACIÓN**

AUTOR

Silvia Mariela Casiano Inga

ASESOR

Marlon Eugenio Vilchez Rivas

<https://orcid.org/0000-0003-2979-0731>

Chiclayo, 2022

**TUTOR INTELIGENTE PARA APOYAR EN EL APRENDIZAJE DE LA
ESCRITURA DEL IDIOMA QUECHUA A PROFESORES
HISPANOHABLANTES EN UN COLEGIO DE INCAHUASI**

PRESENTADA POR
Silvia Mariela Casiano Inga

A la Facultad de Ingeniería de la
Universidad Católica Santo Toribio de Mogrovejo
para optar el título de

INGENIERO DE SISTEMAS Y COMPUTACIÓN

APROBADA POR

María Ysabel Arangui García
PRESIDENTE

Mariana Chavarry Chankay
SECRETARIO

Marlon Eugenio Vilchez Rivas
VOCAL

Dedicatoria

A Dios por haber sido hasta ahora quien me brinda las fuerzas para seguir adelante y no rendirme, es por eso que todo se lo debo a Él.

A mis padres Asunción y Paula por su apoyo incondicional, su confianza, su amor y los ánimos para no rendirme.

Agradecimientos

A Dios por brindarme sabiduría y su guía en todo el camino de mi vida universitaria.

A mis padres que siempre me han ayudado en todo lo que me propongo.

A una persona especial en mi vida que siempre estuvo dándome ánimos para no rendirme y seguir adelante.

A mis asesores; Ing. Raúl Huarote Zegarra, Ing. Marlon Vílchez Rivas y a la Ing. Jury Aquino Trujillo; por su apoyo, tiempo, paciencia, sugerencias, consejos y compartir sus conocimientos para llevar a cabo esta investigación con éxito.

A la institución Educativa especialmente al director, quien brindó información necesaria para llevar a cabo este proyecto y que fuera posible el desarrollo de este software.

Índice

Resumen	5
Abstract	6
Introducción.....	7
Revisión de literatura.....	9
Materiales y métodos	13
Resultados y discusión	16
Conclusiones	31
Recomendaciones	31
Referencias.....	32
Anexos	34

Resumen

En la zona de Incahuasi, provincia de Ferreñafe, región de Lambayeque, existe un problema de comunicación entre profesores y alumnos, en los colegios de educación básica regular (EBR), debido a que los profesores no tienen formación en el idioma quechua de la localidad y trae como consecuencia que los alumnos no puedan comprender los contenidos y conocimientos que el maestro imparte, sobre todo los estudiantes de nivel primaria. Es por eso que la siguiente investigación se realizó con el objetivo de apoyar en el aprendizaje de la escritura del idioma quechua a los profesores hispanohablantes de un colegio de Incahuasi. El software se desarrolló siguiendo cada fase de la Metodología MeISE para concretar su fin educativo, haciendo una aplicación interactiva que permite el aprendizaje del idioma quechua. Además, se usó técnica de inteligencia artificial de Deep Learning acompañado con librerías como Keras y Tensorflow para ejecutar redes neuronales recurrentes. También se diseñó y se desarrolló de un chat-bot para fortalecer el aprendizaje del idioma haciendo uso del algoritmo de Distancia de Levenshtein como corrector ortográfico generando a su vez una recomendación de tipo similitud. Así mismo, para la validación del sistema se aplicó una ficha de calificación de escala de tipo Likert de 7 puntos al sistema tutor inteligente (STI) obteniendo un resultado de Totalmente de acuerdo por parte de dos profesores y el director, así llegando a la conclusión que una validación de la herramienta permite orientar de manera oportuna y correcta a los profesores de la institución, ya que con esto se puede conocer la expectativa inicial de las opiniones y las actitudes hacia al sistema tutor inteligente.

Palabras clave: Sistema tutor inteligente, Deep Learning, Redes neuronales, aprendizaje de idiomas, Quechua.

Abstract

In the Incahuasi area, Ferreñafe province, Lambayeque region, there is a communication problem between teachers and students, in the regular basic education schools (EBR), because the teachers do not have training in the local Quechua language. The consequence is that students cannot understand the content and knowledge that the teacher imparts, especially elementary-level students. That is why the following research was carried out with the objective of supporting the learning of the writing of the Quechua language to the Spanish-speaking teachers of a school in Incahuasi. The software was developed following each phase of the MeISE Methodology to specify its educational purpose, making an interactive application that allows the learning of the Quechua language. In addition, Deep Learning artificial intelligence technique was used accompanied with libraries such as Keras and Tensorflow to execute neural networks. Dialogflow was also used to carry out the development of a chat-bot to strengthen language learning. At the same time, the Levenshtein Distance algorithm was used as a spell checker. Likewise, for the validation of the system, a 7-point Likert-type scale rating sheet was applied to the intelligent tutor system (STI), obtaining a result of Totally agree by two teachers and the director, thus reaching the conclusion that a validation of the tool allows to orient in a timely and correct way to the professors of the institution, since with this it is possible to know the initial expectation of the opinions and attitudes towards the intelligent tutor system.

Keywords: Intelligent tutor system, Deep Learning, Neural networks, language learning, Quechua.

Introducción

El quechua es considerado como idioma principal del Perú desde épocas pre coloniales [1] y tiene 3 799 780 hablantes de quechua como primera lengua, que representa el 13,6% respecto de la población total, según censo en el 2017 [2].

En los últimos años nos sometemos en una discusión para pretender examinar el provenir de dicha lengua originaria, así mismo, el marco real del idioma quechua en el Perú, determinándose qué es la niñez cuando se adquiere la lengua materna y por lo tanto es la etapa en la cual se debería de fomentar la adquisición de este idioma, siendo la escuela el segundo hogar de los niños en formación, debiendo encontrarse en ella los recursos para fortalecer el aprendizaje de su lengua materna, erigiéndose como un actor principal en esta labor de reforzamiento el profesor escolar [3].

Uno de los rasgos que definen el aprendizaje de un niño es ese anhelo de disfrutar realizando algo de manera muy fácil. En cambio, los adultos conciben el aprendizaje simplemente para alcanzar un objetivo cada vez más complejo y mejorar con el tiempo según lo que está aprendiendo. Es por eso que si a un niño se le asigna un objetivo demasiado difícil provocará que se frustre y pierde la noción de lo que hará. A ningún niño le agrada realizar algo que no le salga bien, aun prometiéndole premios por su logro. Ellos disfrutan subiendo escalones más pequeños y observando lo bien que les da. En resumen, los niños priman el disfrute y los adultos, el progreso [4].

Pese a todo eso se cree necesario aprender un idioma distinto al que ya sabemos ya sea porque es un requisito laboral o para comunicarnos con otras personas que no necesariamente hablan castellano, ya que es debido a la diversidad cultural. Y en Perú podemos encontrar diferentes lenguas maternas. Cabe recalcar que no todas las lenguas originarias disfrutan de esa misma usabilidad, la mayoría están muy extendidas y habladas por las personas y otros limitados a una única región del país. El especialista de la Unesco sobre cuestiones de patrimonio cultural, Moreno, dice que [5]: "De todos los idiomas originarios solo cuatro se hablan en los Andes y las otras cuarenta y tres se hablan en la Amazonía".

Entre los primeros idiomas sobresale el castellano, que es el que se habla más en el Perú, después sigue el quechua, la segunda lengua más hablada y le sigue también el Aymara. Las que siguen, y que, por suerte, el Ashaninka es el más vital es ya que lo hablan aproximadamente 97.477 pobladores en los departamentos de Ucayali, Loreto, Apurímac, Huánuco, Ayacucho,

Junín, Pasco y Cuzco. Y a la vez existen idiomas como el sharanahua, que solo lo hablan 600 personas, según Moreno [5].

Pese a la diversidad cultural, no todos hablan el mismo idioma en el Perú y eso influye mucho en la educación de los estudiantes, en el estudio de elaborado por Antonio Girón López afirma que tiene que ver mucho el idioma, ya que es un punto clave en la preservación y propagación de los temas de la forma de vida de cada agrupación de individuos [6]. Sin embargo, la defensoría del Pueblo menciona que el Perú necesita de aproximadamente 1400 profesores que se expresen en lenguas nativas para brindar una buena enseñanza a los estudiantes y en especial a los que habitan en pueblos indígenas, pero los retos para lograr esta meta son difíciles por lo que el Ministerio de Educación está ampliando ofertas de formación en educación de maestros en todas las provincias y regiones con la población autóctono con la única finalidad de intensificar las políticas de educación intercultural bilingüe [7].

Debido a la información recopilado, en la zona de Incahuasi, provincia de Ferreñafe, región de Lambayeque, existe un problema de comunicación entre profesores y alumnos, en los colegios de educación básica regular (EBR), debido a que los profesores no tienen formación en el idioma quechua de la localidad y trae como consecuencia que los alumnos no puedan comprender los contenidos y conocimientos que el maestro imparte, sobre todo los estudiantes de nivel primaria. El gerente Regional de Educación de Lambayeque, Daniel Suarez Becerra, expresó en el año 2019 [8] que en los pueblos rurales y caseríos existen alumnos de educación primaria que son bilingües y los maestros que llegan a realizar su labor no son bilingües, y a raíz de eso se iba perdiendo esa identidad cultural de los alumnos. Esto sucede a que el contexto lingüístico de Lambayeque es que se habla el castellano (93,7% de la población) [2] ya que los profesores al querer aprender a comunicarse en quechua, presentan dificultades en la pronunciación, en la producción de textos y comunicación, puesto que su lengua materna es el castellano. Ante esta realidad, surgió la siguiente interrogante, ¿De qué manera el desarrollo de una herramienta tecnológica apoyaría en el aprendizaje de la escritura del idioma Quechua a los profesores hispanohablantes?

Por ello se propuso una alternativa de solución de software cuyo objetivo general es desarrollar un sistema tutor inteligente para apoyar en dicho aprendizaje de escritura, apoyándose con los objetivos específicos de determinar la exactitud del sistema tutor inteligente, diagnosticar el nivel de manejo de la escritura del idioma a través de un pre-test

para luego clasificarlo en aprendiz, básico o intermedio, determinar el porcentaje de errores obtenidos por los profesores durante sus pruebas para generar una recomendación y obtener una validación positiva de la usabilidad del sistema para corroborar si es factible el manejo del software.

Esta investigación se justifica porque es un antecedente elaborado dentro del contexto de la inteligencia artificial usando Deep Learning con redes neuronales recurrentes, las aplicaciones utilizadas para su desarrollo son software libre y en versión de prueba, o generando costos en su implementación. Además, existe beneficios económicos para los involucrados. Los maestros que son de instituciones publica reducirán sus posibles gastos en adquisición de material extra para fortalecer su conocimiento en el idioma y por el mismo hecho que son de contexto público, pues no son bien pagados. Esta aplicación no solo podría ayudar a esta institución de Incahuasi, quien es el objeto de estudio, sino que servirá para cualquier colegio de Incahuasi y Cañarís u otras instituciones rurales de la región o país.

Revisión de literatura

Para llevar a cabo la investigación, se tomó en cuenta ciertos antecedentes a nivel internacional, nacional y local.

Cabrera y Kotz [9] realizan una investigación que se basa en implementar técnicas de inteligencia artificial como proceso de lenguaje natural para apoyar en el tratamiento gramatical en los diferentes tipos de errores de escritura que tienen los estudiantes y la inestabilidad y poca efectividad en el tipo de retroalimentación en el tratamiento de dichos errores. El objetivo principal de este trabajo es, reparar al problema de tener mejoras en todos los procesos de aprendizaje del idioma español como lenguaje extranjero con fines profesionales y académicos, con la construcción de un tutor inteligente.

Quispe [10] realiza una investigación surge debido a la deficiencia del aprendizaje del idioma quechua en Bolivia, es por ello que se implementa un tutor inteligente con el motivo de afianzar ese aprendizaje, para ello se utilizó herramientas de desarrollo web. Su objetivo general de esta tesis es implementar un sistema tutor inteligente para que los niños de 7 a 12 años tengan acceso al aprendizaje y comunicación de la lengua. Esta investigación se llevó a cabo con el uso de agentes inteligentes y técnicas de inteligencia artificial. Ante todo, esto se pudo obtener un buen resultado.

Benavides [11], diseñó un sistema inteligente donde representó el comportamiento de un tutor que pueda enseñar a los alumnos de la I.E. Flores de Villa en Lima, para mejorar sus

competencias académicas, a través de un aprendizaje personalizado. Para este proyecto solo se visualizó el proceso de diseño del STI, y el desarrollo del proyecto, presentando su arquitectura empresarial, de información, de aplicación y tecnológica. Se desarrolló en un entorno web, donde se estableció ambientes virtuales para facilitar el aprendizaje del alumno.

Bases teóricas

Deep Learning y las Redes Neuronales

El deep learning o aprendizaje profundo es una subrama dentro de las redes neuronales. Cuando dichas redes tienen gran cantidad de capas de nodos mayor será la profundidad de la red. Permite al sistema aprender elementos cada vez más complejos. De esta manera, ayuda a que las computadoras no solo aprendan significados, sino también a que comprendan simbolismos y contextos completos y dinámicos [12].

Por otro lado, las redes neuronales artificiales (RNA) son modelos computacionales que procesan información imitando el funcionamiento de las neuronas biológicas, ya que están compuestas por “neuronas” que reciben, transmiten y envían información. Su objetivo es ayudar a que los sistemas informáticos puedan funcionar tal como un cerebro humano en cuanto a aprendizaje y pensamiento [12].

Las redes neuronales recurrentes tienen la capacidad de analizar una secuencia de datos ya que ofrece una posibilidad de analizar tipos de datos como videos, textos, conversaciones, imágenes e incluso secuencias de ADN. Estas usan el concepto de **recurrencia** (generar la salida), a la que llamaremos *activación*, no solo usa la entrada actual sino la activación generada en la iteración previa (Ver Fig.1). Es decir, las Redes Neuronales Recurrentes usan un cierto tipo de “memoria” para generar la salida deseada. [13]

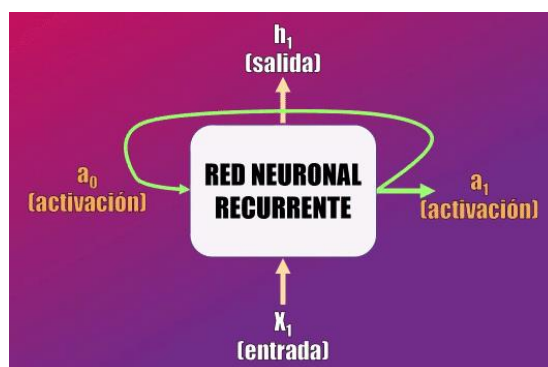


Fig. 1 Red Neuronal recurrente

Metodología de Ingeniería de Software Educativo

Conforme los estudios realizados por Álvaro Galvis [14], la Metodología de Ingeniería de Software Educativo (MeISE) es una metodología de desarrollo de software que considera una secuencia de etapa de un procedimiento metodológico, son:

Etapas de análisis: Esta es la etapa en donde se determina el entorno donde se va a crear la aplicación y asignar los requerimientos que se tendrá en cuenta en la solución interactiva, como también añadirle otras soluciones en base a las necesidades educativas [14].

Etapas de diseño: En esta etapa se construye una plantilla del tutor inteligente, basándose a los resultados recogidos en la etapa anterior, se debe considerar ser lo más claro con los datos que rodean el entorno del STI a diseñar los usuarios, área del contenido, prioridad y necesidad educativa, limitaciones y recursos físicos y lógicos [14].

Etapas de desarrollo: Se realiza la implementación de la plataforma utilizando los datos que ya se ha obtenido en las etapas ya mencionada anteriormente [14].

Etapas de prueba piloto: Esta es la fase donde se comienza el proceso de depuración del Software, partiendo de su utilización por destinatario. Es necesario elaborar y realizar ciertas validaciones que serán hechas por especialistas en prototipos entre las fases de diseño y de prueba en los módulos que se han desarrollado, con la seguridad que estén operativos [14].

Etapas de prueba de campo: La fase o etapa de prueba de campo de un SE, implica más que la utilización de los usuarios finales. Si se requiere más bien, corroborar que la aplicación que se ha desarrollado, cumpla con las especificaciones y las necesidades de los usuarios [14].

Sistema tutor inteligente

Se puede afirmar que un STI, es aquel que trabaja como un profesor particular de los alumnos, por lo consiguiente, debe tener libertad para poder responder en relación con las necesidades cognitivas de los estudiantes (Ver Fig.2). El emplear la Enseñanza Asistida por Computadora (EAC), con respecto a las características propias del estudiante, el resultado del motivo de sus errores y el procedimiento de Enseñanza – Aprendizaje Inteligente (SEAI), denominados también como Sistemas Tutores Inteligentes (STI); quienes han comprobado una

alta efectividad en distintas aplicaciones. Sin embargo, su elaboración necesita de un arduo trabajo del conocimiento y es complejo [15].

Módulo de dominio del conocimiento: Abarca la información y habilidad en la que el estudiante será enseñado. En este módulo se estructurará todo el material didáctico de estudio como vídeos, diapositivas, archivos en formatos como pdf, etc. con la finalidad de comprender mejor el concepto. Esto le permitirá al tutor conectar el material y recuperarlo de una manera más fácil; para de esta manera decidir la temática más factible y sugiriendo al estudiante, de acuerdo con lo que conoce o no, y con lo que está listo para aprender [16].

Módulo del estudiante: Se centra en establecer el nivel en el que se encuentra el estudiante de conocimiento, de acuerdo a sus estilos de aprendizaje, además de controlar su progreso respectivo.

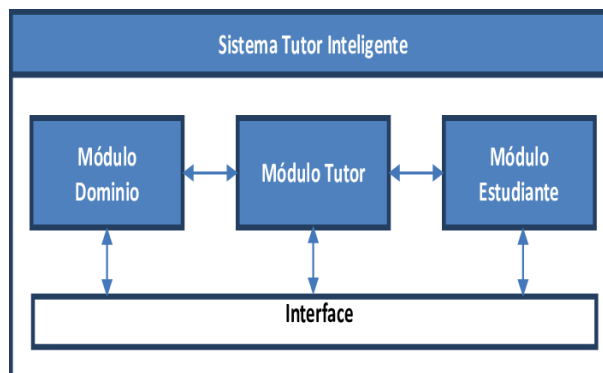
Este módulo está interconectado con el Módulo de dominio del conocimiento, tomando como referencia las peculiaridades del alumno y no solamente del conocimiento, así como la probabilidad que aprendió verdaderamente los conceptos [16].

Módulo tutor: Es aquí donde se recibe toda la información proveniente del módulo del estudiante con el módulo del dominio, con la finalidad de fijar una estrategia de tutoría personalizada.

También, es el encargado de determinar cuáles serán los métodos de evaluación y la retroalimentación del módulo del estudiante [16].

Módulo interfaz o de entornos: Es el módulo en donde se presentan los objetos visuales para la interacción con el estudiante y controlar adecuadamente sus acciones.

Las interfaces deben ser atractivas, pero sobre todo de uso fácil, de esta manera el estudiante no malgaste su tiempo y pueda emplearlo de la mejor manera posible. Centrando su atención en el entorno visual, mejorando así el proceso de enseñanza-aprendizaje del contenido [16].



.Fig. 2 Arquitectura de un sistema tutor inteligente [16]

Materiales y métodos

Para este proyecto se empleó una investigación de tipo experimental ya que se utilizó una variable independiente para manipularlo y se pretende que esta tenga un posible efecto [17]. El nivel de investigación es pre-experimental porque trabaja con un grupo y tiene un grado mínimo de control en concordancia con lo que se trabaja tan solo con un grupo que sería los profesores de la I.E de Incahuasi y las unidades de análisis no son asignadas aleatoriamente [17].

El diseño del diagrama es de la siguiente manera:

G O₁X O₂

Donde:

- G: Grupo de profesores que participarán
- O₁: Pre-prueba para analizar el aprendizaje del idioma quechua, antes de la implementación del sistema tutor inteligente.
- X: Aplicación de la solución (tutor inteligente)
- O₂: Post-prueba para analizar el aprendizaje del idioma quechua, después de la implementación del sistema tutor inteligente.

Para este proyecto se emplearon los siguientes métodos de investigación y se visualiza en la siguiente tabla:

TABLA I
MÉTODOS DE INVESTIGACIÓN

Método	Descripción
Analítico	Estudio y análisis de la problemática de la I.E con respecto al aprendizaje de la escritura del idioma quechua
Deductivo	Planificar la solución del Sistema tutor inteligente (STI) para cubrir con las necesidades de la institución.
Revisión de la literatura	Se ejecutó la búsqueda de los antecedentes y bases teóricas para la ejecución y entrenamiento del STI.
Experimento	Se entrenó el algoritmo de medición de distancias por similitud de palabras
Implementación	Se desarrolló el STI para apoyar en el aprendizaje de la escritura del idioma quechua

Para recolectar los datos se tuvo en cuenta las siguientes técnicas e instrumentos que fueron útiles y quedan registrada en la siguiente tabla:

TABLA II
TÉCNICAS E INSTRUMENTOS DE RECOLECCIÓN DE DATOS

Técnicas	Instrumentos	Elementos de la población	Propósito
Entrevista	Cuestionario (Ver Anexo N° 02)	Director de la I.E de Incahuasi	Se extrajo información sobre la problemática de la institución
Revisión documental	Matriz de Categoría (Ver Anexo N° 02)	Director de la I.E de Incahuasi	Se levantó la información sobre el problema que se presenta en la institución.
Entrevista	Cuestionario (Ver Anexo N° 03)	Docentes de la I.E de Incahuasi	Se conoció la experiencia que tiene el profesor cuando dicta sus clases en el idioma castellano.

TABLA III
MATRIZ DE CONSISTENCIA

<u>FORMULACIÓN DEL PROBLEMA</u>		<u>MÉTODOLÓGÍA DE INVESTIGACIÓN</u>	
¿De qué manera el desarrollo de una herramienta tecnológica apoyaría en el aprendizaje de la escritura del idioma Quechua a los maestros hispanohablantes?		<u>TIPO DE INVESTIGACIÓN</u>	
		Experimental	
<u>OBJETIVO GENERAL</u>	<u>MÉTODO</u>	<u>DESCRIPCIÓN</u>	
Desarrollar un sistema tutor inteligente para apoyar en el aprendizaje de la escritura del idioma quechua a profesores hispanohablantes de un colegio en Incahuasi.	Análítico	Estudio y análisis de la problemática de la I.E con respecto al aprendizaje de la escritura del idioma quechua	
	Deductivo	Planificar la solución del Sistema tutor inteligente (STI) para cubrir con las necesidades de la institución.	
	Revisión de la literatura	Se ejecutó la búsqueda de los antecedentes y bases teóricas para la ejecución y entrenamiento del STI.	
	Experimento	Se entrenó el algoritmo de medición de distancias por similitud de palabras	
<u>TÉCNICAS</u>	<u>INSTRUMENTOS</u>	<u>ELEMENTOS DE LA POBLACIÓN</u>	<u>PROPÓSITO</u>
Entrevista	Cuestionario (Ver Anexo N° 02)	Director	Se extrajo información sobre la problemática del colegio
Revisión documental	Matriz de Categoría (Ver Anexo N° 02)	Director	Se levantó la información sobre el problema que se presenta en la institución.
Entrevista	Cuestionario (Ver Anexo N° 03)	Docentes	Conocer la experiencia que tiene el profesor hispanohablante
<u>OBJETIVOS ESPECÍFICOS</u>	<u>DESCRIPCIÓN DEL LOGRO DE LOS OBJETIVOS ESPECÍFICOS</u>	<u>INDICADORES</u>	
Implementar un algoritmo para generar recomendaciones en sus previas respuestas de los profesores.	Se usa un algoritmo para mostrar las recomendaciones necesarias	Nivel de exactitud en las recomendaciones que muestra el sistema.	
Diagnosticar el nivel de manejo de la escritura del idioma a través de un pre-test para luego clasificarlo en aprendiz, básico o intermedio.	Clasificar a los profesores según puntajes y esta clasificación lo hará por un rango de calificación de aprobatoria o desaprobatoria.	Número de profesores en cada nivel con su calificación respectiva.	
Determinar el porcentaje de errores obtenidos por los profesores durante sus pruebas para generar una recomendación	Evaluar el aprendizaje del idioma a través de las recomendaciones que el sistema arroja en cada error de respuesta.	Porcentajes de Errores en sus respuestas	
Obtener una validación positiva de la usabilidad del sistema para corroborar si es factible el manejo del software	Obtener y analizar todos los resultados que arroja el software después de la interacción del profesor y estos pasarán a ser contrastado en preguntas a través de una encuesta para saber la percepción del usuario	Nivel de Percepción de la aceptación del profesor a través de una encuesta teniendo en cuenta el Modelo de aceptación de tecnología (TAM).	

Resultados y discusión

Luego de seguir con la investigación, se logró los siguientes resultados:

En base a la Metodología:

Fase 1: Análisis de necesidades educativas

En esta etapa se recolectaron las fuentes de información, para obtener las posibles causas concernientes a la problemática detectada y de esta manera analizar las alternativas de solución de acuerdo con las necesidades, donde se llegó a la conclusión de utilizar un sistema tutor inteligente. Se realizó la planeación del desarrollo del STI.

Para el modelado del tratamiento funcional y especificaciones del software, se realizaron el modelado del sistema actual del aprendizaje del estudiante. Dentro de las especificaciones, se organizó la arquitectura del sistema tutor inteligente.

Fase 2: Diseño

Para el cumplimiento de esta fase, se desarrolló el entorno del diseño del STI, se definieron los tipos del diseño en el ámbito educativo, comunicacional y computacional.

Para la parte educativa, se limita el alcance y contenido para la aplicación, se tomó en cuenta la temática de saludos, verbos y oraciones.

Se representó el diagrama del diseño educativo del STI.

Para la parte comunicacional entre el usuario y el sistema, se mostraron las especificaciones básicas y las diferentes interfaces del Sistema tutor, la funcionalidad de cada interfaz se detalla en el Anexo N° 04 que a la vez se estableció un manual de usuario.

Para el diseño computacional, se muestra el apoyo al estudiante, así como la identificación de los actores y los escenarios. Además de los diagramas de caso de uso para cada uno de los módulos.

Se representó el diagrama entidad relación del STI, con las tablas de base de datos que se emplearon.

Fase 3: Desarrollo

Con la finalidad de comprobar la eficacia del Aprendizaje Supervisado Deep Learning y Redes Neuronales Recurrente de tipo LSTM, se usó un lenguaje de programación Python en la cual se implementó todo el sistema tutor inteligente. A la vez, se usó el framework Jupyter Notebook para crear y entrenar la red neuronal recurrente donde se trabajó con 50 datos de entrada, 45 neuronas, y 2 capa ocultas de tipo dense junto con 1 salida que en este caso viene a ser la Función de activación “Softmax” que

transforma las salidas a una representación de probabilidades, y su sumatoria es 1. Se consideró la función de optimización de tipo “*rmsprop*” (tasa de aprendizaje) y de pérdida de tipo “*categorical_crossentropy*” (pérdida de entropía cruzada categórica que calcula la cantidad que un modelo debe buscar minimizar durante el entrenamiento) para ejecutar el entrenamiento con 256 iteraciones (epochs) que viene a ser el número de veces en la que se ejecuta la red neuronal recurrente.

Implementación de la Red Neuronal Recurrente Supervisado

Paso 1: Importar librerías las necesarias

Las librerías más importantes para llevar a cabo la construcción y entrenamiento de la red neuronal recurrente son (Ver Fig.3):

- **Keras:** es una API de redes neuronales que se ejecuta sobre tensorflow (framework para correr Keras); además crea prototipos fáciles y admite redes convolucionales y recurrentes. Para esta investigación se usó con el fin de crear y entrenar una red recurrente de tipo LSTM para revisar la ortografía de las palabras de manera rápida.
- **Tensorflow:** es una biblioteca de códigos abiertos para construir y entrenar redes neuronales para detectar patrones y correlaciones, análogos de aprendizaje y razonamiento usados por los humanos.

```
In [1]: import pandas as pd
import string
import re
import io
import numpy as np
from unicodedata import normalize
import keras, tensorflow
from keras.models import Model, load_model
from keras.layers import Input, LSTM, Dense

Using TensorFlow backend.
```

Fig. 3 Importación de Librerías

Paso 2: Leer la data

En este caso, se trabajó con un diccionario de palabras en el idioma quechua y se recolectó de una página web con la siguiente dirección <https://runasimi.de/runaespa.htm>. Luego, este diccionario se convirtió a un formato txt para cargarlo en el Jupyter Notebook como una dataset cuyo nombre esta como words.txt conteniendo 25 409 palabras (Ver Fig.4).

```

Reading data

In [2]: def leer_data(file):
        data = []
        with io.open(file, 'r') as file:
            for entry in file:
                entry = entry.strip()
                data.append(entry)
        return data

In [3]: data = leer_data('dataset/words.txt')

In [4]: len(data)

Out[4]: 50

```

Fig. 4 Carga la dataset words.txt

Paso 3: Limpieza y pre procesamiento de datos

Una vez cargada la dataset, se procedió con la limpieza de los datos a fin de evitar que en cada palabra vaya caracteres no permitidos.

Para esta limpieza se eliminó caracteres no imprimibles, los espacios innecesarios, se verificó que los datos tengan código ASCII, cada palabra no tenga un formato UTF8, convertir a minúsculas, que las palabras no tengan número u otros elementos, sino que estén dentro del alfabeto. Esta técnica de limpieza de datos se realizó con las siguientes sentencias de código dentro del framework de Jupyter (Revisar Fig.5).

```

limpieza de datos

In [10]: def limpiar_oraciones(sentence):
        re_print = re.compile('[^%s]' % re.escape(string.printable))
        table = str.maketrans('', '', string.punctuation)
        cleaned_sent = normalize('NFD', sentence).encode('ascii', 'ignore')
        #cleaned_sent = str.maketrans(sentence)
        cleaned_sent = cleaned_sent.decode('UTF-8')
        cleaned_sent = cleaned_sent.split()
        cleaned_sent = [word.lower() for word in cleaned_sent]
        cleaned_sent = [word.translate(table) for word in cleaned_sent]
        cleaned_sent = [re_print.sub('', w) for w in cleaned_sent]
        cleaned_sent = [word for word in cleaned_sent if word.isalpha()]
        return ' '.join(cleaned_sent)

```

Fig. 5 Limpieza datos

Paso 4: Construir conjunto de datos de entrada y salida

Como se trata de una red recurrente entonces no acepta caracteres a la entrada entonces debemos convertir cada uno de estos a una representación numérica (Ver Fig.6).

```

In [19]: def construir_data(oraciones_espanish_limpias, oraciones_quechua_limpias):
        input_dataset = []
        target_dataset = []
        input_characters = set()
        target_characters = set()

        for oracion_spanish in oraciones_espanish_limpias:
            input_datapoint = oracion_spanish
            input_dataset.append(input_datapoint)
            for char in input_datapoint:
                input_characters.add(char)

        for oracion_quechua in oraciones_quechua_limpias:
            target_datapoint = "\t" + oracion_quechua + "\n"
            #target_datapoint = oracion_quechua
            target_dataset.append(target_datapoint)
            for char in target_datapoint:
                target_characters.add(char)

        return input_dataset, target_dataset, sorted(list(input_characters)), sorted(list(target_characters))

In [20]: input_dataset, target_dataset, input_characters, target_characters = construir_data(oraciones_espanish_limpias,
        oraciones_quechua_limpias)

```

Fig. 6 Conversión de caracteres a numéricos

Se obtiene que de las 25 409 palabras que contiene la dataset, solo hay 25 caracteres en la cual forma cada uno de las palabras y estas serán usadas para generar nuestros datos de entradas para el entrenamiento de la red neuronal (Ver Fig.7).

```
In [23]: len(input_characters)
Out[23]: 24

In [24]: len(target_characters)
Out[24]: 25

In [25]: print(input_characters)
[' ', 'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'y', 'z']

In [26]: print(target_characters)
['\t', '\n', 'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'w', 'y']
```

Fig. 7 Caracteres de entrada y salida de la red neuronal recurrente

Paso 5: Definición de metadatos

Se empieza a construir los elementos para ver cómo van a entrar al codificador y decodificador porque es necesario para la entrada de cadenas en números.

Se definió los números de puntos de datos quienes vienen a ser los 24 elementos dentro de los caracteres de datos de entrada. Los números de tokens de entrada únicos que son los 25 elementos dentro de los caracteres de datos de salida. Y buscan la longitud máxima para secuencia de entrada dentro de las 25 409 palabras, que en este caso serían 14 para entradas y salidas (Ver Fig.8).

Estos datos obtenidos se usarán para definir bien nuestra red neuronal recurrente.

```
In [27]: def construir_metadata(input_dataset, target_dataset, input_characters, target_characters):
num_encoder_tokens = len(input_characters)
num_decoder_tokens = len(target_characters)
max_encoder_seq_length = max([len(data_point) for data_point in input_dataset])
max_decoder_seq_length = max([len(data_point) for data_point in target_dataset])

print('Número de puntos de datos:', len(input_dataset))
print('Número de tokens de entrada únicos:', num_encoder_tokens)
print('Número de tokens de salida únicos:', num_decoder_tokens)
print('Longitud máxima de secuencia para entradas:', max_encoder_seq_length)
print('Longitud máxima de secuencia para salidas:', max_decoder_seq_length)

return num_encoder_tokens, num_decoder_tokens, max_encoder_seq_length, max_decoder_seq_length

In [28]: num_encoder_tokens, num_decoder_tokens, max_encoder_seq_length, max_decoder_seq_length = construir_metadata(input_dataset,
target_dataset,
input_characters,
target_characters)

Número de puntos de datos: 50
Número de tokens de entrada únicos: 24
Número de tokens de salida únicos: 25
Longitud máxima de secuencia para entradas: 14
Longitud máxima de secuencia para salidas: 14
```

Fig. 8 Definir los metadatos

Paso 6: Mapeos de datos

Se crean diccionarios para los datos de caracteres de entrada y salida para generarle un ID o valoración de puntuación haciendo que estos se conviertan en un valor numérico y puedan entrar en manera secuencial en la red recurrente (Ver Fig. 9).

```
In [29]: def build_indices(input_characters, target_characters):
input_char_to_idx = {}
input_idx_to_char = {}
target_char_to_idx = {}
target_idx_to_char = {}

for i, char in enumerate(input_characters):
input_char_to_idx[char] = i
input_idx_to_char[i] = char

for i, char in enumerate(target_characters):
target_char_to_idx[char] = i
target_idx_to_char[i] = char

return input_char_to_idx, input_idx_to_char, target_char_to_idx, target_idx_to_char

input_char_to_idx, input_idx_to_char, target_char_to_idx, target_idx_to_char = build_indices(input_characters,
target_characters)
```

Fig. 9 Mapeos de caracteres a indexar

Se visualiza que tanto a los datos de entradas como de salidas en definición de caracteres se le dio una valorización para que la red recurrente pueda codificar y decodificar las palabras (Ver Fig.10).

In [30]:	input_char_to_idx	In [31]:	target_char_to_idx
Out[30]:	{ ' ': 0, 'a': 1, 'b': 2, 'c': 3, 'd': 4, 'e': 5, 'f': 6, 'g': 7, 'h': 8, 'i': 9, 'j': 10, 'l': 11, 'm': 12, 'n': 13, 'o': 14, 'p': 15, 'q': 16, 'r': 17, 's': 18, 't': 19, 'u': 20, 'v': 21, 'y': 22, 'z': 23}	Out[31]:	{ '\t': 0, '\n': 1, 'a': 2, 'b': 3, 'c': 4, 'd': 5, 'e': 6, 'f': 7, 'g': 8, 'h': 9, 'i': 10, 'j': 11, 'k': 12, 'l': 13, 'm': 14, 'n': 15, 'o': 16, 'p': 17, 'q': 18, 'r': 19, 's': 20, 't': 21, 'u': 22, 'w': 23, 'y': 24}

Fig. 10 Valoración de los caracteres de datos de entrada y salida

Paso 7: Definición de codificador

Va a recibir el num_encoder_tokens que son 50 datos que van a entrar a la red recurrente que vendría ser la entrada.

Luego se crea la red LSTM cuyo número de neuronas es de 45 (Ver Fig.11)

```
In [42]: encoder_inputs = Input(shape=(None, num_encoder_tokens))
encoder = LSTM(num_neuronas, return_state=True)
encoder_outputs, state_h, state_c = encoder(encoder_inputs)
encoder_states = [state_h, state_c]
```

Fig. 11 Codificador de la red recurrente LSTM

Paso 8: Definición del decodificador

Se crea la capa oculta de tipo dense junto con las salidas que en este caso viene a ser la Función de activación Softmax que transforma las salidas a una representación de probabilidades, y su sumatoria es 1 (Ver Fig.12).

```
In [43]: decoder_inputs = Input(shape=(None, num_decoder_tokens))
decoder_lstm = LSTM(num_neuronas, return_sequences=True, return_state=True)
decoder_outputs, _, _ = decoder_lstm(decoder_inputs, initial_state=encoder_states)
decoder_dense = Dense(num_decoder_tokens, activation='softmax')
decoder_outputs = decoder_dense(decoder_outputs)
```

Fig. 12 Decodificador de la red recurrente LSTM

Paso 9: Construir el modelo

Llamamos la función Model y le damos los parámetros de entrada del codificador y el decodificador y los parámetros de salida quien viene acompañado con la función de activación softmax.

Luego llamamos la función compile que va a guardar la función de optimización de tipo “rmsprop” (tasa de aprendizaje) y de pérdida de tipo “categorical_crossentropy” (pérdida de entropía cruzada categórica que calcula la cantidad que un modelo debe buscar minimizar durante el entrenamiento) que tendrá el algoritmo.

Finalmente, llamamos a la función summary para que nos muestre el resumen de todo el modelo (Ver Fig.13)

```
In [44]: model = Model(inputs=[encoder_inputs, decoder_inputs],
                      outputs=decoder_outputs)

In [45]: model.compile(optimizer='rmsprop', loss='categorical_crossentropy')
model.summary()

Model: "model_1"
-----
Layer (type)                Output Shape              Param #   Connected to
-----
input_1 (InputLayer)        (None, None, 24)         0
-----
input_2 (InputLayer)        (None, None, 25)         0
-----
lstm_1 (LSTM)                [(None, 256), (None, 287744)  input_1[0][0]
-----
lstm_2 (LSTM)                [(None, None, 256), 288768  input_2[0][0]
                                     lstm_1[0][1]
                                     lstm_1[0][2]
-----
dense_1 (Dense)              (None, None, 25)         6425     lstm_2[0][0]
-----
Total params: 582,937
Trainable params: 582,937
Non-trainable params: 0
```

Fig. 13 Construcción de modelo

Paso 10: Entrenar al modelo

Para entrenar el modelo se usa la función Fit() que toma como parámetros al encoder y decoder input data, decoder target, el batch size que es 256 al igual que la época o número de iteración que se ejecuta la red neuronal. También, se coloca una cierta proporción del conjunto de entrenamiento como conjunto de validación que es 0.1 (Ver Fig.14).

```
In [46]: model.fit([encoder_input_data, decoder_input_data], decoder_target_data,
                batch_size=batch_size,
                epochs=epochs,
                validation_split=0.1)

45/45 [=====] - 1s 32ms/step - loss: 1.8332 - val_loss: 1.7306
Epoch 2/256
45/45 [=====] - 0s 2ms/step - loss: 1.8054 - val_loss: 1.6927
Epoch 3/256
45/45 [=====] - 0s 2ms/step - loss: 1.7525 - val_loss: 1.6874
Epoch 4/256
45/45 [=====] - 0s 2ms/step - loss: 1.6672 - val_loss: 1.6931
Epoch 5/256
45/45 [=====] - 0s 3ms/step - loss: 1.7344 - val_loss: 1.6677
Epoch 6/256
45/45 [=====] - 0s 3ms/step - loss: 1.6934 - val_loss: 1.6261
Epoch 7/256
45/45 [=====] - 0s 2ms/step - loss: 1.6221 - val_loss: 1.5948
Epoch 8/256
45/45 [=====] - 0s 2ms/step - loss: 1.5429 - val_loss: 1.6186
Epoch 9/256
45/45 [=====] - 0s 2ms/step - loss: 1.5040 - val_loss: 1.6513
Epoch 10/256
45/45 [=====] - 0s 2ms/step - loss: 1.5043 - val_loss: 1.6376
Epoch 11/256
```

Fig. 14 Entrenamiento del modelo

Paso 11: Guardar el entrenamiento del modelo

Lo que se obtiene del entrenamiento de la red recurrente, el archivo H5 neural_machine-translation.h5 se coloca en la vista del STI; es decir en el main para así tener una eficacia de las respuestas de los profesores.

Paso 12: Comprobamos que la red haya aprendido

Llamamos la función predict () que permite predecir las etiquetas de los valores de los datos sobre la base del modelo entrenado (Ver Fig.15)

```
In [49]: def decodificar_oracion(input_seq):
        states_value = encoder_model.predict(input_seq)
        #print ("states_value: ", states_value)
        target_seq = np.zeros((1, 1, num_decoder_tokens))
        target_seq[0, 0, target_char_to_idx['\t']] = 1.
        stop_condition = False
        decoded_sentence = ''
        while not stop_condition:
            output_tokens, h, c = decoder_model.predict([target_seq] + states_value)
            #print ("output_tokens: ", output_tokens.round(2))
            sampled_token_index = np.argmax(output_tokens[0, -1, :])
            sampled_char = target_idx_to_char[sampled_token_index]
            decoded_sentence += sampled_char

            if (sampled_char == '\n' or len(decoded_sentence) > max_decoder_seq_length):
                stop_condition = True

            target_seq = np.zeros((1, 1, num_decoder_tokens))
            target_seq[0, 0, sampled_token_index] = 1.

            states_value = [h, c]

        return decoded_sentence
```

Fig. 15 Pruebas del entrenamiento

Se observa que el modelo pudo predecir de manera exitosa, aunque hubo algunos errores, pero es parte del entrenamiento. Por ejemplo, para el ítem $i=0$, la oración de entrada es “deliciosa” y la frase decodificada debería ser “mishki” (Ver Fig.16).

```
In [51]: for i in range(len(data)):
         print("i= ",i)
         decode(i)

i= 0
Oración de entrada: deliciosa
Frase decodificada: sikwiq

i= 1
Oración de entrada: comida
Frase decodificada: mikunata

i= 2
Oración de entrada: como
Frase decodificada: mikuni

i= 3
Oración de entrada: chompa
Frase decodificada: chumpanqa

i= 4
Oración de entrada: color amarillo
Frase decodificada: qarwa
```

Fig. 16. Resultados de las pruebas

Creación del chat-bot

También, se creó el chat-bot como un sistema de recomendación que ayudará al usuario a escribir correctamente las palabras y cada error que cometa el chat-bot le brindará esa sugerencia de traducción. Se definió primero bajo que enfoque de tipo de sistema de recomendación se usará y este fue el filtrado basado en contenido. Se hizo uso del algoritmo de distancia de Levenshtein para a partir de ahí jalar todas las recomendaciones en el chat bot.

Para validar el modelo de la construcción y ejecución de la red neuronal recurrente de tipo LSTM se hizo la validación por juicio de expertos. Fueron dos (2) expertos que respondieron a esta ficha de evaluación quienes son conocedores de la rama de inteligencia artificial. La ficha contiene 5 preguntas (ver Anexo 6) y los resultados obtenido fueron conveniente con un (90) % de aceptación calificándose en un índice alto de PROBABLE, entonces se puede decir que con la construcción de esta red neuronal permite aprender las palabras del español al quechua.

Fase 4: Pruebas al sistema

Para la fase de pruebas al sistema, se validó de acuerdo con las pruebas de testeo de software; la prueba de caja negra, revisando los datos. La prueba de caja blanca analizó el Inicio de Sesión al sistema, de acuerdo con el código fuente del STI.

En base a los objetivos:

Implementación de un algoritmo para generar recomendaciones en sus previas respuestas de los profesores.

Se usó un tipo de sistema de recomendación enfocado a filtrado basado en contenido donde encuentra la similitud entre las diferentes palabras, en este caso será calculado por la similitud que tenga el usuario al ingresar la palabra correcta junto con la similitud de la dataset.

Para mostrar la similitud y la recomendación, se usó del algoritmo de La Distancia de Levenshtein (número mínimo de operaciones para transformar una cadena de caracteres en otra). Quien se implementó en un chat-bot para que se visualice mejor las recomendaciones que ofrece el sistema.

Todo este proceso se puede visualizar en el apartado de IV Resultados y discusión en la sesión 4.1.3. Fase 3: Desarrollo.

A continuación, se presenta el código principal (Revisar Fig.17) donde se visualiza como al ser entrenado el algoritmo muestra las recomendaciones de las posibles respuestas según similitud de palabras que haya ingresado el usuario.

```
In [62]: #MEDICION DE DISTANCIAS POR SIMILITUD DE PALABRAS
palabra_de_checkeo = "mikunit"
lista_de_palabras = word_set
print (lista_de_palabras)
inicio_algoritmo = time.time()
resultados = revision_ortografia(palabra_de_checkeo, lista_de_palabras)
fin_algoritmo = time.time()
tiempo_total = fin_algoritmo - inicio_algoritmo

print ("\nResultados del algoritmo")
for count, word in enumerate(resultados, start=1):
    print (count, word)
print ("\nLa palabra de prueba es: %s." % palabra_de_checkeo)
print ("Tiempo de calculo: ", tiempo_total, "seconds\n")

{'wasita', 'sikwi', 'aparisha', 'mushuc', 'kuyaylla', 'sikwina', 'sikya', 'chupikawan', 'yanapamay', 'abansay', 'chumpanqa', 's
hkrata', 'jatun', 'achachaw', 'mishki', 'yana', 'yuyaychakuy', 'bandira', 'maypitaq', 'kuliyyuga', 'silgis', 'yurichiq', 'alli
chaqe', 'imaraq', 'wawakuna', 'yuraq', 'syiluqa', 'qhilla', 'qarwa', 'aqchanqa', 'silbay', 'tuldyawlla', 'kulurninqami', 'yur
i', 'abugadu', 'yuyanacha', 'sikwiq', 'abunu', 'silakuy', 'mikuni', 'perupa', 'profisur', 'achacha', 'mikunata', 'sila', 'sikwi
y', 'lachiwana', 'noqa', 'sili'}

Resultados del algoritmo
1 mikunata
2 mikuni

La palabra de prueba es: mikunit.
Tiempo de calculo: 0.19997406085859375 seconds
```

Fig. 17. Código de recomendaciones según similitud de palabras

Diagnóstico del nivel de manejo de la escritura del idioma a través de un pre test para luego clasificarlo en aprendiz, básico o intermedio.

Para saber el nivel de la escritura del idioma para cada profesor se realizó un pre test con 20 preguntas. Este proceso solo son los conocimientos con la que inicia cada usuario para separarlos por niveles y que el tutor inteligente pueda recomendar en que temas debería reforzar.

La población estuvo compuesta por 20 profesores las cuales se dividieron en dos grupos de 10 profesores cada uno y se obtuvieron los siguientes resultados que se mostrará en la siguiente tabla:

TABLA IV
RESULTADOS DEL PRE TEST APLICADA EN EL STI

N°	G1		G2	
	Nivel	Puntuación	Nivel	Puntuación
1	Aprendiz	30	Aprendiz	34
2	Aprendiz	50	Básico	56
3	Aprendiz	34	Básico	57
4	Aprendiz	48	Aprendiz	45
5	Aprendiz	20	Aprendiz	23
6	Aprendiz	34	Aprendiz	45
7	Aprendiz	49	Aprendiz	34
8	Aprendiz	45	Básico	51
9	Aprendiz	39	Aprendiz	43
10	Aprendiz	40	Básico	51
Promedio	Aprendiz	38.9	Aprendiz	43.9

Para el G1, el promedio general de puntuación es 38.9 indicando que los 10 profesores ingresan con un nivel aprendiz. Mientras tanto para el G2, el promedio es de 43.9 indicando que también están en un nivel aprendiz.

Para el grupo 1 se detectó que los 10 profesores inician con un nivel APRENDIZ del manejo de escritura del idioma (Ver Fig.18), que equivale al 100% de profesores, haciendo a su vez que el sistema tutor inteligente le proponga temas de sugerencias para fortalecer el aprendizaje.

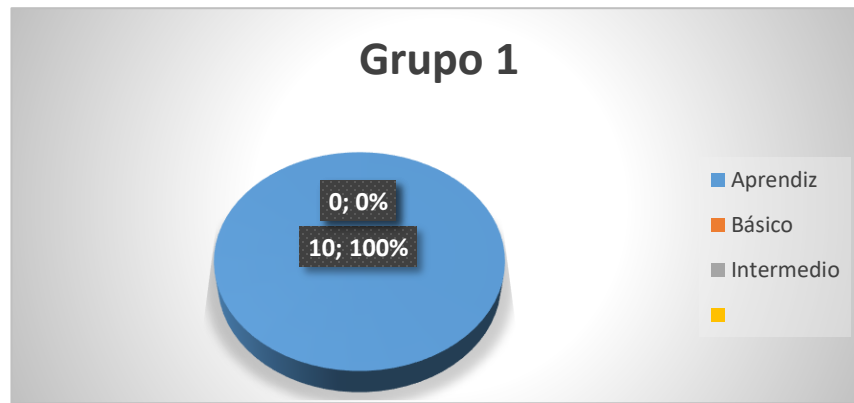


Fig. 18 Nivel del Grupo 1

En cambio, para el grupo 2 se detectó que de los 10 profesores; 6 inician con un nivel APRENDIZ (Ver Fig.19) que equivale al 60% y 4 inician con el nivel BÁSICO que equivale el 40%, haciendo a su vez que el sistema tutor inteligente le proponga temas de sugerencias para fortalecer el aprendizaje según el nivel.

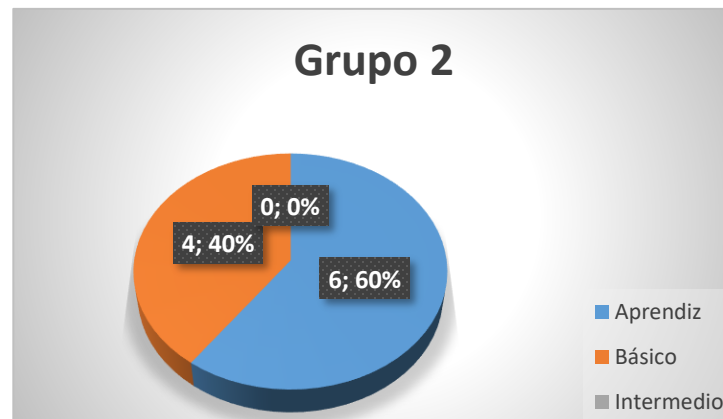


Fig. 19 Nivel del grupo 2

Determinación del porcentaje de errores obtenidos por los profesores durante sus pruebas para generar una recomendación

Para determinar este porcentaje de errores se hizo bajo dos visiones, la primera fue mediante el chat-bot (Ver Fig.20) y el segundo por un post test (Ver Fig. 21) y se hará el análisis uno por uno ya que esto funciona en diferentes perspectivas.

Para el chat-bot no se verán físicamente los porcentajes, pero si se mostrará como el tutor identifica el error de la escritura mediante el algoritmo de medición de distancias por similitud de palabras. El usuario ingresará su traducción y el tutor junto con el

algoritmo medirá el porcentaje de error de manera interna y le mostrará una recomendación.



Fig. 20 Recomendación que brinda el chat-bot frente a un error del usuario

```
127.0.0.1 - - [04/Oct/2021 20:19:23] "GET /get?
msg=mikunata&PalabraTraducir=comida HTTP/1.1" 200 -
Palabras similares encontradas: 1
1 mikunata
encontradooo
pivote: 2
pos encontrado: 1
```

Fig. 21 Medición del error de manera interna

Para el pos-test si se presentará el porcentaje al finalizar las 20 preguntas asignadas de manera aleatoria. Para evidenciar este objetivo solo se tomará en cuenta el primero puntaje que se obtenga porque el software presenta la lista que puntaje que se obtiene en cada intento del post test. Cabe resaltar que mientras más intento se tenga en el post-test menos error de porcentaje tendrá debido a que irá aprendiendo. Algunos resultados del primer intento se muestran en la siguiente tabla:

TABLA V
RESULTADOS DEL POST TEST PARA VER EL PORCENTAJE DE ERROR

N°	G1		G2	
	Puntuación	Porcentaje de error	Puntuación	Porcentaje de error
1	50	50%	48	52%
2	65	35%	66	34%
3	58	42%	68	32%
4	61	39%	50	50%
5	30	70%	45	55%
6	59	41%	50	50%
7	60	40%	49	51%
8	59	41%	65	35%
9	63	37%	59	41%
10	55	45%	61	39%
Promedio	56	44%	56.1	43.9%

Cada vez que el usuario ingrese su posible respuesta, el tutor inteligente le dará la sugerencia de traducción (Revisar Fig. 22).



Fig. 22 Interfaz de sugerencia de traducción

Obtención de una validación positiva de la usabilidad del sistema para corroborar si es factible el manejo del software.

Se aplicó una ficha de encuesta teniendo en cuenta el Modelo de aceptación de Tecnología (TAM). Esta ficha fue aplicado a dos profesores y al director para validar la facilidad de uso y la utilidad percibida.

Los resultados de la ficha permiten que el sistema tutor inteligente tenga una retroalimentación posterior, por lo tanto, se obtuvo una valoración del 90%, la ficha se muestra en el Anexo N° 5.

Los resultados de esta investigación fueron positivos ya que se mostró un eficiente aprendizaje de la escritura del idioma por parte de los profesores y corroborado con el director. Además, se obtuvo una validación buena sobre el uso del tutor inteligente por parte de dos profesores y el director; haciendo que este trabajo tenga una aceptación sobre su utilidad y usabilidad regido al Modelo TAM. A diferencia de [9] que realizó una investigación que se basa en implementar técnicas de inteligencia artificial como proceso de lenguaje natural para apoyar en el tratamiento gramatical en los diferentes tipos de errores de escritura que tienen los estudiantes haciendo que estos mejoren en todos los procesos de aprendizaje del idioma español como lenguaje extranjero con fines profesionales y académicos, con la construcción de un tutor inteligente. En cambio, en la presente investigación se implementó un chat-bot donde además de dar tratamiento a los errores en sus pruebas también dan recomendaciones para mejorar su aprendizaje.

Así mismo, los siguientes autores [10] realizaron una investigación que surge debido a la deficiencia del aprendizaje del idioma quechua en Bolivia e implementaron un tutor inteligente utilizando herramientas de desarrollo web. Al igual que esta investigación, también se usó la metodología MeISE, pero lo que diferencia es que los autores obviaron una fase de dicha metodología haciendo que este software tenga algunas complicaciones futuras. Lo que también se añadió a esta investigación es el grado de dificultad ya que los autores solo tomaron en cuenta un nivel básico y el grado de dificultad que se logró en mi investigación es de aprendiz, básico e intermedio.

En [18] desarrolla el tutor inteligente teniendo en cuenta la misma metodología, MeISE para hacer que las interfaces sean más interactivas con el estudiante. Lo que diferencia de esta investigación es que el autor obvia la Fase de Pruebas al sistema haciendo que esta sea un poco favorable al momento de la ejecución del software. El autor no utiliza técnica de inteligencia artificial para el desarrollo de este tutor inteligente, en cambio en mi investigación se usó una red neuronal recurrente con Deep Learning junto con un algoritmo de distancia de Levenshtein para construir un chat bot y que este ayude en el aprendizaje del idioma quechua.

De igual manera en [19] se hizo una revisión de todas las plataformas para elegir un entorno de E-Learning que permitiera fortalecer el aprendizaje de la lengua y utilizó el método mixto a un nivel descriptivo/analítico. Analizó que existen más de 500 soluciones LMS (Learning Management System) eligiendo la plataforma Chamilo que está orientado al usuario donde realiza un pago en línea para acceder a los módulos. Por ende, no propuso un prototipado funcional propio para poner en práctica sus postulados, lo que hace diferente a esta investigación es que si se muestra el software construido y las fases por la que paso para su ejecución, también se hace saber que este tutor inteligente es totalmente gratis y se puede lograr el aprendizaje previo a la escritura del idioma.

Para [11] diseñó un sistema inteligente donde representara el comportamiento de un tutor que pueda enseñar a los alumnos de la I.E. Flores de Villa en Lima, para mejorar sus competencias académicas, a través de un aprendizaje personalizado, por ende solo se visualizó la planificación que debería seguirse en el desarrollo de un Software Inteligente presentando su arquitectura empresarial, de información, de aplicación y tecnológica, más no realiza la construcción del sistema para validar que realmente esta plataforma va a contribuir con sus propuestas. Lo que hace diferente a esta investigación es que, si se llevó a cabo el diseño, la implementación y la ejecución del software mediante una metodología y con el uso de técnicas de inteligencia artificial, además se hizo un análisis mediante un modelo TAM para percibir si este sistema contribuye en el aprendizaje de la escritura del idioma Quechua.

Por último, para [20], los autores propusieron implementar un software para mejorar sus competencias comunicativas del idioma quechua y para eso utilizaron la metodología SCRUM ya que no es un sistema inteligente, en cambio en esta investigación se usó otra metodología que esta orienta para agentes inteligentes, el caso de MeISE. En ambas investigaciones se logra la aceptación del software, afirmando que si se logró ayudar en las competencias del aprendizaje del idioma quechua cumpliendo así con cada objetivo planteado.

Conclusiones

Se logró implementar el sistema tutor inteligente para apoyar en el aprendizaje de la escritura del idioma quechua, luego de una previa revisión teórica como la Metodología idónea para seguir su implementación, el nivel mínimo requerido por el Ministerio de educación para que el sistema tenga un grado de dificultad acorde con lo establecido, el algoritmo de inteligencia artificial. Toda esta información fue de suma importancia para tener en cuenta que información iba a contar el tutor para apoyar al profesor hispanohablante.

Se logró tener un diagnóstico del nivel de manejo de la escritura del idioma mediante un pre test donde luego se clasificó a cada profesor según lo correspondido como nivel aprendiz, básico o intermedio.

En este trabajo, también se determinó los porcentajes de errores obtenido durante sus pruebas que es el caso del post – test brindándole a la vez recomendaciones sobre sus posibles errores.

Se obtuvo una validación positiva de la usabilidad del sistema, corroborando la factibilidad del software. Fue validado por la misma institución educativa que comprende el director y los profesores.

Recomendaciones

Se recomienda esta investigación para trabajos futuros donde el sistema tutor inteligente también sea adaptado para reforzar otros idiomas.

Se recomienda agregar un módulo de pronunciación para que el tutor corrobore si el usuario está pronunciando correctamente o no las palabras en quechua.

Se recomienda, también, agregar un módulo para escuchar la pronunciación de las palabras del idioma y así apoyar aún más en el aprendizaje del quechua.

Se recomienda al Ministerio de Educación del Perú a incorporar este tutor inteligente dentro de la maya curricular de las instituciones educativas donde se presente este idioma para seguir reforzando tanto a alumnos y profesores para apoyar en sus clases respectivas.

Referencias

- [1] L. Andrade Ciudad, «Diez noticias sobre el quechua en el último censo peruano,» *Letras*, vol. 90, n° 132, pp. 41-70, 2019.
- [2] INEI, «INEI,» [En línea]. Available: <https://www.inei.gov.pe/estadisticas/indice-tematico/trade/>. [Último acceso: 26 Mayo 2020].
- [3] M. D. C. Ruíz Calatrava, «El aprendizaje de una lengua extranjera a distintas,» *Espiral. Cuaderno del profesorado*, vol. 2, n° 3, pp. 98-103, 2009.
- [4] T. M. Gonzales Portillo, *La Gran Pausa: La guía del músico profesional; Cuerpo, mente, metodologías y pedagogía para avanzar en tu carrera. (Spanish Edition)*, Edición Kindle, 2016.
- [5] L. Blasco, «¿Cuántas lenguas originarias existen en Perú y cuáles de éstas se,» *BBC News Mundo*, 6 Noviembre 2018. [En línea]. Available: <https://www.bbc.com/mundo/noticias-46074381>.. [Último acceso: 7 Julio 2020].
- [6] A. Girón López , ««La lengua como instrumento de aprendizaje escolar,»» [En línea]. Available: https://mandripm.files.wordpress.com/2011/05/1er_e_0614.pdf. [Último acceso: 7 Julio 2020].
- [7] A. Noticias, «Defensoría: Perú necesita unos 14 mil profesores que hablen lenguas nativas,» *America Noticias*, 22 Febrero 2018. [En línea]. Available: <https://www.americatv.com.pe/noticias/actualidad/defensoria-peru-requiere-al-menos-14-mil-profesores-que-hablen-lenguas-nativas-n311607>. [Último acceso: 25 Mayo 2020].
- [8] L. Industria, «298 colegios implementan una educación intercultural bilingüe,» *La Industria*, 9 Noviembre 2019. [En línea]. Available: <https://www.laindustriadechiclayo.pe/noticia/1573077038-298-colegios-implementan-una-educacion-intercultural-bilingue>. [Último acceso: 5 Julio 2020].
- [9] A. Ferreira Cabrera y G. Kotz Grabole, «Un sistema tutorial inteligente para el tratamiento de errores gramaticales del español como lengua extranjera con fines académicos,» Chile, 2016.
- [10] V. P. Quispe Castañeta., «Tutor inteligente web para la enseñanza de la lengua quechua en niños de 7 a 12 años de edad (nivel básico),» La Paz, 2017.

- [11] A. Benavides Cabrera, «Diseño de un software inteligente para mejorar el aprendizaje en estudiantes de la institución educativa Flores de Vila, Lima-2019,» Lima, 2019.
- [12] E. A. Group, «Redes Neuronales Artificiales y Deep Learning, explicado para dummies,» [En línea]. Available: <https://blog.enzymeadvisinggroup.com/redes-neuronales-artificiales-y-deep-learning>. [Último acceso: 23 Julio 2021].
- [13] CodificandoBits, «Introducción a las Redes Neuronales Recurrentes,» 8 Junio 2019. [En línea]. Available: <https://www.codificandobits.com/blog/introduccion-redes-neuronales-recurrentes/#reconocimiento-de-escritura>. [Último acceso: 3 Octubre 2021].
- [14] A. Gavlis Panqueva , Ingeniería de software educativo, Bogotá: Ediciones Uniandes, 1997.
- [15] L. Ceccaroni, «Inteligencia Artificial,» 2017. [En línea]. Available: <https://www.cs.upc.edu/~luigi/II/IA-2007-fall/ia.html>. [Último acceso: 19 Octubre 2020].
- [16] G. D. L. C. León Rodríguez, «La inteligencia artificial en la educación superior. Oportunidades y amenazas,» *IRJ*, vol. 2, n° 8, pp. 412-422, 2017.
- [17] S. Palella Stracuzzi y F. Martins Pestana, "Metodología de la investigación cuantitativa", Caracas: FEDUPEL, 2012.
- [18] M. Mamani Ulo, «Tutor inteligente para la enseñanza de lectura y escritura del idioma aymara,» La Paz, 2017.
- [19] M. A. Herrera Bendezú, «Elección de una plataforma E-learning idónea para el aprendizaje de la lengua quechua,» Ayacucho, 2018.
- [20] L. Á. Carrera Sicha y D. E. Pizarro Chavez, «Apliación movil para mejorar el aprendizaje de los estudiantes en la lengua quechua de la Academia Yachay Wasi,» Universidad Autónomo del Perú, Lima, 2018.
- [21] A. Cortez Vásquez, H. Vega Huerta y . J. Pariona Quispe, «Procesamiento de lenguaje natural,» *Revista de Ingeniería de Sistemas e Informática*, vol. 62, n° 2, pp. 45-54, 2009.

Anexos

**ANEXO N° 01. CONSTANCIA DE APROBACIÓN DEL PRODUCTO ACREDITABLE
DE LA ENTIDAD DONDE SE EJECUTÓ LA TESIS**



PERÚ

Ministerio
de EducaciónGerencia regional de
Educación de Lambayeque

I.E. "10082" "Sagrado Corazón de Jesús"

Primaria – Secundaria

Incahuasi – Uryupampa

RD N° 3249-2004



EL DIRECTOR DEL COLEGIO N° "10082" "SAGRADO CORAZÓN DE JESÚS" DEL
DISTRITO DE INCAHUASI, PROVINCIA DE FERREÑAFE, QUE SUSCRIBE,
OTORGA LO SIGUIENTE:

**CONSTANCIA DE APROBACIÓN
DEL PRODUCTO ACREDITABLE**

A: Silvia Mariela Casiano Inga

Por medio de la presente certifico la aprobación del tutor inteligente para apoyar en el aprendizaje de la escritura del idioma quechua como producto acreditable de su trabajo de investigación de fin grado, cumpliendo los requisitos establecidos y objetivos planteados por este.

Se expide la presente a pedido escrito de la interesada para los fines pertinentes.

Ferrenafe, Julio 2021




 Prof. Carlos del Carmen Benito Carlos
 DIRECTOR

ANEXO N° 02. MATRIZ DE CATEGORIA Y GUÍA DE ENTREVISTA AL DIRECTOR

MATRIZ DE CATEGORÍA

CATEGORÍA	DESCRIPCIÓN
Perspectiva teórica	Nuevas teorías de las formas de enseñanza de un idioma o idiomas. En este caso será el quechua.
Dimensión explicativa	Se aplicará herramientas tecnológicas para apoyar en las nuevas formas de enseñanza del idioma quechua.
Situación problemática	Se observa, por parte de los profesores, que desconocen la expresión y escritura del idioma y no tienen una ayuda para lograr el aprendizaje del idioma.
Formulación de alternativas de solución	Implementar, con la ayuda de las nuevas alternativas tecnológicas, un sistema tutor inteligente para apoyar en el aprendizaje del idioma quechua.

GUÍA DE ENREVISTA AL DIRECTOR DE LA INSTITUCIÓN EDUCATIVA DE INCAHUASI

ENTREVISTA N°01

Nombre: Prof. Oscar Del Carmen Bernilla Carlos **Fecha:** 18-08-2020

Institución Educativa: I.E. "N° 10082" "Sagrado Corazón de Jesús" **Hora:** 10:00 a.m.



Lugar: Incahuasi - Uryupampa

Objetivo:

Se realizó esta entrevista con la finalidad de conocer la realidad problemática que ocurre dentro de la institución educativa para así darle una solución tecnológica que brinde apoyo.

Preguntas:

1. Según su apreciación, ¿Cuál es la deficiencia con la que actualmente cuenta la institución?
2. ¿Cómo considera usted la forma en que actualmente se maneja la información?
3. ¿Qué sugiere usted para dar solución?
4. ¿Estaría dispuesto que una estudiante de ingeniería de sistemas le dé solución al problema que está atravesando?
5. ¿Estaría dispuesto a brindar la información necesaria para realizar el proyecto de investigación que me corresponde?

Prof. Oscar del Carmen Bernilla Carlos
DIRECTOR

ANEXO N° 03. ENTREVISTA A UN PROFESOR HISPANOHABLANTE DE LA INSTITUCIÓN EDUCATIVA

ENTREVISTA N°01

Nombre: Prof. María Consuelo Piscoya Cavero

Fecha: 21-08-2020

Institución Educativa: I.E. "N° 10082" "Sagrado Corazón de Jesús" **Hora:** 10:00 a.m.

Lugar: Incahuasi – Uryupampa

Objetivo:

Se realizó esta entrevista con la finalidad de conocer la experiencia que tiene el profesor cuando dicta sus clases en el idioma castellano y cómo es la comunicación con sus alumnos. Y sobre si necesita de alguna ayuda tecnológica.

Preguntas:

1. ¿Cuál es el cargo o qué curso dicta en el colegio?

Respuesta: Dicta el curso de Ciencias Sociales en el grado de 1° a 5° de secundaria.

2. ¿Cuál es la didáctica para llevar a cabo sus clases con sus alumnos?

Respuesta: Como toda clase, llega al aula y dicta la temática. Hace uso de papelote y gráficos de apoyo

3. ¿Las clases la expresa en el idioma quechua? ¿Cómo hace para que sus estudiantes entiendan sus clases en el idioma que ellos no saben?

Respuesta: Las clases las dicta en el idioma que ella sabe. A veces le pregunta a sus alumnos si la entendieron y ella pide que le ayude a traducir algunas palabras que los alumnos no entienden el significado de la palabra en español.

4. ¿Ha buscado algún material, libro, diccionario o cualquier material para guiarse?

Respuesta: Expresa que no existe mucho material para aprender. A veces encuentra algunos links de guías, pero no logra aprender ya que no tiene a alguien que le diga si va aprendiendo o no

5. ¿Se ha inscrito en algún curso virtual para aprender quechua?

Respuesta: Declara que por el momento si se ha inscrito en un curso, pero dejó de asistir por cuestiones personales y de profesión. Y cada vez que retornaba a las clases online, el profesor ya había pasado de tema y perdió la coherencia.

6. ¿Le gustaría aprender el idioma quechua?

Respuesta: Expresa que es de gran ayuda poder comunicarse con sus alumnos, no solo con ellos sino también el director, padres de familias u otras personas de la localidad.

7. ¿Cree usted que con la ayuda de un software le ayudaría a aprender el idioma?

Respuesta: Expresó que si sería bueno porque tendría una ayuda y le daba su reporte de logro de aprendizaje.



Prof. Oscar del Carmen Bernilla Carlos
DIRECTOR

ANEXO N° 04. MANUAL DE USUARIO

The screenshot shows a registration form titled "Registro". At the top, there are two tabs: "Entrar" and "Registro", with "Registro" being the active tab. Below the tabs, there are three input fields: "Usuario" (with a person icon), "Clave" (with a lock icon), and "Correo" (with an envelope icon). At the bottom of the form is a large orange button labeled "Registrarse".

Ilustración 1: Interfaz de registro de usuarios

The screenshot shows a login form titled "Ingreso". At the top, there are two tabs: "Entrar" and "Registro", with "Entrar" being the active tab. Below the tabs, there are two input fields: "Usuario" (with a person icon) and "Clave" (with a lock icon). At the bottom of the form is a large orange button labeled "Iniciar Sesión".

Ilustración 2: Interfaz de inicio de sesión

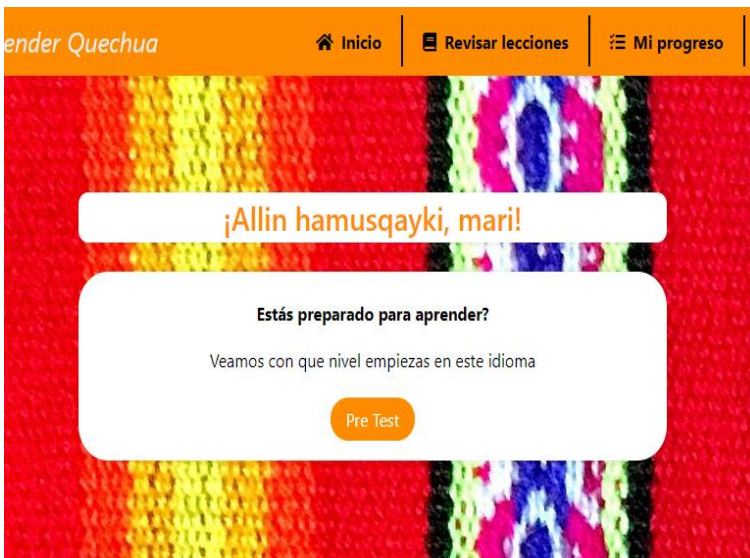


Ilustración 3: Panel de inicio

The screenshot shows a pre-test question interface. At the top, it says "Este test consiste en 20 preguntas con la finalidad de probar su nivel de aprendizaje. Selecciona la respuesta correcta:". Below this, the question is numbered "1:" and asks "Cuál es la traducción correcta: 'Mi nombre es...?'. There are two cartoon characters, a girl and a boy, next to the question. Below the question are three radio button options: "Sutiyimi", "Maypi tiyakunki", and "Maymanta kanki". At the bottom left, there is a blue button labeled "Sig."

Ilustración 4: Interfaz de preguntas en el Pre-test

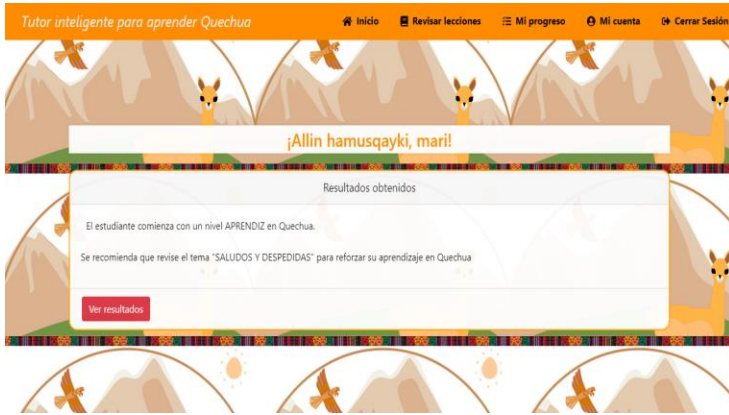


Ilustración 5: Interfaz de resultados del desarrollo

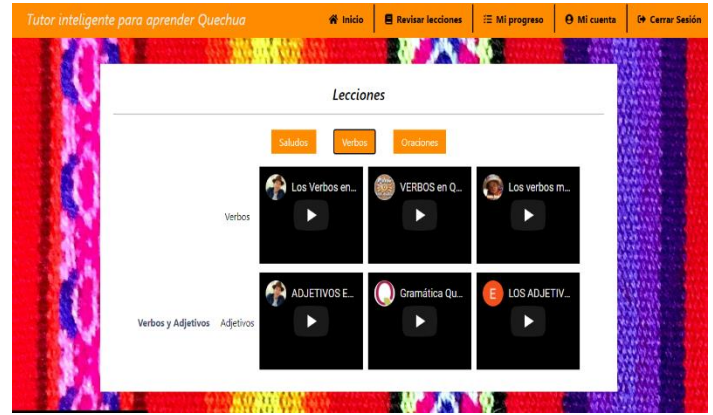


Ilustración 6: Interfaz de revistar temas

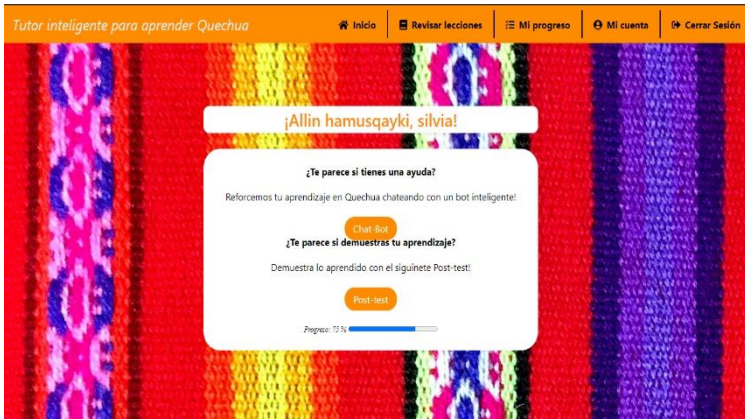


Ilustración 7: Interfaz de Post-test y chat-bot

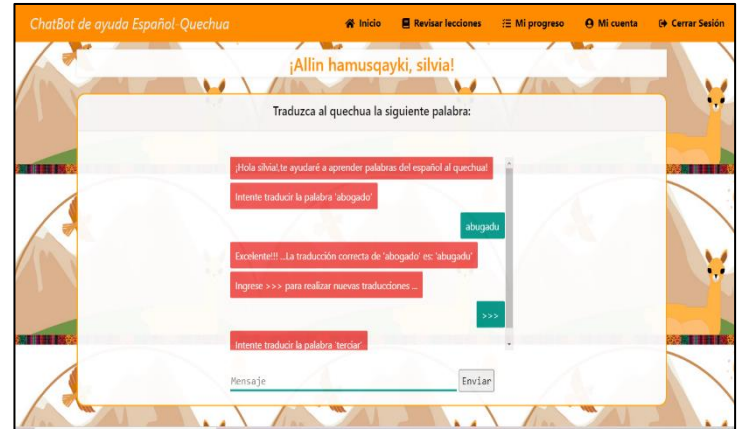


Ilustración 8: Interfaz de interacción con el chat-

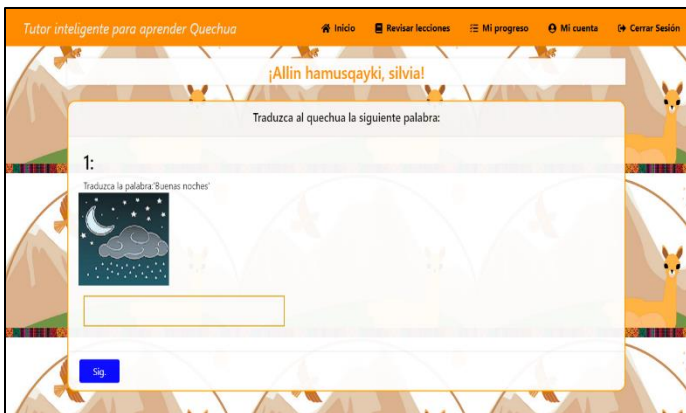


Ilustración 9: Interfaz de interacción con el Post-

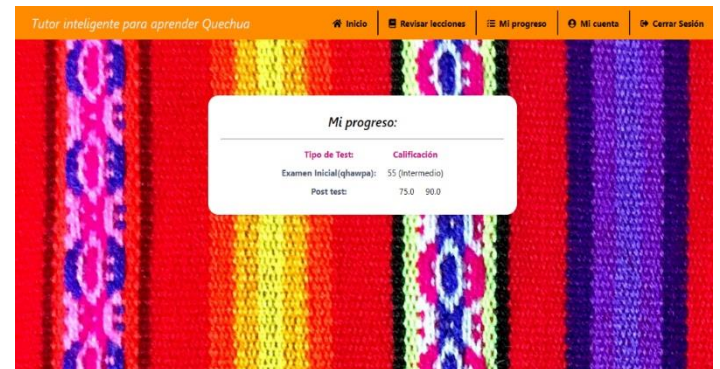


Ilustración 10: Interfaz de Progreso

ANEXO N° 05. FICHA DE ENCUESTA CON EL MODELO DE ACEPTACIÓN TECNOLÓGICA

Utilidad percibida

Indicador

Índice alto= Muy útil

Índice bajo= No útil

Escala tipo Likert de 7 puntos

	1	2	3	4	5	6	7	
Improbable	Totalmente en desacuerdo	Muy en desacuerdo	En desacuerdo	Indiferente	De acuerdo	Muy de acuerdo	Totalmente de acuerdo	Probable

Cuestionario

Por favor marque con una [X] según corresponda.

	Calificación						
1. El Sistema Tutor Inteligente ayuda a conseguir información necesaria para el aprendizaje de la escritura del idioma.	1	2	3	4	5	6	7
2. Utilizando el Tutor Inteligente mejoraría mi escritura en el idioma	1	2	3	4	5	6	7
4. El Tutor Inteligente es útil en mi estudio.	1	2	3	4	5	6	7
5. Usando el Tutor Inteligente me permite saber mi nivel de mi aprendizaje (pre-test)	1	2	3	4	5	6	7
6. El tutor inteligente me da ayuda previa a mis errores en mis pruebas (post-test)	1	2	3	4	5	6	7



 Prof. Oscar del Carmen Berniña Carlos

 DIRECTOR

Facilidad de uso percibida

Indicador

Índice alto= Fácil

Índice bajo= Difícil

Escala tipo Likert de 7 puntos

	1	2	3	4	5	6	7	
Improbable	Totalmente en desacuerdo	Muy en desacuerdo	En desacuerdo	Indiferente	De acuerdo	Muy de acuerdo	Totalmente de acuerdo	Probable

Cuestionario

Por favor marque con una [X] según corresponda.

	Calificación						
1. Es fácil el uso del tutor inteligente	1	2	3	4	5	6	7
2. Me resulta interactivo el uso del sistema Tutor Inteligente.	1	2	3	4	5	6	7
3. La interacción con el tutor Inteligente es clara y entendible.	1	2	3	4	5	6	7
4. El Tutor Inteligente es flexible para interactuar.	1	2	3	4	5	6	7
5. Sería fácil para mí llegar a ser hábil en el uso Tutor Inteligente.	1	2	3	4	5	6	7
6. El tutor inteligente presenta un orden en los temas haciendo que sea fácil encontrarlos.	1	2	3	4	5	6	7



Prof. Oscar del Carmen Bernita Carlos
DIRECTOR

Utilidad percibida

Indicador

Índice alto= Muy útil

Índice bajo= No útil

Escala tipo Likert de 7 puntos

	1	2	3	4	5	6	7	
Improbable	Totalmente en desacuerdo	Muy en desacuerdo	En desacuerdo	Indiferente	De acuerdo	Muy de acuerdo	Totalmente de acuerdo	Probable

Cuestionario

Por favor marque con una [X] según corresponda.

	Calificación						
1. El Sistema Tutor Inteligente ayuda a conseguir información necesaria para el aprendizaje de la escritura del idioma.	1	2	3	4	5	6	<input checked="" type="checkbox"/>
2. Utilizando el Tutor Inteligente mejoraría mi escritura en el idioma	1	2	3	4	5	<input checked="" type="checkbox"/>	7
3. El Tutor Inteligente es útil en mi estudio.	1	2	3	4	5	<input checked="" type="checkbox"/>	7
4. Usando el Tutor Inteligente me permite saber mi nivel de mi aprendizaje (pre-test)	1	2	3	4	5	6	7
5. El tutor inteligente me da ayuda previa a mis errores en mis pruebas (post-test)	1	2	3	4	5	6	<input checked="" type="checkbox"/>
6. El Sistema Tutor Inteligente ayuda a conseguir información necesaria para el aprendizaje de la escritura del idioma.	1	2	3	4	5	<input checked="" type="checkbox"/>	7



Prof. Oscar del Carmen Bernita Carlos
DIRECTOR

Facilidad de uso percibida

Indicador

Índice alto= Fácil

Índice bajo= Difícil

Escala tipo Likert de 7 puntos

	1	2	3	4	5	6	7	
Improbable	Totalmente en desacuerdo	Muy en desacuerdo	En desacuerdo	Indiferente	De acuerdo	Muy de acuerdo	Totalmente de acuerdo	Probable

Cuestionario

Por favor marque con una [X] según corresponda.

	Calificación						
1. Es fácil el uso del tutor inteligente	1	2	3	4	5	6	<input checked="" type="checkbox"/>
2. Me resulta interactivo el uso del sistema Tutor Inteligente.	1	2	3	4	5	<input checked="" type="checkbox"/>	7
3. La interacción con el tutor Inteligente es clara y entendible.	1	2	3	4	5	<input checked="" type="checkbox"/>	7
4. El Tutor Inteligente es flexible para interactuar.	1	2	3	4	5	<input checked="" type="checkbox"/>	7
5. Sería fácil para mí llegar a ser hábil en el uso Tutor Inteligente.	1	2	3	4	5	<input checked="" type="checkbox"/>	7
6. El tutor inteligente presenta un orden en los temas haciendo que sea fácil encontrarlos.	1	2	3	4	5	6	<input checked="" type="checkbox"/>



Prof. Doctor del Carmen Berniba Carlos
DIRECTOR

ANEXO N° 06. VALIDACIÓN POR JUICIO DE EXPERTOS**JUICIO DE EXPERTOS**

Esta ficha tiene como finalidad obtener la validación del modelo de la red neuronal recurrente quien apoya en la traducción de palabras en español a quechua.

Indicador

Índice alto= Probable

Índice bajo= Improbable

Escala tipo Likert de 7 puntos

	1	2	3	4	5	6	7	
Improbable	Totalmente en desacuerdo	Muy en desacuerdo	En desacuerdo	Indiferente	De acuerdo	Muy de acuerdo	Totalmente de acuerdo	Probable

Questionario

Por favor marque con una [X] según corresponda.

	Calificación						
1. La construcción de una red neuronal recurrente de tipo LSTM ayuda en las traducciones de idiomas.	1	2	3	4	5	<input checked="" type="checkbox"/>	7
2. Las entradas establecidas en la red neuronal son correctas.	1	2	3	4	5	<input checked="" type="checkbox"/>	7
3. El uso de las 45 neuronas dentro de la red neuronal es suficiente para que el modelo sea eficiente.	1	2	3	4	5	<input checked="" type="checkbox"/>	7
4. Los resultados de la prueba de entrenamiento fueron adecuados.	1	2	3	4	5	<input checked="" type="checkbox"/>	7
5. Se evidencia que el modelo de la red neuronal recurrente si traduce las palabras a quechua.	1	2	3	4	5	<input checked="" type="checkbox"/>	7

Nombres y apellidos: Raúl Eduardo Huarote Zegarra

D.N.I: 32983830

Grado académico: Maestría

JUICIO DE EXPERTOS

Esta ficha tiene como finalidad obtener la validación del modelo de la red neuronal recurrente quien apoya en la traducción de palabras en español a quechua.

Indicador

Índice alto= Probable

Índice bajo= Improbable

Escala tipo Likert de 7 puntos

	1	2	3	4	5	6	7	
Improbable	Totalmente en desacuerdo	Muy en desacuerdo	En desacuerdo	Indiferente	De acuerdo	Muy de acuerdo	Totalmente de acuerdo	Probable

Cuestionario

Por favor marque con una [X] según corresponda.

	Calificación						
1. La construcción de una red neuronal recurrente de tipo LSTM ayuda en las traducciones de idiomas.	1	2	3	4	5	6	✗
2. Las entradas establecidas en la red neuronal son correctas.	1	2	3	4	5	6	✗
3. El uso de las 45 neuronas dentro de la red neuronal es suficiente para que el modelo sea eficiente.	1	2	3	4	5	6	✗
4. Los resultados de la prueba de entrenamiento fueron adecuados.	1	2	3	4	5	6	✗
5. Se evidencia que el modelo de la red neuronal recurrente si traduce las palabras a quechua.	1	2	3	4	5	6	✗

Nombres y apellidos: Mitchell Paulo Blancas Nuñez

D.N.I: 45551142

Grado académico: Maestría