

UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO



**FACULTAD DE CIENCIAS DE LA INGENIERÍA
INGENIERÍA EN TELEMÁTICA**



DOCENTE:

Ing. Guerrero Ulloa Gleiston Cicerón

MATERIA:

Aplicaciones Telemáticas Basadas En La Web

AUTORES:

Cáceres Rivas Edison Michael

Jaramillo Mora Julio Ivonik

Loza Yanez Kevin Adrian

Quintero Arias Dany Mishel

CURSO:

8^{vo} Telemática

FECHA:

9 de Marzo del 2023

PROYECTO:

“Traductor A Lenguajes De Señas”

ECUADOR – LOS RIOS – QUEVEDO

2022-2023

INDICE

1.	TÍTULO	1
2.	AFILIACIONES.....	2
3.	RESUMEN.....	2
4.	INTRODUCCIÓN.....	2
5.	REVISIÓN DEL ESTADO DEL ARTE:.....	3
6.	SISTEMA PROPUESTO	6
7.	METODOLOGÍA DE DESARROLLO	6
8.	RESULTADO Y DISCUSIONES	9
8.1.	Resultados	9
8.2.	Discusión	9
9.	CONCLUSIONES Y RECOMENDACIONES	9
9.1.	Conclusión.....	9
9.2.	Recomendaciones	10
10.	BIBLIOGRAFÍA.....	10
11.	ANEXOS	11
11.1.	ANEXO A: MANUAL DE USUARIO	11
11.2.	ANEXO B: REPORTE DE ACIERTO Y DESACIERTO	16
11.3.	ANEXO C: MANUAL TÉCNICO.....	17

ILUSTRACIONES

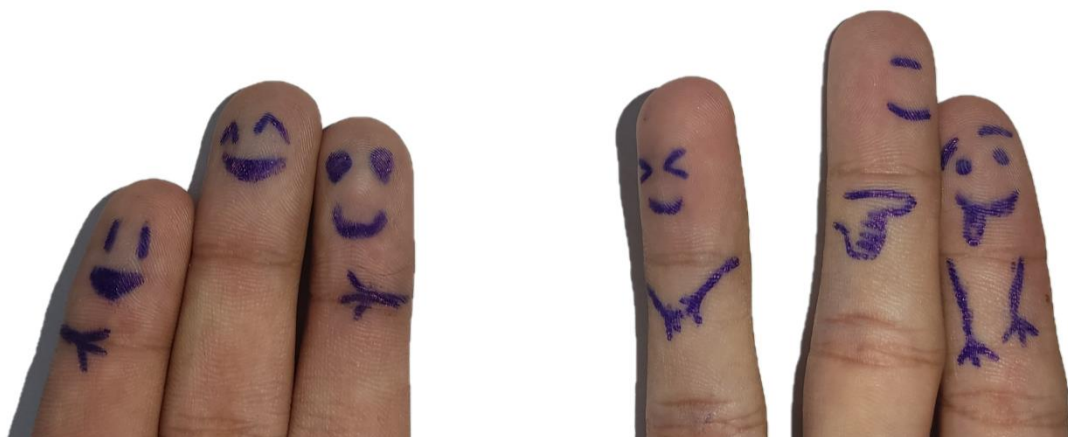
ILUSTRACIÓN 1:	DIAGRAMA DE FLUJO INICIAL PENSANDO PARA EL DESARROLLO DE LA APLICACIÓN WEB ..	8
ILUSTRACIÓN 2:	PÁGINA INICIAL DEL TRADUCTOR (SESIÓN ADMINISTRADOR).	11
ILUSTRACIÓN 3:	PÁGINA INICIAL DEL TRADUCTOR (SESIÓN ADMINISTRADOR).	12
ILUSTRACIÓN 4:	PÁGINA PRINCIPAL DEL TRADUCTOR (SESIÓN ADMINISTRADOR).	12
ILUSTRACIÓN 5:	PÁGINA PRINCIPAL DEL TRADUCTOR (SESIÓN USUARIO INVITADO).	13
ILUSTRACIÓN 6:	PASOS PARA TRADUCIR UNA PALABRA INGRESADA MANUALMENTE.	13
ILUSTRACIÓN 7:	PASOS PARA TRADUCIR UNA PALABRA INGRESADA MANUALMENTE.	14
ILUSTRACIÓN 8:	PASOS PARA INGRESAR UNA PALABRA POR MEDIO DEL MICRÓFONO Y LUEGO TRADUCIRLA.	14
ILUSTRACIÓN 9:	SECCIÓN DONDE SE MUESTRA LOS RESULTADOS DE LAS PALABRAS TRADUCIDAS.....	15
ILUSTRACIÓN 10:	GRÁFICO SOBRE LOS ACIERTOS Y DESACIERTOS.....	16

TABLAS

TABLA 1:	TOTAL DE ACIERTOS Y DESACIERTOS.	16
----------	---------------------------------------	----

1. TÍTULO

“Hablando con
las manos,
traductor del
español escrito al
lenguaje de
señas.”



2. AFILIACIONES

- Ing. Gleiston Guerrero, supervisor de Proyecto Final en la materia “Aplicaciones Telemáticas basadas en la Web”.
- Facultad de Ciencias de la Ingeniería, Universidad Técnica del Estado de Quevedo, Quevedo 27/01/2023, Ecuador
- Correspondencia: dquintero@uteq.edu.ec, jjaramillom@uteq.edu.ec, klozay@uteq.edu.ec, ecaceresr2@uteq.edu.ec, gguerrero@uteq.edu.ec.

3. RESUMEN

En la Unidad Educativa Especializada “Fe y Alegría” localizada en la ciudad de “Santo Domingo De los Tsáchilas”, se presenta un problema con los integrantes de dicha institución y personas en general que la conforman, por aquello se expone una aplicación web la cual tiene como función principal traducir palabras del español escrito a lenguajes de señas, se presenta bajo el nombre de “Hablando Con Las Manos”.

Está desarrollada como tema de proyecto final de la materia “Aplicaciones Telemáticas Basadas en la Web”, la misma que beneficia a la educación para las personas que utilizan el lenguaje de señas y las que no la dominan, se realizó con el programa “NetBeans” con códigos basados en lenguajes de programación dirigidos para la interpretación de los navegadores.

Para empezar, se realizó un diseño previo necesario para presentar una interfaz amigable e intuitiva, seguidamente se recopiló los datos necesarios para el funcionamiento interno de la aplicación web, por último, se desarrolla el código basado con toda la información y términos obtenidos de acuerdo con las necesidades que se presentaron.

4. INTRODUCCIÓN

Las personas con discapacidad auditiva poseen una gran dificultad para llevar a cabo una comunicación verbal, debido a que por su condición no puede escuchar el sonido simbólico de cada letra y por ende este genera su conflicto al tratar de comunicarse, sin embargo, también existe las personas que, si poseen esta capacidad auditiva, pero tiene la incapacidad de vocalizar las palabras.

El 3 de diciembre del 2022 se registró en el Ecuador un aproximado de 75.418 habitantes que presenta discapacidad auditiva, los cuales no poseen alternativas para solventar su discapacidad, sin embargo, son los niños los cuales muestran más oportunidades para poder solucionar dicho problema, aunque para aquello actualmente es necesario acudir a un centro especializado.

Es evidente que en el Ecuador no todas las Unidades Educativas poseen planes de educación para las personas con discapacidad auditiva. Por otra parte, entre las Unidades Educativas Ecuatorianas que poseen planes dirigido el desarrollo educativo para dichas personas, entre ellas existe la “Unidad Educativa Fe y Alegría” la cual aun siendo un centro especializado presenta problemáticas.

Por aquello, “Hablando Con Las Manos” es el nombre de la Aplicación Web que se realiza como proyecto final de la materia Aplicaciones Telemáticas Basadas en la Web, el cual es un traductor enfocado a convertir el español escrito a lenguaje de señas. Esta iniciativa se toma a raíz de un problema social que se presenta en la Unidad Educativa Especializada “Fe y Alegría”.

En esta Unidad Educativa existen personas con dificultades auditivas, estas generan una problemática a la hora de su comunicación y educación, esta condición física presenta un conflicto verbal tanto para el individuo que lo padece como para las personas que desea comunicarse con el individuo.

Para solventar esta problemática se procede a desarrollar el presente proyecto con el objetivo de facilitar el proceso de comunicación para ambas partes, además con el uso de este traductor se desea promover la educación en esta área para el beneficio de los individuos que tienen estas características físicas.

Cuando se realizó la visita a la institución se pudo constatar las dificultades que se presenta al momento de hablar o enseñar a los niños que tienen esta característica física, la experiencia que exponían el cuerpo de profesores era muy inspiradora además de compleja, esta condición no solo afecta al individuo, también afecta a los familiares cercanos.

Por ende, al presentar este proyecto se busca facilitar estos aspectos cotidianos aumentando el aprendizaje del lenguaje de señas para ambas partes, ya sea el emisor o receptor. Aunque para ello se debe capacitar a los docentes encargados para el uso correcto de la Aplicación Web, “Hablando Con Las Manos”.

5. REVISIÓN DEL ESTADO DEL ARTE:

- **Lectura de palabras por personas sordas usuarias de lengua de señas mexicana.**

Word reading in Mexican deaf signers Elizabeth Mendoza, Donna Jackson - Maldonado.

Centro de Estudios Lingüísticos y Literarios, Facultad de Lenguas y Letras, Universidad Autónoma de Querétaro, Campus Aeropuerto, Santiago de Querétaro, Querétaro, México. [1]

Contexto: Este trabajo fue diseñado en el país de México en el año 2020 y tiene como objetivo determinar si las personas sordas pueden entender el lenguaje de señas en la lectura de palabras escritas en español. Se usaron pares de palabras semánticamente relacionadas y semánticamente no relacionadas, divididas a su

vez 2 subgrupos, fonológicamente relacionadas y fonológicamente no relacionadas en el lenguaje de señas. Los participantes respondieron más rápido a las palabras semántica y fonológicamente relacionadas, mientras que, por el contrario, respondieron más lento en las palabras semánticamente no relacionadas que sí estaban fonológicamente relacionadas.

Comparación: Este proyecto está orientado más a la educación de las personas sordas y “hablando con las manos” es una aplicación web que permite la comunicación con las personas que presentan discapacidad auditiva.

- **Desarrollo de algoritmos para traductor automático de lenguaje de señas mexicanas (LSM).**

ESPEJEL CABRERA, JOSUE Fecha: 2021-10-27. [2]

Contexto: El presente estudio tiene como fin el analizar los diferentes algoritmos para poder crear un traductor capaz de ayudar a la comunicación entre una persona sorda y una oyente, esto es un estudio a futuro mientras “Hablando con las manos” es un traductor que ya está funcionando.

Comparación: Mientras que Espejel Cabrera [2], se enfoca en el desarrollo de algoritmos para traducir LSM a texto, el traductor de lenguaje de señas tiene una funcionalidad diferente al ser capaz de recibir comandos de voz para traducir texto a lenguaje de señas, lo que lo hace útil para aquellos que no pueden escribir o no tienen acceso a un teclado. Además, la capacidad de agregar imágenes en la interfaz de administración también puede mejorar la experiencia del usuario.

- **Impacto de la aplicación Reader como herramienta de apoyo en la lectoescritura de las personas con discapacidad auditiva de la Asociación Central de Sordos El Alto.**

Carlos Eduardo Rodrigo Lozal. [3]

Contexto: La presente aplicación tiene como objetivo establecer si la aplicación Reader es una herramienta efectiva para el aprendizaje de la sintaxis del idioma español ya que es una app diseñada para que las personas con discapacidad auditiva puedan aprender a leer el español escrito de una manera más simple y sin problemas y a la vez también puede ser usada por personas no discapacitadas.

Comparación: Esta aplicación está orientada para la educación de las personas con discapacidades y “hablando con las manos” es una aplicación que ayuda a la comunicación con personas que presentan discapacidad auditiva.

- **Comunicación mediante una interfaz amigable por texto o reconocimiento de voz desarrollado en Android Studio.**

MYRIAN CARGUACUNDO, PATRICIA CONSTANTE Publicado: 2019-04-30. [4]

Contexto: El proyecto tiene como objetivo crear una herramienta que permita a las personas sordas o con discapacidad auditiva comunicarse de manera efectiva en situaciones cotidianas. El traductor funciona mediante la entrada de texto o voz, que luego se convierte en señas a través de un avatar animado que las representa. El proyecto también incluye una base de datos de palabras y frases comunes en la lengua de señas ecuatoriana, para mejorar la precisión y la eficacia del traductor.

Comparación: El presente proyecto tiene como fin lograr la comunicación de las personas oyentes con las personas sordas mediante una app con ingreso de voz y “hablando con las manos” puede desarrollar la traducción mediante el ingreso de voz y mediante un cuadro de texto se puede ingresar la palabra a traducir.

- **Diseño e implementación de un sistema traductor de lenguaje de señas de manos a un lenguaje de texto mediante visión artificial en un ambiente controlado**

Edwin Chiguano, Nathaly Moreno, Luis Corrales

Facultad de Ingeniería Eléctrica y Electrónica, EPN Quito, Ecuador [5]

Contexto: En este proyecto se ha realizado un sistema que traduce de lenguaje de señas a lenguaje de texto usando visión artificial, para permitir que personas con discapacidad auditiva y/o visual que manejan el lenguaje de señas puedan comunicarse con el resto de las personas que no manejan este lenguaje.

Comparación: El presente proyecto comparando con “hablando con las manos” uno traduce mediante visión artificial y el otro mediante voz y video.

- **Desarrollo de una aplicación móvil para traductor de lenguaje de señas mediante el uso de servicios web**

Geraldin Pinzón Bayona, Yesid Giovanni Sanabria Orjuela

Universidad Católica de Colombia facultad de ingeniería ingeniería de sistemas y computación bogotá 2021 [6]

Contexto: El autor se enfoca en la importancia de tener una aplicación de traducción de idiomas en la actualidad, debido a la creciente necesidad de comunicación entre personas de diferentes idiomas y culturas. El artículo describe el proceso de desarrollo de la aplicación, incluyendo la selección de tecnologías, la definición de requisitos y la implementación del software, también describe cómo se creó una aplicación móvil para la traducción de idiomas y los aspectos clave del proceso de desarrollo de la misma.

Comparación: Esta aplicación consiste en una aplicación móvil desarrollada con el software Poser para el diseño de la aplicación GUI, a su vez se usó AppInventor y el software Adobe Flex para la creación de la aplicación móvil, el reconocimiento y registro de texto a comparación de la aplicación web “hablando con las manos” que está desarrollada con Neatbeans.

- **Prototipo traductor en tiempo real para la población sorda en Colombia.**

Johan Daniel Aguirre Carpetá, Daniel Mauricio Valderrama Perdomo

Universidad Distrital Francisco José de Caldas facultad tecnológica tecnología en sistematización de datos Bogotá dc. 2018 [7]

Contexto: El objetivo principal del proyecto o del prototipo en general es ayudar a las personas sordas a comunicarse con hablantes de otros idiomas en tiempo real utilizando tecnología de reconocimiento de voz y traducción automática. El trabajo describe la metodología utilizada para el desarrollo del prototipo, incluyendo el análisis de requisitos, la selección de tecnologías y la implementación del software, esta describe un trabajo de grado que presenta un prototipo de traductor en tiempo real para la población sorda, y también la

metodología utilizada para el completo desarrollo del prototipo.
Comparación: Este proyecto está realizado con LEAP MOTION dónde el manejo de información se hace por medio de un enlace en el cual la persona interactúa con sus manos y el programa identifica la señal, traduciéndola en palabras o en voz en tiempo real y nuestro traductor es desarrollado en Neatbeans permitiendo ingresar la palabra a traducir mediante texto o voz.

- **Diseño de un guante con sensores de flexibilidad que traducen letras del abecedario del lenguaje sordo mudo utilizando microphyton**

Arianna Lisbeth Morales Iturralde

Universidad Politécnica Salesiana sede Guayaquil [8]

Contexto: Este proyecto trata sobre un guante que, al usarlo y hacer la seña, su traducción será mostrada en una pantalla realizada en un software libre,

Comparación: Al momento de comparar el presente proyecto con el traductor “hablando con las manos” la diferencia que uno traduce mediante el uso de un guante, y el otro necesita ingresar la palabra ya sea mediante un texto o hablando

6. SISTEMA PROPUESTO

La aplicación web “Hablando con las manos, traductor de español escrito a lenguaje de señas” es un traductor que nos permite la comunicación con las personas que presentan discapacidad auditiva.

Este traductor está realizado en el entorno de desarrollo libre Neatbeans, el cual nos permite hacer consultas de señas cargadas en el programa, podemos ingresar una palabra y el traductor la buscare en su base de datos, el mismo nos muestra en pantalla la imagen relacionada con el texto antes ingresado, además no solo podemos utilizar el ingreso de texto por teclado ya que este traductor también está desarrollado para que las personas puedan dictar la palabra y la aplicación web va a realizar el mismo proceso que se hace al ingresar texto manualmente.

Cabe recalcar que este traductor permite hacer el ingreso de más archivos con formatos .gif .jpg .png para que el administrador pueda añadir más contenido con el cual el usuario final pueda utilizarlo en cada traducción.

7. METODOLOGÍA DE DESARROLLO

“Hablando con las manos” se originó y se desarrolló con el fin de poder entender palabras del español escrito a lenguaje de señas mediante un traductor desarrollado por los integrantes del proyecto.

En el presente proyecto la metodología que usamos va describiendo el proceso que lleva a cabo para el desarrollo de la aplicación web. En la ilustración 1, muestra el diagrama de

flujo como secuencia de ejecución que se utilizará en la aplicación web, la cual, fue desarrollado de forma que tenga mucha facilidad al utilizar el traductor.

Para el desarrollo de “Hablando Con Las Manos”, los requisitos funcionales fueron propuesto por los integrantes del grupo mediante una consulta social sobre las dificultades de las personas con discapacidad auditiva, es decir, el proyecto intenta solventar problemas comunes que se presenta en la comunicación entre una persona con dicha discapacidad y una persona considera medicamente sana.

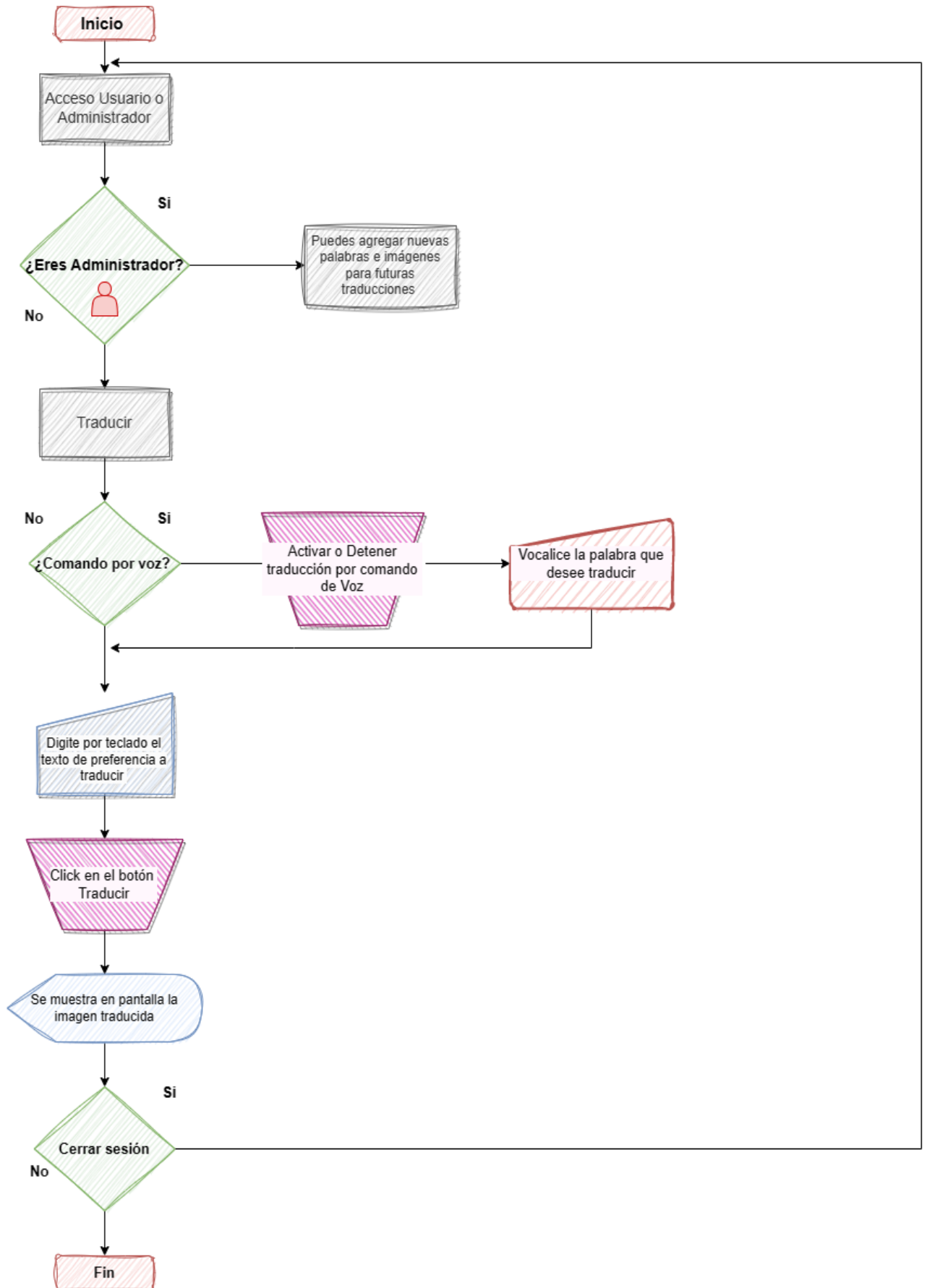


Ilustración 1: Diagrama de flujo inicial pensando para el desarrollo de la aplicación web

La comunicación entre los integrantes del grupo fue presencial y telemática, donde de forma presencial se hizo reuniones con todos los miembros del grupo en los diferentes lugares de la Universidad Estatal De Quevedo. Por tanto, en los cambios o modificaciones, además de la realización de este documento se realizó mediante herramientas de conferencia virtuales, dando así resultados que se presenta en la siguiente sección.

8. RESULTADO Y DISCUSIONES

8.1. Resultados

La aplicación web denominada “hablando con las manos” tuvo los resultados esperados, se pudo comprobar su eficiencia ingresando alguna palabra y la aplicación web devuelve la traducción incluso si le decimos verbalmente una palabra la aplicación la reconocerá y mostrará en pantalla su significado en lenguaje de señas.

8.2. Discusión

El principal resultado obtenido en nuestro proyecto fue lograr que el traductor realice traducciones certeras. El traductor “hablando con las manos” puede traducir de español escrito a lenguaje de señas o de español dictado a lenguaje de señas.

El traductor ha sido probado con niños que tienen discapacidades auditivas junto con algún individuo considerado físicamente sano y podemos constatar que funciona a la perfección. Las limitaciones que tiene nuestro traductor es que no puede ser ofrecido al mercado para que lo use la comunidad en general.

El traductor “hablando con las manos” seguirá mejorando para que pueda ser una herramienta de total utilidad no solo en la institución “Fe y alegría” si no para todas las áreas en las que sea requerido.

9. CONCLUSIONES Y RECOMENDACIONES

9.1. Conclusión

La discapacidad auditiva es una problemática muy grande a nivel mundial y las personas que la padecen a diario son discriminadas sin razón alguna, la visita a la unidad educativa “Fe y Alegría” ayudo a este grupo de jóvenes a entender el valor de poder escuchar y es este motivo que origino la idea de la creación de un traductor de español escrito a lenguaje de señas denominado “Hablando Con Las Manos”.

Esta aplicación web fue desarrollada con NetBeans ya que es un software que nos permite la creación de aplicaciones web y trabajar con base de datos, una vez finalizada la aplicación se procedió a probarla y fue un éxito y podrá ayudar a muchas personas para

que puedan comunicarse de una manera más eficiente con las personas con discapacidad auditiva.

9.2. Recomendaciones

- El traductor es realizado con fines educativos y debe ser utilizado como tal.
- Al momento de escribir una palabra en la aplicación web no deberá existir faltas ortográficas.
- Al momento de decir una palabra en la aplicación web se deberá hablar de manera clara y pausada

10. BIBLIOGRAFÍA

- [1] D. J.-M. Elizabeth Mendoza, «Lectura de palabras por personas sordas usuarias de lengua,» 2019. [En línea]. Available: <https://www.sciencedirect.com/science/article/abs/pii/S021446031930049X?via%3Dihub>.
- [2] J. E. Cabrera, «Desarrollo de algoritmos para traductor automático de lenguaje de señas mexicanas (LSM).,» 2021. [En línea]. Available: <http://ri.uaemex.mx/handle/20.500.11799/111836>.
- [3] C. E. Rodrigo Lozal, «Impacto de las aplicaciones Reader como herramienta de apoyo en la lectoescritura de las personas con discapacidad auditiva,» 2019. [En línea]. Available: http://scielo.org.bo/pdf/rfer/v17n17/v17n17_a12.pdf..
- [4] M. Carguacundo y P. Constante, Traductor de texto y voz de lengua de señas ecuatoriana a través de un avatar implementado para dispositivos Android, 2019.
- [5] N. M. L. C. Edwin Chiguano, «Diseño e implementación de un sistema traductor de lenguaje de señas,» 2018. [En línea]. Available: https://bibdigital.epn.edu.ec/bitstream/15000/4924/1/PAPER_CHIGUANO_MORENO.pdf.
- [6] G. P. BAYONA, «DESARROLLO DE UNA APLICACIÓN MÓVIL PARA TRADUCTOR DE LENGUAJE,» 2021. [En línea]. Available: <https://repository.ucatolica.edu.co/server/api/core/bitstreams/01081195-75db-44ef-a09c-f64186552411/content>.
- [7] J. D. A. CARPETA, «PROTOTIPO TRADUCTOR EN TIEMPO REAL PARA LA POBLACIÓN SORDA,» 2018. [En línea]. Available: <https://repository.udistrital.edu.co/bitstream/handle/11349/15925/ValderramaPerdomoDanielMauricio2019.pdf?sequence=1&isAllowed=y>.
- [8] A. Morales, «diseño de una guante con sensores para gente sorda muda,» 2021. [En línea].

11. ANEXOS

11.1. ANEXO A: MANUAL DE USUARIO

Este manual contendrá los pasos e indicaciones para el correcto uso de la aplicación web titulada como “Hablando con las manos”, se explicará cada apartado con sus respectivas indicaciones o significados para una mayor comprensión del usuario final.

Paso 1:

Al cargar la página inicial se mostrará un apartado con un “Log In” junto con los botones “acceder” y “traducir”.

Esta página inicial se puede tener dos opciones principales, la del ingreso como administrador de la aplicación web o como usuario invitado para el uso de este. Si el administrador desea ingresar deberá colocar el nombre de usuario junto con su respectiva contraseña (Usuario: administrador | Contraseña: 123456) para de esta manera tener las opciones destinadas para el mismo, una vez escrito los datos para iniciar sesión es necesario seleccionar el botón “acceder” para redirigirnos a la página principal.

Si es un usuario invitado el que desea utilizar la aplicación web solo deberá seleccionar el botón “traducir” e inmediatamente lo redireccionará a la página destinada para el usuario final.

SESIÓN ADMINISTRADOR

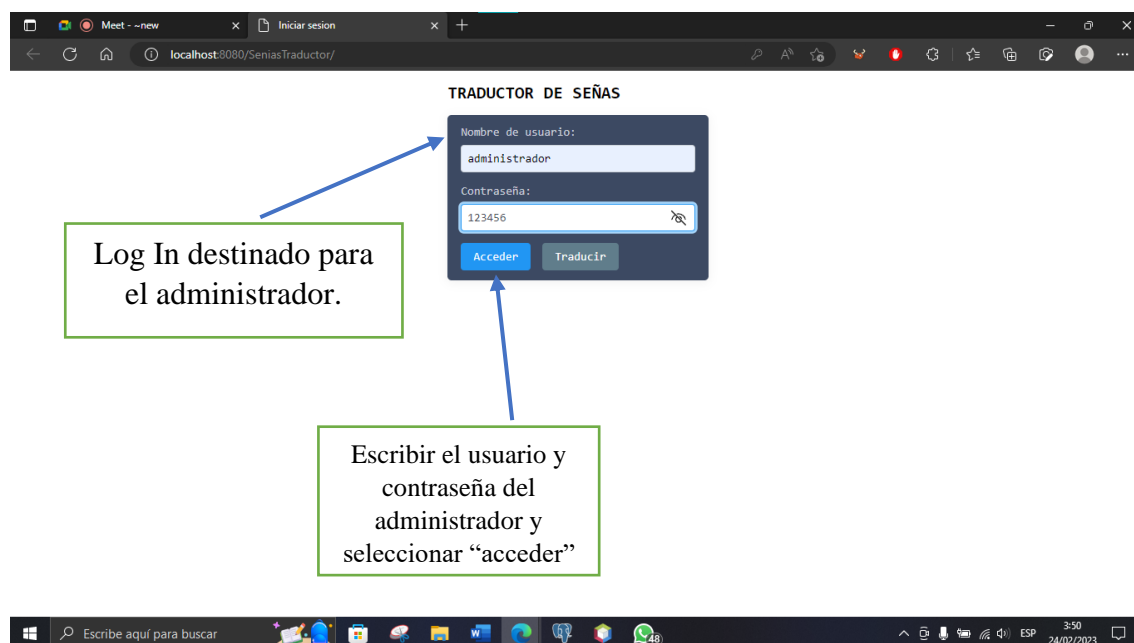


Ilustración 2: Página inicial del traductor (Sesión Administrador).

SESIÓN USUARIO INVITADO

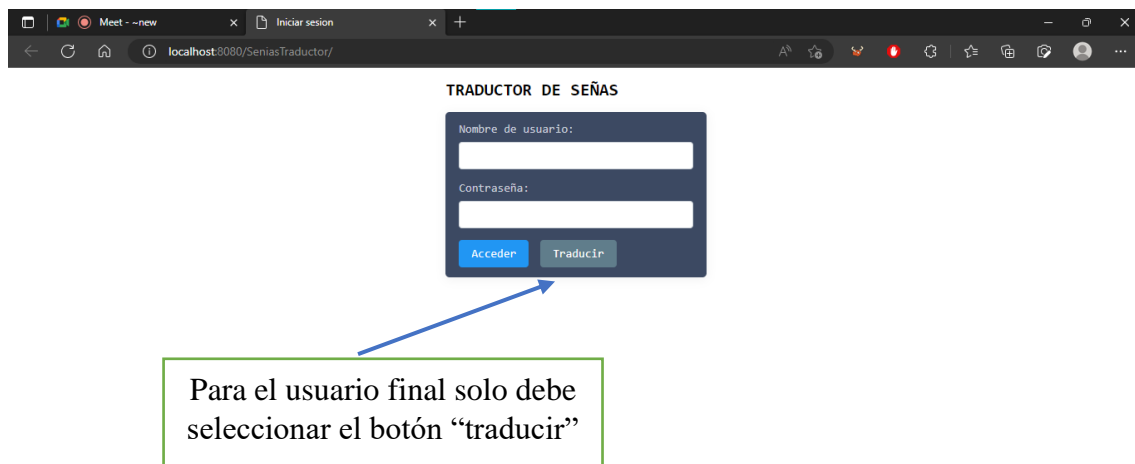


Ilustración 3: Página inicial del traductor (Sesión Administrador).

Paso 2:

Una vez realizado el paso 1, nos direccionará a la página principal donde dependiendo nuestro tipo de “inicio de sesión” (administrador o usuario invitado) tendremos diferentes opciones.

SESIÓN ADMINISTRADOR

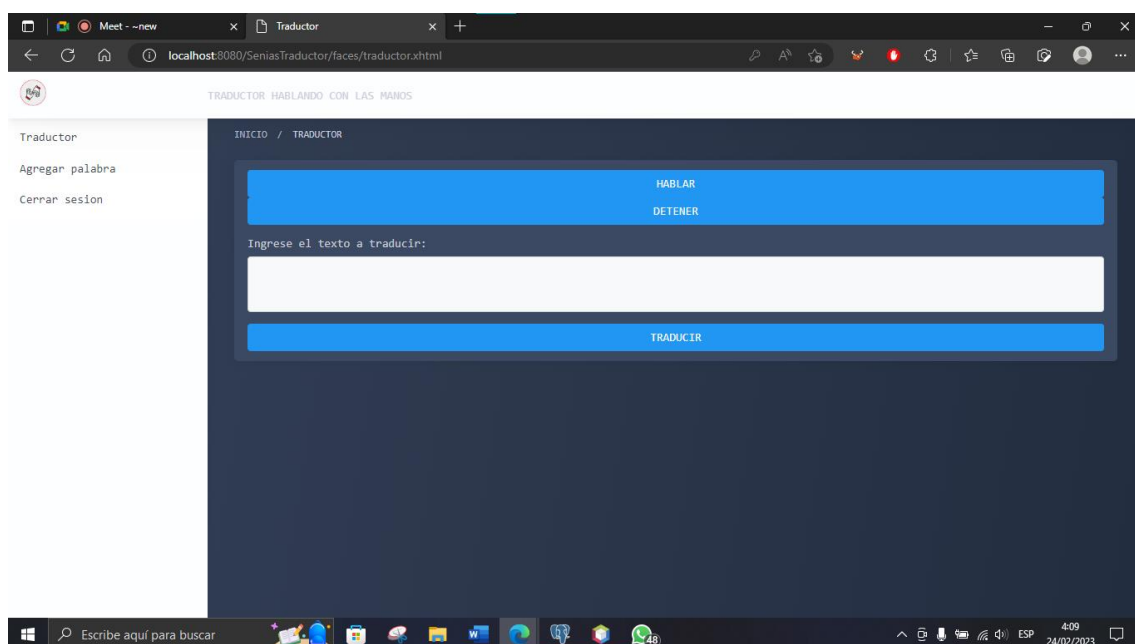


Ilustración 4: Página principal del traductor (Sesión Administrador).

SESIÓN USUARIO INVITADO

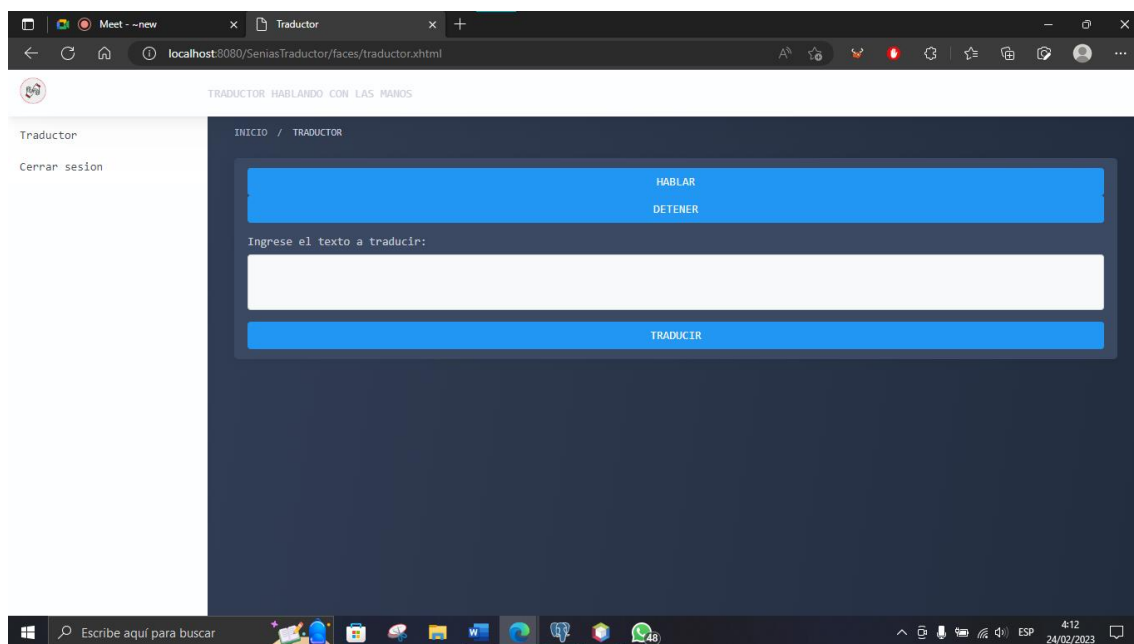


Ilustración 5: Página principal del traductor (Sesión Usuario Invitado).

Paso 3:

En el paso tres se mostrará los pasos para poder utilizar el traductor de forma correcta, donde se ingresará la palabra deseada y al hacer la traducción automáticamente generará un apartado con su respectiva traducción de dicha palabra ingresada. Vale recalcar que este proceso funciona de la misma manera en los dos tipos de usuarios existentes en esta aplicación web (administrador y usuario invitado).

TRADUCCIÓN DE PALABRAS DESEADAS DE FORMA MANUAL

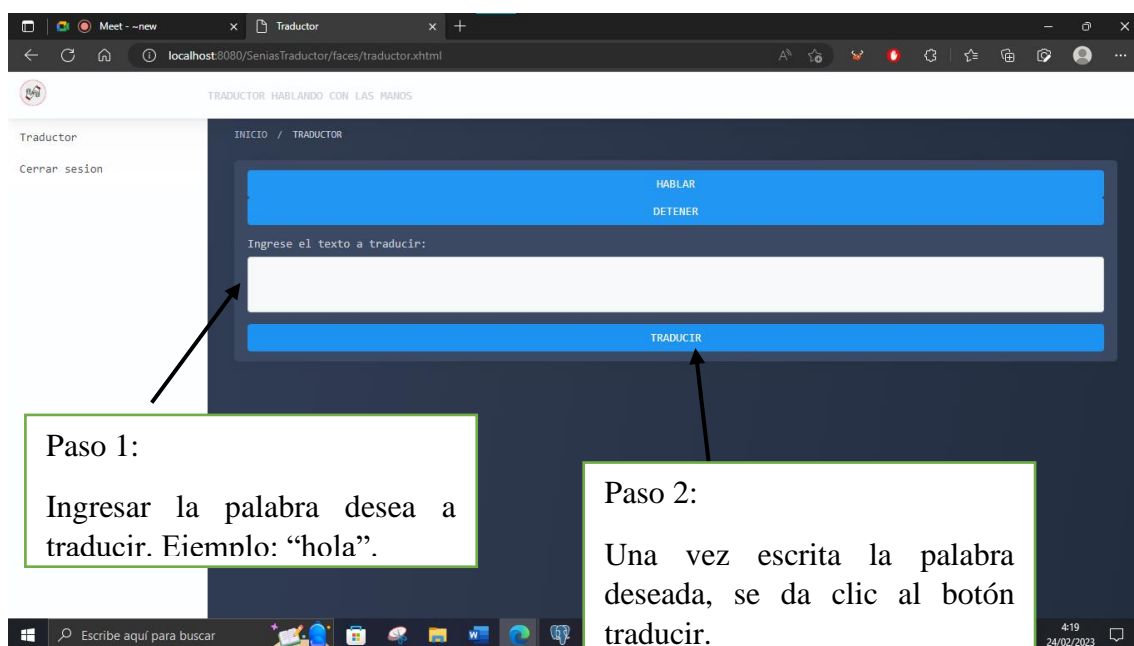


Ilustración 6: Pasos para traducir una palabra ingresada manualmente.

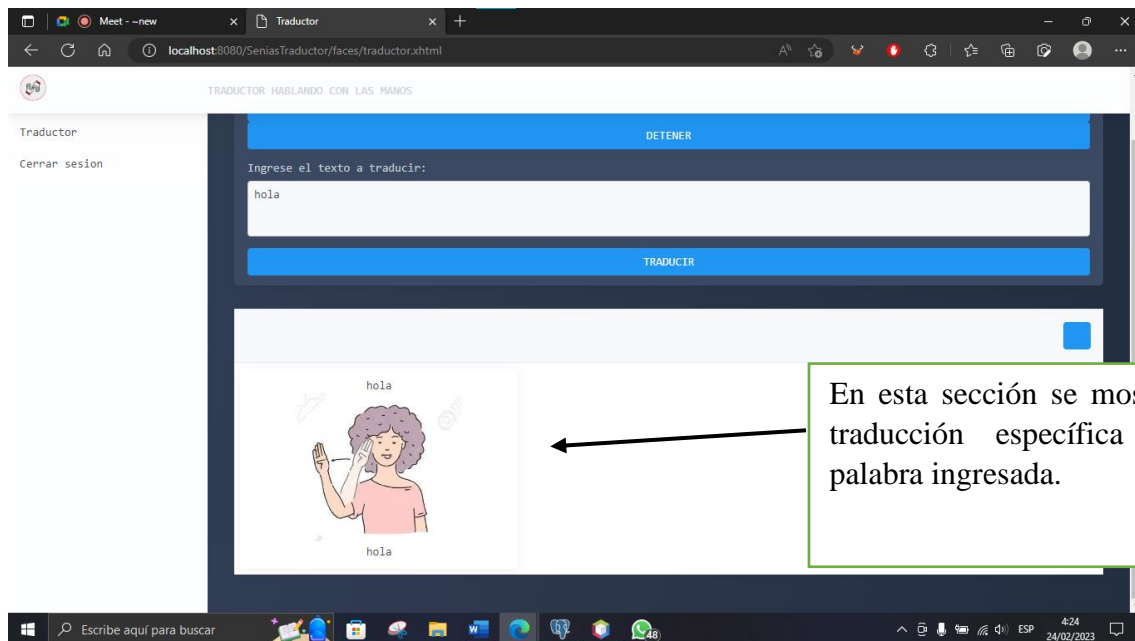
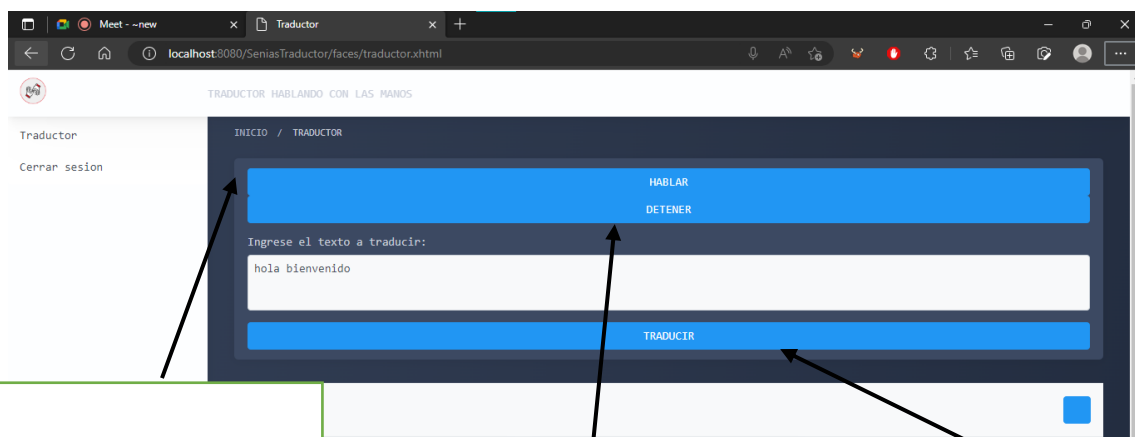


Ilustración 7: Pasos para traducir una palabra ingresada manualmente.

TRADUCCIÓN DE PALABRAS DESEADAS INGRESADA POR MICROFONO



Paso 1:

Primero damos clic en el botón “HABLAR”, seguidamente vocalizamos la palabra deseada, por ejemplo: “hola bienvenido”.

Paso 2:

Cuando se termina de vocalizar las palabras a traducir es necesario dar clic en el botón “DETENER” para que se pueda mostrar lo hablado en el apartado “Ingresa el texto a traducir”.

Paso 3:

Una vez realizados los pasos 1 y 3, solo es necesario pulsar el botón “TRADUCIR” para generar la traducción correspondiente.

Ilustración 8: Pasos para ingresar una palabra por medio del micrófono y luego traducirla.

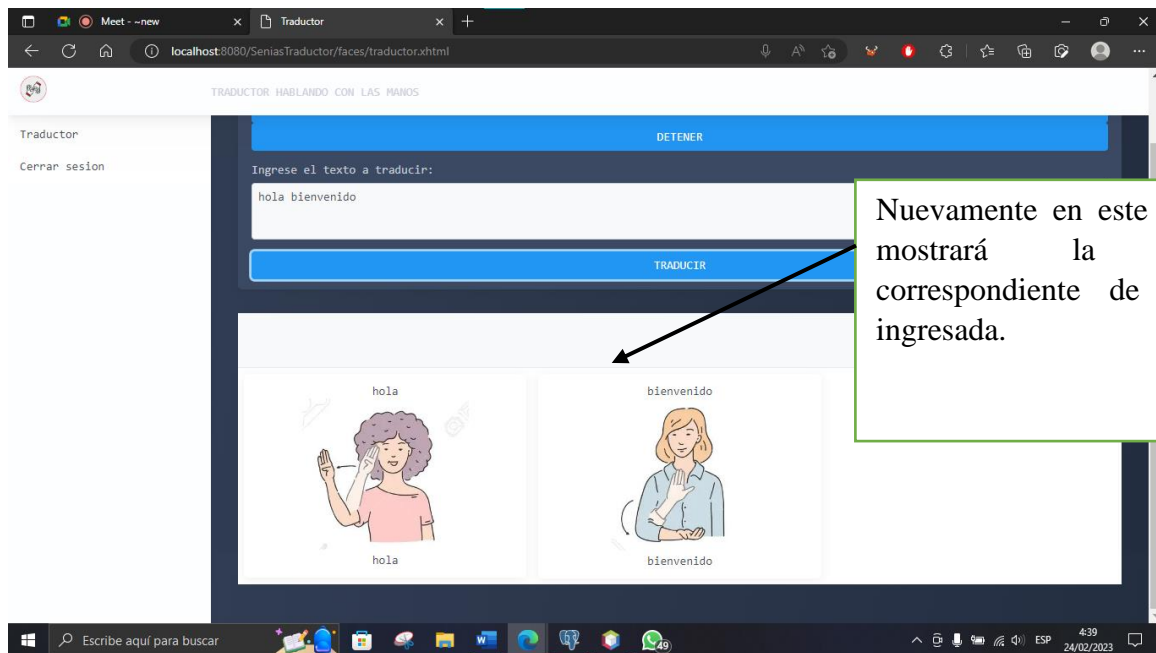


Ilustración 9: Sección donde se muestra los resultados de las palabras traducidas.

11.2. ANEXO B: REPORTE DE ACIERTO Y DESACIERTO

“Hablando con las manos” es un proyecto el cual tiene como acción principal traducir palabras escritas a lenguaje de señas, sin embargo, también posee la funcionalidad de poder ingresar el texto mediante micrófono, es decir, al presionar botón “Hablar” activaremos el micrófono y al presionar el botón “Detener” lo detendremos para que seguidamente se muestre el texto ingresado mediante voz.

Si bien es cierto, existe posibilidades de errores al momento que el dispositivo detecte la voz y muestre correctamente lo dicho, por aquello se tomó una muestra de 200 intentos para mencionar el porcentaje de error detectado en dicha funcionalidad. Para aquello se presente la siguiente tabla.

<i>Tabla Reporte de Error</i>		
<i>Intentos</i>	<i>Aciertos</i>	<i>Desaciertos</i>
200	137	63

Tabla 1: Total de aciertos y desaciertos.

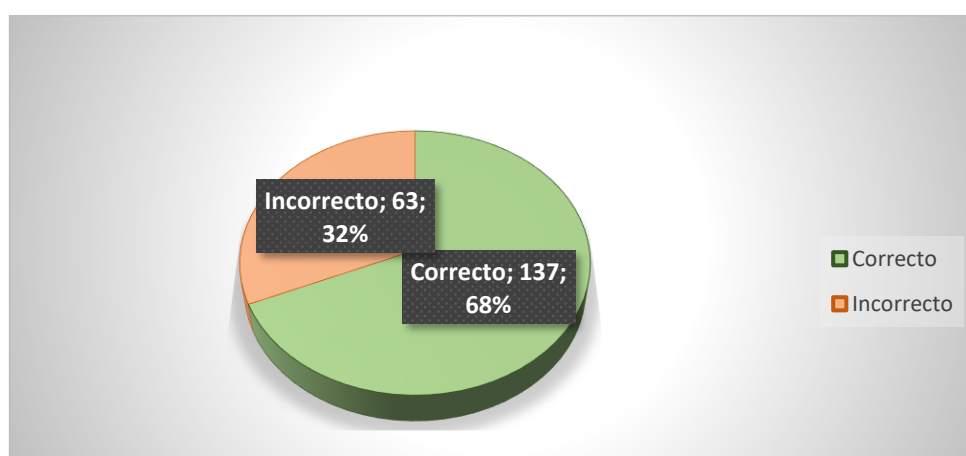


Ilustración 10: Gráfico sobre los aciertos y desaciertos.

De acuerdo con la ilustración presentada vemos que esta funcionalidad muestra una cantidad de desaciertos, donde de los 200 intentos realizados 63 de estos fueron erróneos debido a que micrófono no detectaba la frase completa o simplemente no encendía el micrófono, si bien es cierto, esto puede presentar una dificultad a la hora de utilizar dicho traductor, aun así, son 137 aciertos que se efectuaron con total cumplimiento.

Suponiendo que los aciertos y desaciertos se producen de manera aleatoria, la probabilidad de que se produzca un error después de un número determinado de intentos es igual a la probabilidad de que el último intento sea un desacierto (es decir, 0.32). Por lo tanto, podemos estimar que, en promedio, se producirá un error cada $1 / 0.32 = 3.125$ intentos.

Sin embargo, es importante tener en cuenta que este es solo un cálculo aproximado y que la frecuencia real de los errores puede variar debido a factores como el nivel de ruido de fondo o la calidad del micrófono.

11.3. ANEXO C: MANUAL TÉCNICO

Herramientas utilizadas

Para la creación de este proyecto se utilizó diferentes herramientas, las cuales son:

- NetBeans
- Microsoft Edge
- PostgreSQL
- Payara Server
- PgAdmin

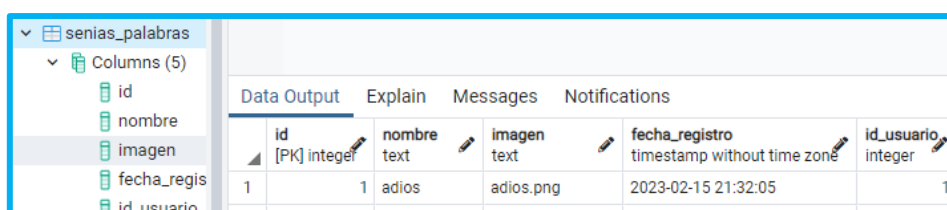
Dependencias

```
<dependencies>
  <dependency>
    <groupId>jakarta.platform</groupId>
    <artifactId>jakarta.jakartaee-api</artifactId>
    <version>${jakartaee}</version>
    <scope>provided</scope>
  </dependency>
  <!-- https://mvnrepository.com/artifact/org.primefaces/primefaces -->
  <dependency>
    <groupId>org.primefaces</groupId>
    <artifactId>primefaces</artifactId>
    <version>11.0.0</version>
  </dependency>
</dependencies>
```

Indicaciones Técnicas para el desarrollo de “Hablando con las manos”

- PgAdmin

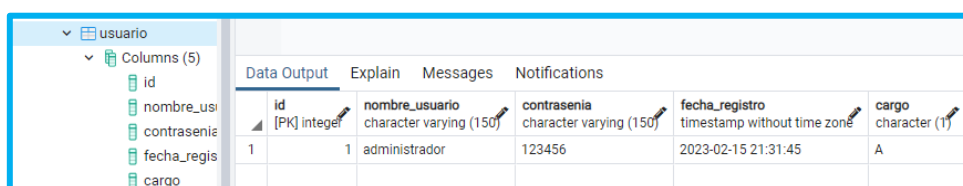
Para la creación de la Aplicación Web “Hablando con las manos”, lo primero que se realizó fue la creación de la base de datos donde se especificó las tablas con sus respectivas columnas, esto con el fin de guardar la información de una forma más ordenada. Las tablas creadas son las siguientes:



The screenshot shows the 'senias_palabras' table in PgAdmin. The table has 5 columns: id (integer, PK), nombre (text), imagen (text), fecha_registro (timestamp without time zone), and id_usuario (integer). The data output shows one row with id=1, nombre='adios', imagen='adios.png', fecha_registro='2023-02-15 21:32:05', and id_usuario=1.

	id [PK] integer	nombre text	imagen text	fecha_registro timestamp without time zone	id_usuario integer
1	1	adios	adios.png	2023-02-15 21:32:05	1

La tabla “senias_palabras” creo con el objetivo de guardar todo el vocabulario en lenguaje de señas, vale recalcar que en esta solo se guardará el nombre del archivo a consultar, es decir, solo habrá una cadena de texto más no un archivo digital con algún formato específico.



The screenshot shows the 'usuario' table in PgAdmin. The table has 5 columns: id (integer, PK), nombre_usuario (character varying (150)), contrasenia (character varying (150)), fecha_registro (timestamp without time zone), and cargo (character (1)). The data output shows one row with id=1, nombre_usuario='administrador', contrasenia='123456', fecha_registro='2023-02-15 21:31:45', and cargo='A'.

	id [PK] integer	nombre_usuario character varying (150)	contrasenia character varying (150)	fecha_registro timestamp without time zone	cargo character (1)
1	1	administrador	123456	2023-02-15 21:31:45	A

En este traductor un individuo puedo añadir más información expandir el vocabulario de lenguaje de señas, este será llamado como “Administrador” el cual tendrá un acceso privilegiado para realizar la acción antes mencionada. Por aquello, la tabla “usuario” es la que se encarga de tener las credenciales para el acceso privado del Administrador, en otras palabras, el objetivo es poder realizar un “Log In” en donde pueda verificar las credenciales antes mencionadas.

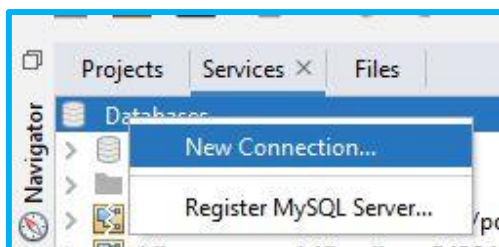
- NetBeans

NetBeans es el programa en donde llevaremos a cabo la mayor parte del proyecto, por ende, lo primero a realizar es la conexión con la base de datos y para ello se necesita configurar el “JDBC Connection Pools” junto con el “JDBC Resources”, este apartado se consigue al iniciar el Payara Server y abrir “View Domain Admin Console”.

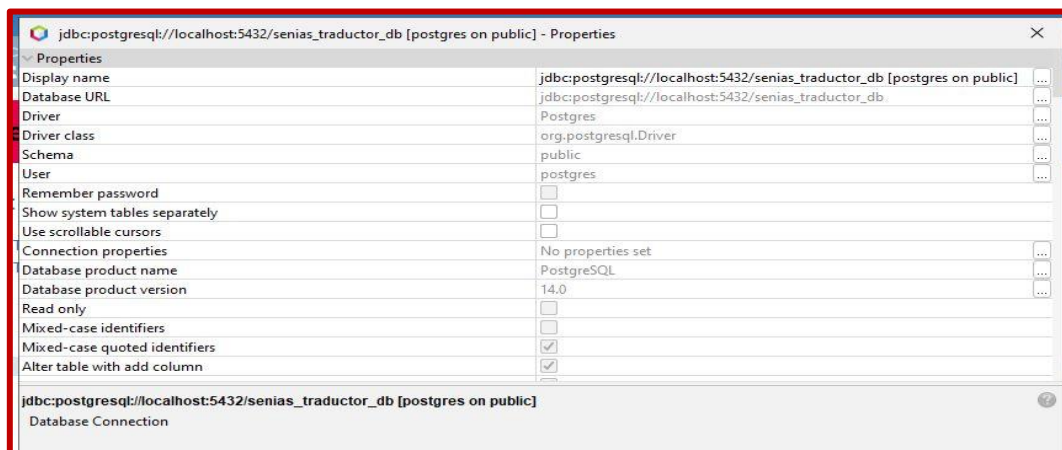
Esto es importante ya que el JDBC Connection Pool es una técnica utilizada en la programación Java para mejorar el rendimiento y la escalabilidad de las aplicaciones que acceden a bases de datos a través del JDBC (Java Database Connectivity).

Por otra parte, Los JDBC Resources son una característica de Java Enterprise Edition (Java EE) que se utiliza para administrar y configurar los recursos JDBC, el cual, es una conexión a una base de datos que se utiliza para ejecutar consultas y actualizar datos. Los recursos JDBC permiten a las aplicaciones Java EE acceder y manipular datos en una base de datos.

Antes de realizar esta configuración se recomienda ubicarse en “Services” y realizar en el “Database” una “New Connection”.



Esta son las propiedades de nuestra conexión realizada.



Una vez realizada la conexión del “Database”, es hora de configurar el JDBC antes mencionado, por ende las configuraciones global que se mantiene en este proyecto son las que se muestran en las siguientes ilustraciones:

General Settings

Pool Name: seniaspool

Resource Type: javax.sql.DataSource
Must be specified if the datasource class implements more than 1 of the interface.

Datasource Classname: org.postgresql.ds.PGSimpleDataSource
Vendor-specific classname that implements the DataSource and/or XADataSource APIs

Driver Classname:
Vendor-specific classname that implements the java.sql.Driver interface.

Ping: ☒ Enabled
When enabled, the pool is pinged during creation or reconfiguration to identify and warn of any erroneous values for its attributes

Deployment Order: 100
Specifies the loading order of the resource at server startup. Lower numbers are loaded first.

Description:

Las propiedades adicionales que se ingresó en el JDBC Connection Pool, fueron en los siguientes apartados mostrados en la siguiente tabla;

<i>Name</i>	<i>Value</i>
DatabaseName	senias_traductor_db
PortNumber	5432
Password	contraseña personalizada*
Url	jdbc:postgresql://localhost:5432/senias_traductor_db
URL	jdbc:postgresql://localhost:5432/senias_traductor_db
User	postgres

Edit JDBC Connection Pool Properties Save Cancel

Modify properties of an existing JDBC connection pool.

Pool Name: seniaspool

Additional Properties (52)

g+ Add Property Delete Properties

Select	Name	Value	Description
<input type="checkbox"/>	AdaptiveFetch	false	
<input type="checkbox"/>	AdaptiveFetchMaximum	-1	
<input type="checkbox"/>	AdaptiveFetchMinimum	0	

Una vez realizado la configuración del JDBC Connection Pool, ahora es necesario la crear un nuevo JDBC Resources para el control de los recursos de este. La configuración de este proyecto fue el siguiente.

Edit JDBC Resource

Edit an existing JDBC data source.

[Load Defaults](#)

JNDI Name: senias_traductor

Pool Name: seniaspool [Use the JDBC Connection Pools page to create new pools](#)

Deployment Order: 100
Specifies the loading order of the resource at server startup. Lower numbers are loaded first.

Description:

Status: ☒ Enabled

Un punto esencial para terminar con la conexión de forma eficiente y satisfactoria es la configuración del “persistence.xml”, el cual se hizo de acuerdo con las configuraciones realizadas anteriormente.

tt.app.seniastraductor

General:

Persistence Unit Name: tt.app.seniastraductor

Persistence Provider: EclipseLink (JPA 2.1)(default)

Data Source: senias_traductor

☒ Use Java Transaction APIs

Table Generation Strategy: ☐ Create ☐ Drop and Create ☒ None

Validation Strategy: ☒ Auto ☐ Callback ☐ None

Shared Cache Mode: ☐ All ☐ None ☐ Enable Selective ☐ Disable Selective ☒ Unspecified

☒ Include All Entity Classes in "SeniasTraductor-1.0" Module

Archivos .xhtml

En el presente proyecto se realizó el encabezado y el menú en secciones diferentes, haciendo que en el dashboard.xhtml sea donde se una los apartados mencionados con anticipación. (Se subraya la parte donde se hace la unión del encabezado.xhtml y menu.xhtml)

Dashboard

```

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://java.sun.com/jsf/html"
      xmlns:f="http://java.sun.com/jsf/core"
      xmlns:ui="http://java.sun.com/jsf/facelets"
      xmlns:p="http://primefaces.org/ui">
  <h:head>
    <f:facet name="first">
      <meta http-equiv="X-UA-Compatible" content="IE=edge" />
      <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
      <meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-
scale=1.0, user-scalable=0"/>
      <meta name="apple-mobile-web-app-capable" content="yes" />
    </f:facet>
    <title><ui:insert name="title">Inicio</ui:insert></title>
  </h:head>
  <h:body styleClass="main-body ui-input-filled">
    <div class="layout-wrapper layout-topbar-light layout-static-active layout-menu-
light layout-static layout-static-active">
      <ui:include src="encabezado.xhtml" />
      <ui:include src="menu.xhtml" />

      <div class="layout-main">
        <ui:include src="breadcrumb.xhtml" />
        <div class="layout-content">
          <ui:insert name="content"/>
        </div>
      </div>
    </div>

    <h:outputStylesheet name="componentes.css" library="css" />
    <h:outputStylesheet name="matriz.css" library="css" />
    <h:outputStylesheet name="_list.scss" library="css" />
  </h:body>
</html>

```

Encabezado

```

<ui:composition
  xmlns="http://www.w3.org/1999/xhtml"
  xmlns:f="http://java.sun.com/jsf/core"
  xmlns:h="http://java.sun.com/jsf/html"
  xmlns:ui="http://java.sun.com/jsf/facelets"
  xmlns:p="http://primefaces.org/ui">

  <div class="layout-topbar">
    <div class="layout-topbar-wrapper">
      <div class="layout-topbar-left">
        <h:link id="logolink" outcome="dashboard" styleClass="layout-topbar-logo">
          <!--<p:graphicImage name="images/#{ guestPreferences.topbarTheme ==
'light' ? 'logo_faenamiento.png' : 'logo-poseidon-dark.png'}" library="poseidon-layout"
width="100%" height="100%"/>-->
          <p:graphicImage name="icono.png" library="images" width="100%"
height="100%"/>
        </h:link>
      </div>
      <div class="layout-topbar-right">

        </div>
      </div>
    </div>
  </ui:composition>

```

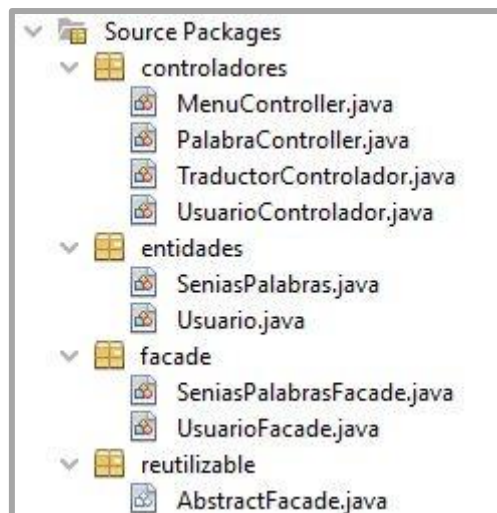

Menú

```
<ui:composition
    xmlns="http://www.w3.org/1999/xhtml"
    xmlns:f="http://java.sun.com/jsf/core"
    xmlns:h="http://java.sun.com/jsf/html"
    xmlns:ui="http://java.sun.com/jsf/facelets"
    xmlns:p="http://primefaces.org/ui">

    <div class="menu-wrapper">
        <div class="layout-menu-container">
            <h:form id="menuform">
                <p:menu widgetVar="PoseidonMenuWidget" style="border: none; width: 100%">
                    <p:menuitem id="traductor" value="Traductor" outcome="traductor"/>
                    <p:menuitem id="nuevapalabra" value="Agregar palabra"
outcome="seniasPalabras" rendered="#{menuController.userConsulta.cargo eq 'A'}/>
                    <p:menuitem id="cerrarsesion" value="Cerrar sesion"
action="#{usuarioControlador.cerrarSesion_()}" />
                </p:menu>
            </h:form>
        </div>
    </div>
</ui:composition>
```

Vista de la interfaz de la Aplicación Web

Sourcer Packages



En este apartado podemos encontrar los controladores necesario para realizar las diferentes consultas realizadas dentro del traductor o el ingreso del administrador, si bien es cierto, este código es generado automáticamente no indica que no se pueda modificar, por ello se realizo cambios que beneficia al objetivo que se desea lograr al realizar una consulta.

En los paquetes controlador se encuentra los archivos;

- ✚ MenuController.java
- ✚ PalabraController.java
- ✚ TraductorControlador.java
- ✚ UsuarioControlador.java

Los cuales a continuación se presenta la utilidad de cada controlador y su comportamiento:

MenuController: El controlador MenuController se encarga hacer las consultas necesarias para llevar el control del menú que se encuentra ubicado en la parte izquierda de la interfaz gráfica del proyecto, es decir, si el administrador comienza la sesión este tendré la opción “Nueva Palabra”, el cual internamente hace el llamado a dicho controlador.

```

16  * @author pc
17  */
18  @Named(value = "menuController")
19  @ViewScoped
20  public class MenuController implements Serializable {
21
22      private Usuario userConsulta;
23
24      /**
25       * Creates a new instance of MenuController
26       */
27      public MenuController() { }
28
29      @PostConstruct
30      public void init() {
31          this.userConsulta = (Usuario) FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get("usuario");
32          System.out.println(this.userConsulta);
33      }
34
35      public void validaAcceso () {
36
37      }
38
39      public Usuario getUserConsulta() {
40          return userConsulta;
41      }
42
43      public void setUserConsulta(Usuario userConsulta) {
44          this.userConsulta = userConsulta;
45      }
46

```

```

1  <?xml version="1.0" encoding="UTF-8" ?>
2  <xhtml xmlns="http://www.w3.org/1999/xhtml"
3  xmlns:jsf="http://java.sun.com/jsf/core"
4  xmlns:html="http://java.sun.com/jsf/html"
5  xmlns:ui="http://primefaces.org/ui" ?>
6
7
8  <h:menuWrapper ?>
9      <h:layoutMenuContainer ?>
10         <h:form id="menuForm" ?>
11             <p:menu widgetVar="PoseidonMenuWidget" style="border: none; width: 100%" ?>
12                 <p:menumitem id="traductor" value="Traductor" outcome="traductor" />
13                 <p:menumitem id="nuevapalabra" value="Agregar palabra" outcome="seniasPalabras" rendered="#{menuController.userConsulta}" />
14                 <p:menumitem id="cerrarsesion" value="Cerrar sesion" action="#{usuarioControlador.cerrarSesion}" />
15             </p:menu>
16         </h:form>
17     </h:layoutMenuContainer>
18 </h:menuWrapper>
19
20

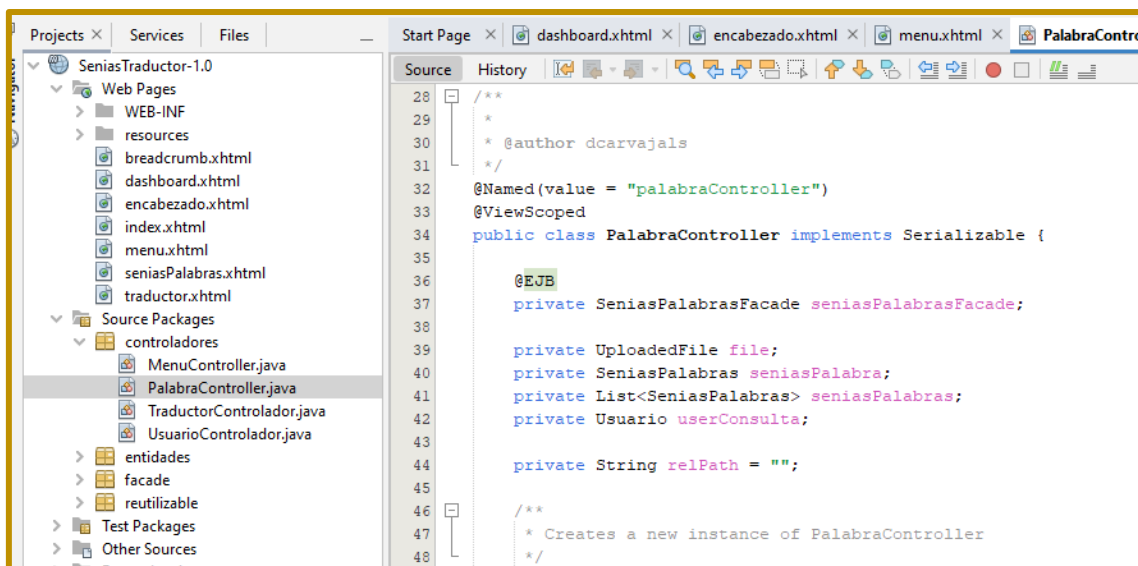
```

Sección menú, opción “Agregar palabra”, se hace el llamado del controlador “MenuController”

Para ello, es necesario centrarnos en las vistas y de esta manera mencionar en qué momento hacen el llamado de dichos controladores.

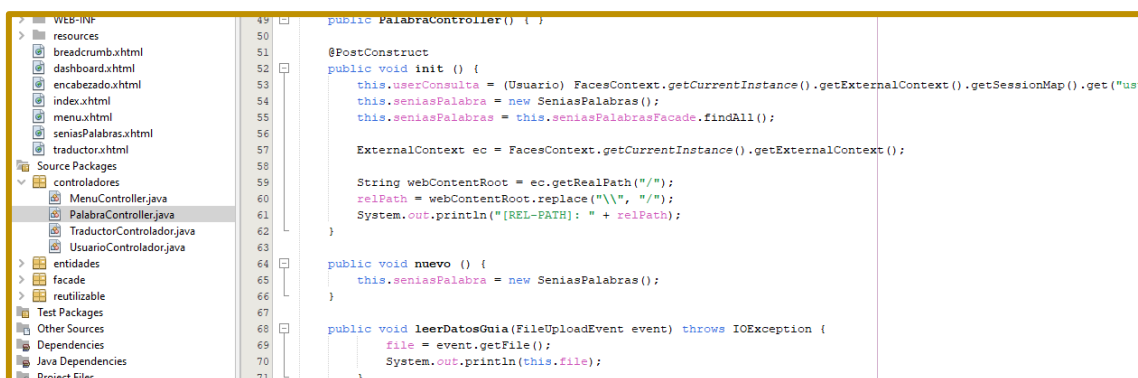
PalabraController: La clase “PalabraController” se encarga de la interacción con la vista y del procesamiento de las acciones realizadas por el usuario.

La clase utiliza varias anotaciones, como *@Named* y *@ViewScoped*, para indicar que se trata de un bean de Java Server Faces (JSF) y especificar el alcance del objeto.



El controlador tiene varios atributos, entre ellos *seniasPalabrasFacade*, que es un objeto de la clase *SeniasPalabrasFacade* que actúa como intermediario entre la base de datos y el controlador. También tiene una lista *seniasPalabras* que contiene todas las palabras en la base de datos y un objeto *seniasPalabra* que contiene los detalles de una sola palabra.

El controlador tiene varios métodos, entre ellos *init()*, que se ejecuta después de que se construye el objeto y se encarga de obtener el usuario de la sesión actual, la lista de palabras de la base de datos y el directorio relativo donde se guardarán las imágenes.



El método *leerDatosGuia(FileUploadEvent event)* se llama cuando se carga una imagen y lee los datos de la imagen. El método *guardarCambios()* guarda los cambios realizados en la base de datos y en el servidor de archivos, mientras que *eliminar()* elimina una palabra de la base de datos.

```

72
73
74 public void guardarCambios () {
75     System.out.println("guardarCambios()");
76     String path = this.relPath + "\\src\\main\\webapp\\resources\\images\\senias";
77     String path2 = this.relPath + "\\resources\\images\\senias";
78
79     //String path = "D:\\app_web\\SeniasTraductor\\src\\main\\webapp\\resources\\images\\senias";
80     //String path2 = "D:\\app_web\\SeniasTraductor\\target\\SeniasTraductor-1.0\\resources\\images\\senias";
81     System.out.println(this.file);
82     this.seniasPalabra.setNombre(this.seniasPalabra.getNombre());
83     int pos = this.file.getFileName().lastIndexOf(".");
84     String ext = this.file.getFileName().substring(pos + 1, this.file.getFileName().length());
85     System.out.println(ext);
86     try {
87         this.seniasPalabra.setNombre(this.seniasPalabra.getNombre().replaceAll(" ", "").toLowerCase());
88         this.seniasPalabra.setFechaRegistro(new Date());
89         this.seniasPalabra.setImagen(this.seniasPalabra.getNombre().replaceAll(" ", "").toLowerCase() + ext);
90         this.seniasPalabra.setIdUsuario(this.usuarioConsultado);
91         this.seniasPalabrasFacade.create(this.seniasPalabra);
92         if(this.file.getSize() > 0) {
93             crearImagen( IOUtils.toByteArray(this.file.getInputStream()), path, this.seniasPalabra.getNombre().replaceAll(" ", "").toLowerCase() + ext);
94             crearImagen( IOUtils.toByteArray(this.file.getInputStream()), path2, this.seniasPalabra.getNombre().replaceAll(" ", "").toLowerCase() + ext);
95             this.seniasPalabra = null;
96             System.out.println("IMAGEN CREADA");
97             this.seniasPalabras = this.seniasPalabrasFacade.findAll();
98             PrimeFaces.current().executeScript("alert('Imagen creada exitosamente.')");
99             PrimeFaces.current().ajax().update("frmListSeniaPalabras:dtSeniaPalabra");
100         }
101     } catch (IOException ex) {System.out.println("ERROR:" + ex.getMessage());}

```

```

101
102
103 }
104
105 public void eliminar () {
106     System.out.println(this.seniasPalabra.getNombre());
107     this.seniasPalabrasFacade.remove(this.seniasPalabra);
108     this.seniasPalabras = this.seniasPalabrasFacade.findAll();
109     PrimeFaces.current().ajax().update("frmListSeniaPalabras:dtSeniaPalabra");
110 }

```

El método *crearImagen(byte[] bytes, String rutaFolder, String nombreImg)* crea una imagen a partir de un arreglo de bytes y la guarda en el servidor de archivos.

```

110
111
112 }
113
114 public void crearImagen (byte[] bytes, String rutaFolder, String nombreImg) {
115     try {
116         Path path = Paths.get(rutaFolder, nombreImg);
117         Files.write(path, bytes);
118     } catch (IOException ex) {
119         System.out.println(ex.getMessage());
120     }

```

TraductorController: En esta parte del código el controlador utiliza la tecnología JSF (JavaServer Faces) y sigue el patrón de diseño MVC (Model-View-Controller) para realizar una traducción de texto a lenguaje de señas.

La clase se encarga de recibir el texto que el usuario ingresa en la vista y, mediante el método *"traducir"*, divide el texto en palabras, busca en una base de datos la señal correspondiente a cada palabra y las guarda en una lista de *"SeniasPalabras"*.

```

21
22
23 */
24 @Named(value = "traductorControlador")
25 @ViewScoped
26 public class TraductorControlador implements Serializable {
27
28     @EJB
29     private SeniasPalabrasFacade seniasPalabrasFacade;
30
31     private String textoATraducir = "";
32     private List<SeniasPalabras> seniasPalabras;
33     private SeniasPalabras seniaPalabra;

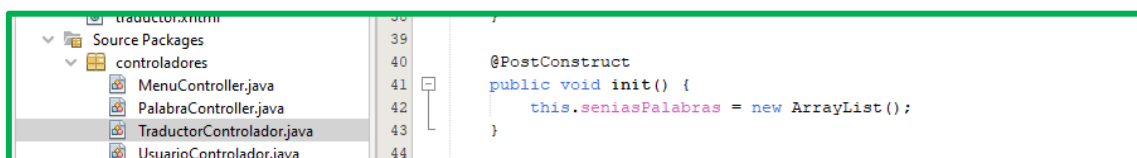
```

La clase "SeniasPalabras" es una entidad JPA (Java Persistence API) que representa las señas y sus palabras correspondientes en la base de datos. La anotación *"@EJB"* permite

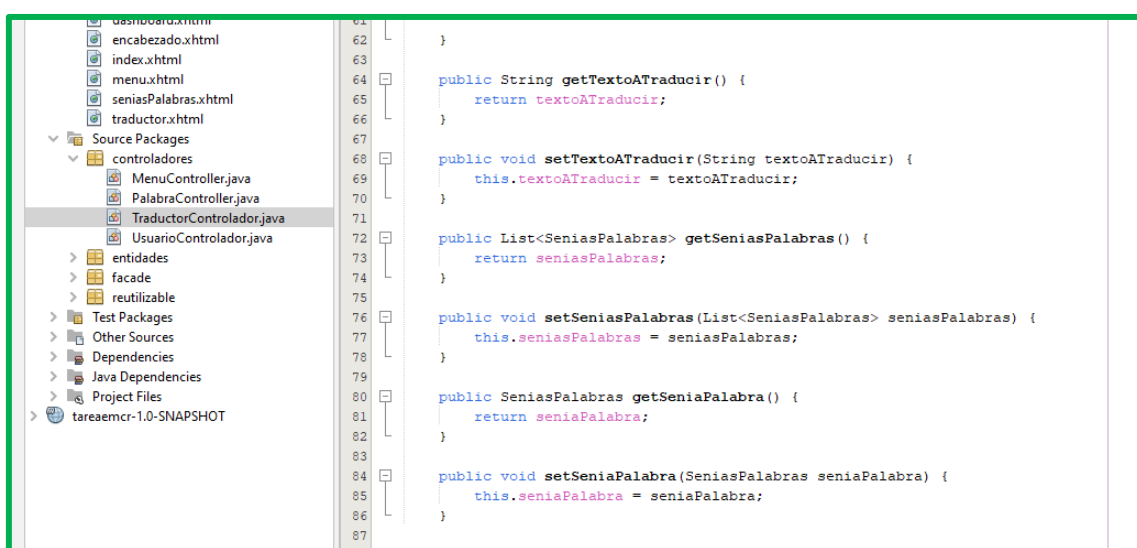
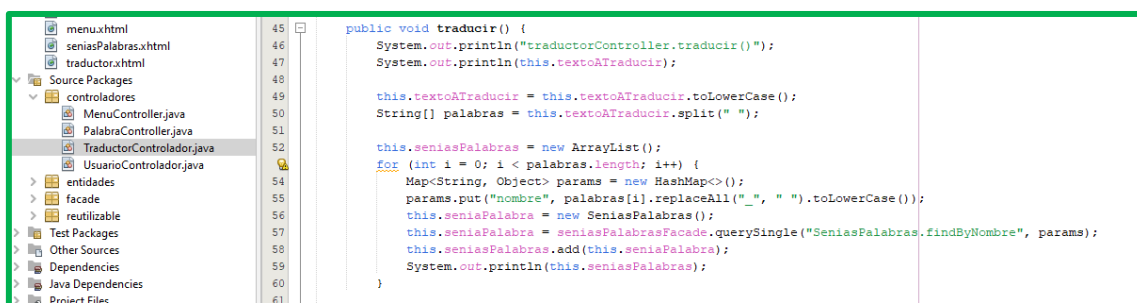
la inyección de dependencias de un EJB (Enterprise JavaBean) que se encarga de acceder a la base de datos.

La anotación "*@Named*" es parte de la especificación de JSF y permite que el controlador sea referenciado por un nombre en la vista, mientras que "*@ViewScoped*" es otra anotación de JSF que indica que el controlador es válido para una vista específica.

El método "*@PostConstruct*" es un método de inicialización que se ejecuta después de que se haya creado una instancia del controlador y se utiliza para inicializar la lista de "*SeniasPalabras*".



Además, se tienen propiedades privadas como "*textoATraducir*", "*seniasPalabras*" y "*seniaPalabra*", que son utilizadas para almacenar el texto a traducir, la lista de señas y palabras correspondientes, y la señal correspondiente a una palabra específica respectivamente. Estas propiedades se acceden mediante los métodos públicos "*getTextoATraducir*", "*getSeniasPalabras*", "*setSeniasPalabras*", "*getSeniaPalabra*" y "*setSeniaPalabra*".



UsuarioController:

El objetivo principal del controlador es gestionar la autenticación y la sesión de usuario en la aplicación.

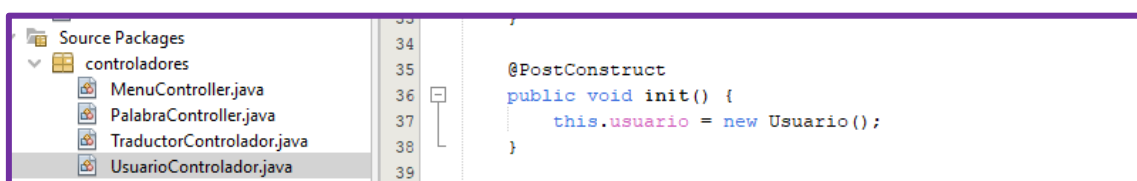
El controlador está definido en el paquete "controladores" y utiliza las clases Usuario y UsuarioFacade definidas en los paquetes "entidades" y "facade" respectivamente. También importa varias clases del framework JSF y utiliza la anotación *@Named* para que pueda ser inyectado en otras partes de la aplicación.



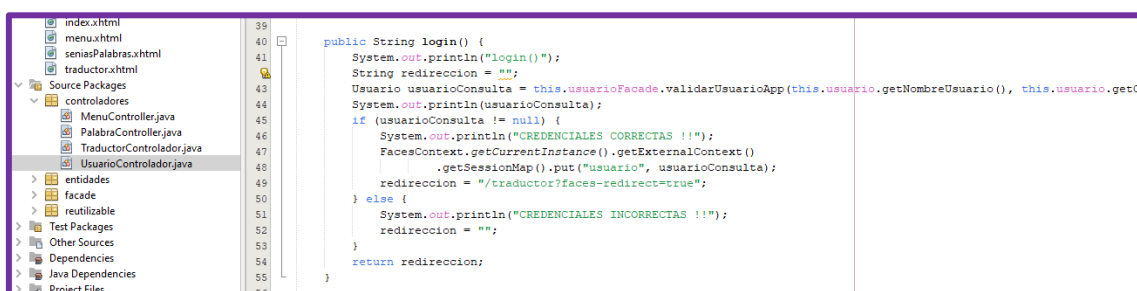
La clase *UsuarioControlador* implementa la interfaz *Serializable* y utiliza la anotación *@ViewScoped* para indicar que la instancia del controlador estará disponible durante toda la duración de la vista en la que se utiliza.

La clase tiene un atributo de tipo *UsuarioFacade* que se inyecta mediante la anotación *@EJB*. El controlador también tiene un atributo usuario de tipo Usuario que se utiliza para almacenar la información del usuario que intenta iniciar sesión en la aplicación.

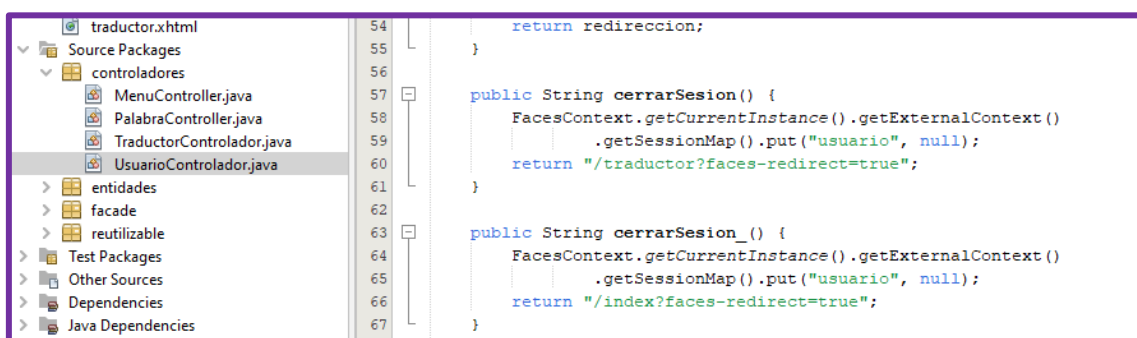
El método *init()* es anotado con *@PostConstruct* y se utiliza para inicializar el objeto usuario en el constructor.



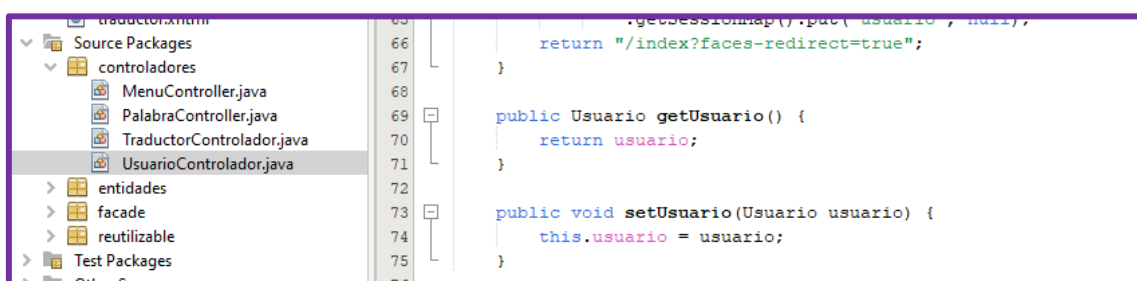
El método *login()* es utilizado para validar las credenciales del usuario y si son correctas, se almacena el objeto Usuario en la sesión del usuario utilizando la clase *FacesContext*. Si las credenciales son incorrectas, se establece una cadena vacía como redireccionamiento.



El método *"cerrarSesion"* se encarga de cerrar la sesión actual del usuario, eliminando la información del usuario almacenada en la sesión. Después de cerrar la sesión, el método redirige al usuario a la vista *"traductor"*.



Por otro lado, el método *"cerrarSesion_"* realiza la misma tarea que el anterior, pero redirige al usuario a la vista *"index"* en lugar de a *"traductor"*.



Por último, los métodos *"getUsuario"* y *"setUsuario"* son métodos de acceso para la propiedad *"usuario"* de la clase *"UsuarioControlador"*. El método *"getUsuario"* devuelve el valor de la propiedad *"usuario"*, mientras que *"setUsuario"* establece el valor de la propiedad *"usuario"* con el valor proporcionado como parámetro. Estos métodos son necesarios para poder acceder y modificar la información de usuario almacenada en la instancia de la clase *"UsuarioControlador"*.