­



Jonatan Carrera Viera – 2º DAW Presencial Mañana

Cómo realizar un traductor de voz en tiempo real **usando Python** de español a japonés, con la posibilidad de seleccionar qué tipo de voz quieres usar en japonés para una personalización completa en un universo virtual.

**Traductor en Python**

**TRADUCTOR DE VOZ EN TIEMPO REAL**

**Espones**

A

Anexos 22

B

Bibliografía 21

C

Conclusiones 19

D

Desarrollo 15

Diseño del Sistema 13

E

Elicitación de Requisitos y Análisis de Riesgos 10

Estado del arte 7

I

Implementación 14

Introducción y Motivación 3

O

Objetivos del proyecto 6

P

Planificación 17

Planificación financiera 18

Pruebas del sistema 16

T

Trabajo futuro 20

# Introducción y Motivación

La utilización de avatares personalizados en lugar de mostrar el aspecto real de las personas en la sociedad ha sido una tendencia cada vez más común en la era digital, especialmente en Japón, donde existen empresas que se dedican exclusivamente a la promoción de estos avatares como forma de entretenimiento. Existen una gran cantidad de motivos por los cuales las personas optan por utilizar avatares en línea en lugar de mostrar su aspecto real. A continuación, entraremos más en profundidad en esto y alguna de las motivaciones para el uso de dichos avatares para mostrarse en el mundo virtual.

En la actualidad, la sociedad ha experimentado un cambio significativo en la forma en que interactuamos y nos presentamos en línea. El uso de avatares personalizados se ha convertido en una práctica común y ampliamente aceptada en plataformas digitales como redes sociales (*Twitch*, *Facebook*, *Youtube* e incluso *Microsoft Teams* recientemente), juegos en línea (juegos de rol y juegos de rol multijugador masivo) y comunidades virtuales (*VRChat* y *Metaverso* entre otros). Estos avatares, que pueden ser representaciones gráficas o digitales de uno mismo, permiten a las personas crear una identidad en línea que difiere de su aspecto físico real.

Una de las principales razones detrás de esta preferencia es la búsqueda de privacidad y protección. La era digital ha llevado consigo la preocupación por la seguridad y la protección de la información personal. Al utilizar un avatar en lugar de mostrar su apariencia real, las personas pueden salvaguardar su privacidad y evitar exponerse a riesgos potenciales, como el robo de identidad, poder salir a la calle con la seguridad de no ser reconocido o el acoso en línea. Los avatares les permiten interactuar en línea de manera más segura y controlar la información que desean compartir sobre sí mismos.

Además de la seguridad, los avatares ofrecen una oportunidad para expresarse creativamente y explorar identidades alternativas. La posibilidad de personalizar un avatar brinda a las personas la libertad de representarse de la manera que deseen, sin las restricciones impuestas por su apariencia física. Esto puede ser especialmente atractivo para aquellos que se sienten inseguros o insatisfechos con su aspecto real, ya que pueden crear una imagen idealizada o aspiracional de sí mismos en el mundo virtual. O simplemente verse como un dinosaurio parlanchín de colores chillones.

Asimismo, los avatares fomentan la diversidad y la inclusión. Al utilizar avatares, las personas pueden representar su identidad de género, etnia, orientación sexual u otras características importantes para ellas, incluso si no se sienten cómodas compartiéndolas en el mundo real. Esto promueve un entorno en línea más inclusivo y elimina las barreras físicas o estereotipos asociados con la apariencia física.

En resumen, la utilización de avatares personalizados en lugar del aspecto real en línea se ha convertido en una práctica común en la sociedad actual. La privacidad, la seguridad, la expresión creativa y la promoción de la diversidad son algunos de los motivos por los que las personas optan por utilizar avatares. Esta tendencia ha transformado la forma en que nos presentamos y nos relacionamos en el entorno digital, permitiéndonos crear identidades virtuales que reflejen nuestra individualidad y preferencias personales. Siendo el máximo exponente la cultura japonesa, que ha conseguido crear todo un negocio alrededor de esta idea.

En este aspecto es dónde ha surgido la idea de que, si podemos llegar a personalizar nuestra imagen en línea, ¿por qué no ir un paso más allá y personalizar también nuestra voz? Ya existen herramientas que permiten verbalizar de manera digital lo que se recibe de manera escrita, pero tienen pocas posibilidades de personalización o, directamente, suenan demasiado robóticas.

En mi caso, no dispongo de un estudio de grabación ni de los conocimientos audiovisuales para producir un banco de voces. También existen *IA* para la imitación de voces de famosos, pero es preferible evitar que la voz se parezca a la de alguien famoso y no incurrir en delitos de suplantación de identidad o alguna demanda por posible copyright (como apunta que en un futuro ocurra). Así que, tras investigar un poco sobre el tema, se ha encontrado una aplicación gratuita que cumple los criterios que se buscaban: variedad de voces, posibilidad de personalización de las voces y adaptada al gran público objetivo actual de este tipo de avatares, los frikis.

Esta aplicación viene encapsulada en un contenedor de *Docker* y una *API* integrada para su interacción desde el propio PC, por lo que facilitará en gran medida el desarrollo de la aplicación que se ha planteado.

A parte de lo ya mencionado, se pretende realizar dicha aplicación en el lenguaje de *Python*, ya que este no ha podido ser aprendido durante el curso lectivo, pero es uno de los lenguajes de programación más extendidos actualmente en el mercado laboral, por lo que se usa como herramienta de aprendizaje para la realización del proyecto.

# Objetivos del proyecto

Lo que se pretende con este proyecto son los siguientes puntos:

* Aprender un nuevo lenguaje de programación puntero.
* La realización de un programa que pudiera ser usado realmente por alguna persona.
* Afianzar conocimientos de Docker.
* Creación de un programa que interactúe con varias APIs, para control de tiempos de ejecución.
* Obtener dichos conocimientos mientras se realiza una tarea que resulte atractiva.

Junto a los motivos expuestos en los puntos anteriores, también se contempla que se pueda llegar a usar por otros usuarios. Para ello se ha revisado *Tkinter* de *Python* para la realización de una interfaz gráfica y así hacerlo más accesible para todo tipo de usuarios, pero es algo que se verá a lo largo del desarrollo cómo de factible podría ser.

# Estado del arte

En los últimos años, el desarrollo de traductores automáticos ha experimentado avances significativos, permitiendo la comunicación efectiva entre personas que hablan diferentes idiomas. Entre las numerosas combinaciones lingüísticas, la traducción del español al japonés ha sido objeto de creciente interés debido a la importancia de la comunicación entre hablantes de ambos idiomas, pues existe un mercado bastante incipiente en productos audiovisuales de temática japonesa para comunidades hispanohablantes.

En la actualidad, existen diversas soluciones de traducción en tiempo real que abordan esta necesidad. Sin embargo, un aspecto clave que ha cobrado relevancia es la capacidad de personalizar las voces utilizadas en la traducción, con el fin de adaptar la experiencia a las preferencias y necesidades individuales. Pero no existe en la actualidad ninguna aplicación que pueda realizar dicha traducción y emisión del texto traducido de manera rápida con una voz que pueda ser personalizada más allá de un par de voces para cada sexo fijas, sin poder modificar nada de las mismas.

En términos de traducción automática español-japonés, los sistemas basados en redes neuronales han demostrado ser altamente efectivos y han superado a los enfoques anteriores. Estos sistemas utilizan algoritmos de aprendizaje profundo para procesar grandes cantidades de datos paralelos, como frases o textos alineados en ambos idiomas. Mediante la extracción de patrones y reglas lingüísticas, estas redes neuronales son capaces de generar traducciones coherentes y de alta calidad.

En cuanto a la personalización de voces, los avances en síntesis de voz y tecnologías de procesamiento de audio han permitido la creación de voces sintéticas más naturales y expresivas. La síntesis de voz basada en unidades o la generación de habla con redes neuronales convolucionales han mejorado significativamente la calidad y la fluidez de las voces generadas por los sistemas de traducción.

Además, se ha investigado la adaptación de las voces sintéticas a través del aprendizaje automático. Los modelos de voz personalizados, entrenados con grabaciones de habla de hablantes nativos, pueden proporcionar una experiencia de traducción más inmersiva y familiar para los usuarios. Estos modelos permiten ajustar el tono, la entonación y el estilo de las voces utilizadas en la traducción, lo que contribuye a una mayor comprensión y aceptación por parte de los usuarios.

Es importante destacar que la disponibilidad de grandes cantidades de datos paralelos español-japonés ha impulsado el rendimiento y la precisión de los sistemas de traducción automática. La creación y el acceso a corpus multilingües han sido fundamentales para el desarrollo y el entrenamiento de modelos de traducción efectivos y adaptables.

Es por todo esto que el estado del arte en el campo de los traductores a tiempo real de español a japonés con voces personalizables ha experimentado avances significativos. Los sistemas basados en redes neuronales han mejorado la calidad de las traducciones, mientras que los avances en síntesis de voz y tecnologías de procesamiento de audio han permitido la personalización de las voces utilizadas en la traducción. El aprendizaje automático y el acceso a grandes conjuntos de datos paralelos español-japonés han sido fundamentales para estos avances, ofreciendo nuevas posibilidades para una comunicación fluida y adaptada a las preferencias individuales en el ámbito de la traducción automática, pero aún no hay ningún software que haga de todo esto una realidad.

Es por ello que con este proyecto se intente aportar un granito de arena a todo este proceso de avance y, puede que sirva de inspiración para que la tecnología de otro pequeño paso hacia adelante en mejorar la comunicación a nivel global entre usuarios de distinto habla, pudiendo ofrecerles un nivel de personalización suficiente con el que puedan expresarse de una manera divertida, cómoda e inclusiva.

# Elicitación de Requisitos y Análisis de Riesgos

Para poder llevar a cabo este proyecto se ha revisado todo lo que hay actualmente en el mercado de traducción y bancos de voces personalizables para poder escoger y ajustar las distintas necesidades a cubrir con sus posibles riesgos, siempre intentando mantener el uso de *APIs* gratuitas para evitar costes en el desarrollo y hacerlo accesible a la mayor cantidad de usuarios posible. Por lo que, a continuación, se exponen los requisitos y riesgos que se han podido encontrar a la hora de la plantear este proyecto.

## Elicitación de requisitos:

1. Identificar las funcionalidades clave del traductor:

* Traducción automática en tiempo real de español a japonés.
* Interfaz de usuario intuitiva y fácil de usar.
* Personalización de voces para ofrecer opciones de voz en japonés.
* Soporte para diferentes formatos de entrada de voz.
* Resultados de traducción precisos y coherentes.

1. Determinar los requisitos técnicos:

* Desarrollo en el lenguaje de programación *Python*.
* Utilización de bibliotecas o *APIs* de traducción, como la *API* de *Google Translate*, *Microsoft Translator* y *OpenAI*.
* Integración con tecnologías de síntesis de voz para la personalización de voces en japonés.

1. Considerar requisitos de rendimiento y escalabilidad:

* Tiempo de respuesta rápido para traducciones en tiempo real.
* Capacidad para manejar una carga de trabajo alta y escalabilidad para soportar múltiples usuarios simultáneos.
* Eficiencia en el uso de recursos computacionales.

1. Determinar requisitos de calidad:

* Traducciones precisas y coherentes con la gramática y el contexto adecuados.
* Minimización de errores y garantía de la calidad de las traducciones generadas.
* Adaptación al lenguaje y estilo de los usuarios.

## Análisis de riesgos:

1. Rendimiento insatisfactorio debido a la carga de trabajo alta.

* Mitigación: Optimización del código y uso eficiente de algoritmos para garantizar un rendimiento óptimo y manejo de cargas de trabajo elevadas.

1. Inexactitudes o errores en las traducciones generadas.

* Mitigación: Realización de pruebas exhaustivas y validación de resultados utilizando muestras de texto y revisión manual de traducciones.

1. Limitaciones en la personalización de las voces en japonés.

* Mitigación: Investigación y selección de tecnologías de síntesis de voz que permitan una amplia personalización y expresividad en el idioma japonés.

1. Dependencia de servicios de traducción y síntesis de voz de terceros.

* Mitigación: Evaluación y selección de proveedores confiables y gratuitos para garantizar la disponibilidad y calidad de los servicios externos.

1. Complejidad técnica y falta de experiencia en el desarrollo en Python.

* Mitigación: Asignación de un equipo de desarrollo con experiencia en Python, o adquisición de conocimientos y habilidades necesarias a través de capacitación o colaboración con expertos en el lenguaje.

# Diseño del Sistema

Tras obtener los requisitos del se opta por el siguiente diseño del sistema:

1. Arquitectura del sistema
   * Uso de cuatro (4) métodos separados más el principal para la ejecución del programa. Divididos en:
     + Método principal.
     + Método de captura de voz.
     + Método de envío a la *API Speech-to-Text* *Whisper* 02 de *OpenAI*.
     + Método de envío a la *API Traductora* de *OpenAI*.
     + Método de envío a la *API Sintetizadora* de *Voicevox* instalada en *Docker*.
     + Método para la reproducción del audio generado al sistema.
2. Docker
   * Contenedor *Docker* para el alojamiento de la *API* de *Voicevox*.
3. Componentes y métodos
   * Componente principal: En este método se realizarán las llamadas a los otros componentes de la aplicación, quedando siempre abierto a la espera de la activación por tecla para iniciar toda la ejecución y otra combinación para la detención del programa.
   * Componente de captura de voz: En este apartado se realizará la captura del sonido por micrófono durante un tiempo determinado, guardándolo en un archivo que será siempre sobre escrito para evitar un consumo de memoria de almacenamiento excesivo.
   * Componente de envío a *Speech-to-Text Whisper* 02: Componente en el que se enviará el archivo de audio generado para su procesamiento a texto mediante autenticación de *key* en *OpenAI* y usando el formato adecuado para la llamada al *endpoint*. El resultado será guardado en una variable dentro del sistema y mostrado por consola para control del usuario.
   * Componente de envío a la *API* de traducción: En este apartado se recogerá el texto generado por el componente anterior y será procesado para su traducción al idioma japonés.
   * Componente de envío a la *API* alojada en un contenedor *Docker* en el equipo: Método por el cual se enviará el texto japonés extraído del componente de traducción para su sintetización en el contenedor de *Docker* que tendremos alojado en la propia máquina, para evitar tiempos de carga en transferencia de datos.
   * Componente de reproducción del audio: En este método se recogerá el archivo de audio generado por la *API* alojada en el contenedor y será reproducido en el canal de salida del micro, para que este pueda ser reproducido por tu canal de comunicación deseado (*Discord*, *Twitch*, canal de comunicación de videojuegos, …).
4. Consideraciones de rendimiento y escalabilidad

* Optimización de las peticiones para que el proceso de traducción no demore más de 10 segundos.
* Refactorización del programa para hacerlo fácil de escalar para futuras mejoras e implementaciones.

# Implementación

# Desarrollo

Lorem ipsum

# Pruebas del sistema

Lorem ipsum

# Planificación

Lorem ipsum

# Planificación financiera

Lorem ipsum

# Conclusiones

Lorem ipsum

# Trabajo futuro

Lorem ipsum

# Bibliografía

Curso Maestro de Python en Udemy - Héctor Costa Guzmán

Speech to Text Whisper - OpenAI

Translation - OpenAI

Voicevox - Hiroshiba Kazuyuki

Docker - Docker

Female Japanese Voice - Meimei Himari

Male Japanese Voice - Ryusei Aoyama

# Anexos

1. Docker: véase la instalación de Docker y su configuración. [[Enlace]](https://www.ionos.es/digitalguide/servidores/configuracion/tutorial-docker-instalacion-y-primeros-pasos/)
2. Cómo instalar Whisper en nuestro entorno para su uso. [[Enlace]](https://github.com/openai/whisper)