# Informe Técnico

## Herramientas Utilizadas

A lo largo de la investigación he utilizado disimiles herramientas para lograr el desarrollo de el proyecto y enriquecer o facilitar su desarrollo, a continuación planteo el porque use cada una de estas.

### Herramienta para la creación del Chatbot

RASA es una herramienta de código abierto que utiliza el entendimiento del lenguaje natural para desarrollar chatbots basados en IA. Proporciona un marco que se puede utilizar para crear chatbots con habilidades de codificación mínimas. RASA permite a los usuarios entrenar y afinar el modelo a través de diversas configuraciones [1]. Este se basa en el lenguaje python, cosa que es un gran punto a su favor, dado que le brinda una gran flexibilidad a la hora de crear funciones para cosas muy particulares como por ejemplo para rellenar formularios, crear acciones para hacer determinadas tareas y etc. Rasa no es el unico framework o herramienta que sirve para esta tarea, hay otras como Dialog Flow de Google la cual hubiese sido una gran opción pero dado nuestro contexto no es viable una plataforma que exija pagos [2], otra herramienta que se examinó a profundidad para usarla en caso de confirmar su viabilidad fue la herramienta IBM Watson, otra plataforma que presentaba el mismo problema del pago de Dialog Flow, por último se investigó sobre Amazon Lex otra herramienta para crear chatbots en la nube, pero, pese a que esta herramienta es sumamente viable, hay que poseer una cuenta de AWS (Amazon Web Services) para utilizarla, de la cual la fase gratuita dura un año con variadas limitaciones, luego hay que pagar por el servicio el cual tiene precios elevados dados los requisitos de este proyecto.

Luego de esta breve exposición, se selecciona RASA como el marco de trabajo para el desarrollo del asistente debido a su naturaleza de código abierto, que no requiere pagos, facilitando su accesibilidad para estudiantes y investigadores con presupuestos limitados [3]. Además, RASA cuenta con una documentación extensa y en constante actualización, lo que facilita su aprendizaje y uso [4]. Tiene una comunidad altamente activa y una documentación amplia y rica.

### Herramienta de reconocimiento de voz

Whisper es una herramienta de inteligencia artificial diseñada para convertir el habla en texto. Es una solución innovadora que ha revolucionado el campo de la transcripción, ya que puede manejar una amplia variedad de idiomas y ofrece un alto nivel de precisión en el reconocimiento de voz [5].

Este se define como un "sistema de reconocimiento automático de voz (ASR)" y ha sido entrenado con más de 680.000 horas de datos. Este modelo puede accederse a través de la web o de forma local y es abierto a todos los usuarios [6]. Además, Whisper puede trabajar con más de 100 idiomas y cuenta con varios modelos de diferentes tamaños para su uso en servidores o computadoras con menos recursos, es clave destacar que mientras más grande sea el modelo, más consumo de recursos tendrá pero tendrá una mayor precisión.

En comparación con otras herramientas de transcripción, Whisper destaca por su capacidad para manejar acentos, ruido de fondo y lenguaje técnico. Además, ha demostrado ser especialmente efectivo para la traducción de voz a texto, superando incluso a los modelos que se especializan en el rendimiento de LibriSpeech [7].

A modo de conclusión Whisper es una herramienta valiosa para realizar transcripciones con un alto nivel de precisión. Su capacidad para manejar acentos y ruido de fondo, así como su precisión en la transcripción y su uso de forma local, lo hacen especialmente útil para este proyecto.

### Framework para la creación de la API

Al elegir un framework, es importante tener en cuenta si se prefiere trabajar con una guía definida o tener más libertad en la configuración. También es importante tener en cuenta otras métricas, como la velocidad de respuesta, el soporte a la carga de peticiones y la concurrencia. Sin embargo, no existe un framework que sea el mejor, ya que existen factores subjetivos como la productividad y las mejores prácticas que hacen imposible medir y comparar de manera objetiva [8].

Para la elección, se tuvo en cuenta que el lenguaje a utilizar sería Python. Dicho esto, se realizó una investigación de cuál sería el más adecuado.

#### Django

Django es un framework de desarrollo web de alto nivel, de código abierto, escrito en Python. Es popular por su versatilidad, productividad y seguridad. Algunas de las empresas más grandes del mundo, como Instagram, Disqus, y la NASA, utilizan Django para sus proyectos [9].

Ventajas:

El Django REST Framework es un marco de trabajo popular para APIs en Python. Ofrece una arquitectura personalizable y modular, ideal para APIs simples y complejas. Viene con operaciones CRUD y un navegador incorporado para probar los puntos finales de la API. Es especialmente útil para proyectos de Machine Learning, ya que Python es el lenguaje de programación estándar para los científicos de Machine Learning. Además, Django puede manejar grandes cantidades de tráfico y proporcionar uso de API a más de 450 millones de usuarios, lo que lo hace escalable y adecuado para soluciones empresariales de gran volumen [10].

Desventajas:

Aunque es fácil de aprender, puede ser difícil de dominar debido a su gran cantidad de configuraciones y ajustes que se deben realizar. No es recomendable para proyectos pequeños debido a su intensivo uso de recursos del servidor . Además, es un framework monolítico que no proporciona mucha flexibilidad en la arquitectura del proyecto, lo que puede ser un inconveniente para este proyecto [10]

### Flask

Flask es un microframework para Python que permite a los desarrolladores construir aplicaciones web de manera rápida y fácil. Fue desarrollado por Armin Ronacher, líder del Grupo Internacional de Pythonistas (POCCO), y se basa en el kit de herramientas WSGI y el motor de plantillas Jinja2. Es un framework altamente usado, respaldado por el uso de compañias como Samsung, Netflix, Uber y Airbnb lo cual es un motivo de peso para tomarlo en cuenta al elegir un framework para la creación del backend de este proyecto [11]

Ventajas:

Flask es un marco de trabajo web que es fácil de entender, lo que lo hace ideal para principiantes. Su simplicidad permite a los desarrolladores tener un control total sobre el desarrollo web, tomando el control creativo de la aplicación y el desarrollo web. Aunque es simple, Flask ofrece muchas características interesantes para usar en el marco. Además, Flask viene con un motor de plantillas que permite usar la misma interfaz de usuario para múltiples páginas. Python puede insertar variables en las plantillas. Flask también permite la prueba unitaria a través de su soporte integrado, servidor de desarrollo incorporado, depurador rápido y despacho de solicitudes RESTful. Es ligero para permitir transitar fácilmente a un marco de trabajo web con algunas extensiones. [12]

Desventajas:

El marco de trabajo Flask es amigable para principiantes y aficionados, lo que puede llevar a un desarrollo de código de baja calidad. Flask maneja cada solicitud una tras otra, lo que puede ralentizar el servicio de múltiples solicitudes. La inclusión de módulos adicionales puede aumentar el riesgo de seguridad si se incluye un módulo malicioso. Aunque se pueden mitigar estos problemas utilizando un hosting especializado en Python, es crucial tener en cuenta estos factores al elegir Flask para el desarrollo web [12]

#### FastAPI

FastAPI es un framework moderno y rápido (de alto rendimiento) para construir APIs con Python 3.6+ basado en estándares abiertos para APIs, como HTTP/1 y HTTP/2, y que incluye soporte para pruebas automáticas, serialización de datos JSON, y declaraciones de tipos para documentación interactiva y automatizada [13]. Con pocos años de salida al mercado, FastAPI promete convertirse en el framework definitivo para la nueva generación de aplicaciones web que ya están naciendo.

Veloz. Solo esa palabra lo define. Es (y no solo en Python, sino hablando en comparación a todos los otros lenguajes de programación) una de las herramientas más veloces para construir un backend, compitiendo con lenguajes como Go o Node.js [14]

Algunas empresas que utilizan FastAPI incluyen a Netflix, Uber, y Microsoft [14]

Ventajas:

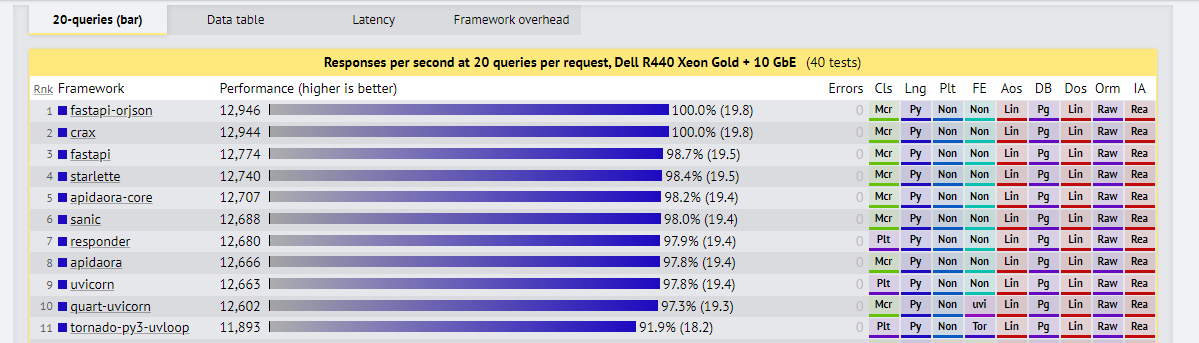
FastAPI es un framework moderno y de alto rendimiento para construir APIs con Python 3.6+. Destaca por su documentación automática, que genera una interfaz interactiva para explorar y probar la API, y por el uso de declaraciones de tipos de Python 3.6 para proporcionar un soporte de editor excelente y autocompletado. Además, FastAPI incorpora soporte para seguridad y autenticación, lo cual es esencial para cualquier API [15].

Desventajas:

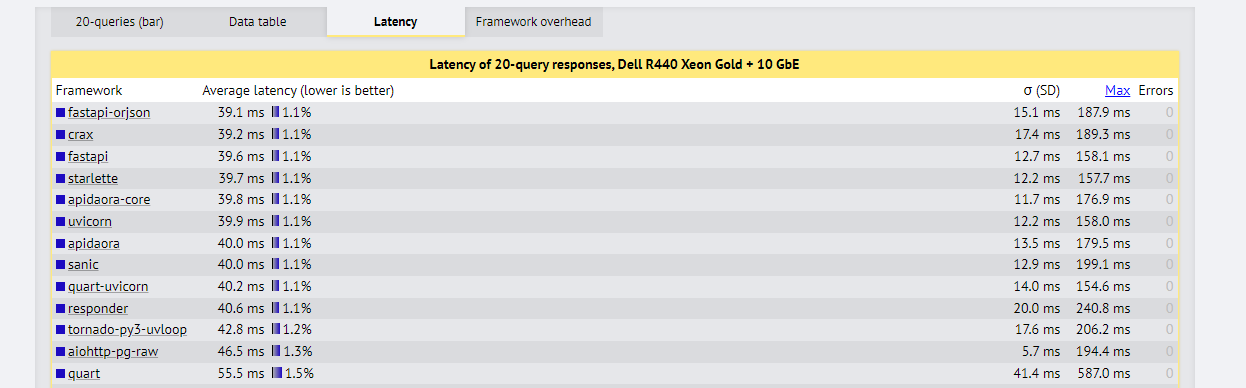
Pese a su gran robustez, es un framework relativamente nuevo, dado esto es más propenso a que no se use con frecuencia. FastAPI aún está en desarrollo y hay muchas cosas que se pueden mejorar. Aunque su comunidad está creciendo, todavía no es comparable a las comunidades de otros frameworks.

#### Benchmark

A continuación se muestra una prueba de Benchmarking realizada en el año 2022 la cual consiste en realizar 20 consultas a un servidor por segundo[16].



Como bien se puede observar en el primer y tercer lugar se encuentra FastApi mostrando su incredible manejo de concurrencia, ligereza y escasa latencia al realizar las consultas.



En esta imagen se puede apreciar la tasa de latencia al realizar las 20 consultas las cuales posicionan a FastAPI en primer y tercer lugar al igual que en la imagen anterior.

Luego de realizar un análisis crítico sobre las ventajas y desventajas de cada uno de los frameworks y de revisar la prueba de benchmark que se plasmó previamente, se llega a la conclusión de que el framework que cumple con todos los requisitos para desarrollar el backend de la aplicación que generará los asistentes virtuales es FastAPI. Esto se debe a que la aplicación necesita tener un rendimiento óptimo en múltiples consultas en tiempo real, debe ser una solución sencilla con una mínima cantidad de configuraciones, y debe ser flexible y fácil de escalar para futuras mejoras.

### Herramienta para realizar el reconocimiento de rostros

La biblioteca face-api.js es una herramienta de JavaScript que permite la detección y reconocimiento de rostros en el navegador. Esta biblioteca se construye sobre la API central de tensorflow.js y es compatible con la detección y reconocimiento de rostros, la identificación de expresiones faciales, la determinación de la edad y el género [17].

En términos de alternativas, existen varias API de reconocimiento facial disponibles, incluyendo las ofrecidas por Amazon, Microsoft, Google, IBM, Face++ de Megvii y Clarifai [17]. Estas API ofrecen una amplia gama de funciones para la detección, verificación y reconocimiento facial.

En comparación con las alternativas, face-api.js se destaca por tener un modelo entrenado previamente, lo que puede facilitar su implementación y uso, además hay que tener en cuenta que los datos no quedan almacenados ni usados por terceros con Face Api y otro punto importante es que es una libreria gratuita a diferencia de las otras tecnologías mencionadas, que algunas pese a que tienen una face gratuita, luego tienen limitaciones o cobros por servicio.

## Análisis sobre los puntos de vista luego de la culminación de la tesis de Duvalón.

Uno de los principales puntos de vista que surgió después de la culminación fue convertir el sistema de generación de asistentes virtuales en un servicio web en lugar de una interfaz de línea de comandos. Esto tendría como principales ventajas la gestión de sesiones de usuario, la autenticación y la autorización, y la capacidad de manejar solicitudes concurrentes de múltiples usuarios.

Otra de las consideraciones que se tuvieron fue que los asistentes deberían ser capaces de interactuar mediante comandos de voz mediante el reconocimiento de audio. También deberían ser capaces de recopilar datos de los usuarios a través de formularios y proporcionar una atención personalizada basada en las emociones que el usuario exprese al interactuar con el asistente mediante el reconocimiento de emociones.

## Análisis de las últimas tendencias en LLM

ChatGPT es un robot virtual con el que podemos conversar, un ChatBot, pero que a diferencia de los empleados por algunas compañías para sus servicios de atención al cliente, es capaz de generar texto de una forma muy coherente y adaptada a cada contexto de forma muy parecida como lo haría un ser humano [18].

Se centra en el usuario, lo que significa que está diseñado para seguir las instrucciones del usuario de manera segura y útil. Este enfoque centrado en el usuario es una innovación significativa que distingue a ChatGPT de los modelos de lenguaje largo anteriores

Este modelo ha presentado grandes avances, en cuanto a sus sesgos y a la capacidad de burlar sus principios, ya esta tarea no es tan sencilla como antes.

Llama 2 (Large Language Model Meta AI 2) es una tecnología de procesamiento de lenguaje natural de última generación lanzada en tres tamaños de modelo: 7B, 13B y 70B. Puede generar texto, resumir o reescribir textos existentes y llevar a cabo conversaciones similares a las interacciones humanas utilizando el modelo de lenguaje Llama 2 [19].

Es importante destacar que Llama 2 es de código abierto, lo que permite a cualquier usuario explorarlo de forma gratuita. También es gratuito tanto para fines de investigación como comerciales [19].

Pese a que Llama 2 no es un modelo totalmente fiable dado que alucina con bastante frecuencia y puede producir sesgos raciales, sexuales, religiosos y etc, existe el modelo Llama 2 Chat el cual esta en óptimas condiciones para realizar una interacción humana sin problemas, comparándose en algunos casos con las de ChatGPT pese a que este último es más eficiente en la mayor parte de las tareas y da resultados más precisos.

La principal ventaja de Llama 2 sobre ChatGPT es que tiene modelos para diferentes requisitos de software, por ejemplo su modelo más pequeño Llama 2 Chat 7B puede desplegarse de forma local en una pc con 4GB de RAM solamente sin utilizar GPU para ejecutar las tareas que oriente el usuario.

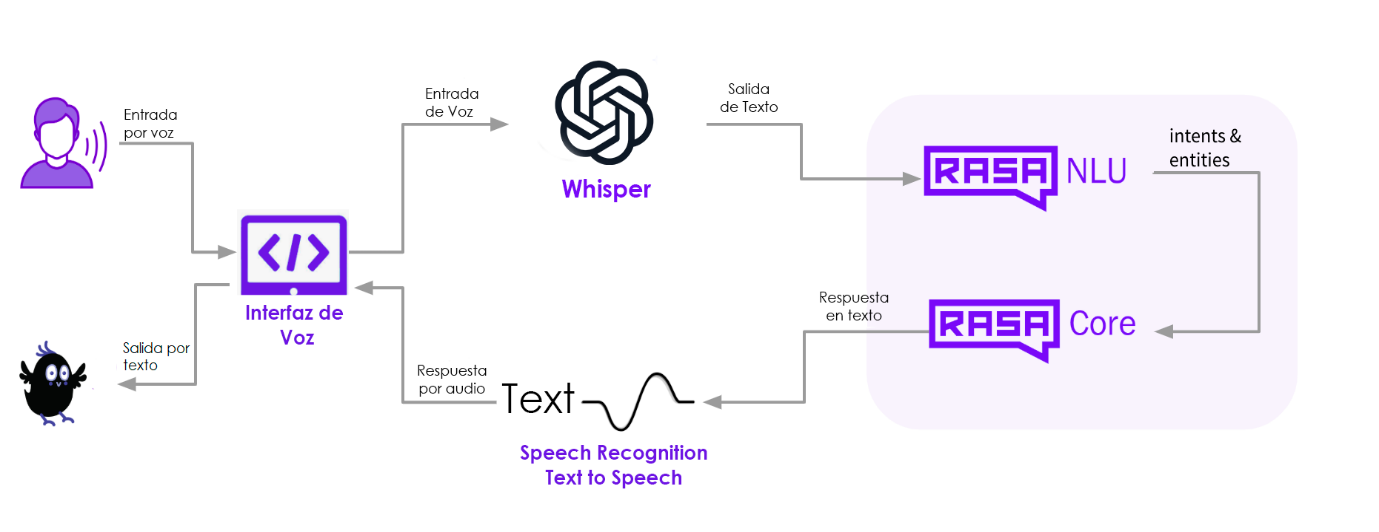
A modo de conclusión, pese al avance de estas tecnologías, en este proyecto se desestima su uso, debido a que estos asistentes abarcan un dominio general, mas no específico, pese a que Llama 2 se le puede hacer un proceso de fine-tunning este proceso es costoso en cuanto a recursos y no sería una alternativa viable, por último pese a sus altos niveles de fiabilidad estos modelos continúan produciendo errores y sesgos, cosa que es inaceptable en el contexto de creación de asistente que nos encontramos, un ejemplo claro sería el asistente para el Hospital Infantil Sur, si este asistente produce un sesgo inapropiado hacia un paciente sería un daño psicológico, el cual se pudiera haber evitado.

### Aportes de la investigación:

* Desarrollar una aplicación web para el sistema de creación de asistentes virtuales para permitir su uso de forma concurrente y accesibilidad sin usar recursos propios de la computadora de el cliente y permitiendo la creación de asistentes de diferentes dominios con mínimas configuraciones.
* Realizar una atención personalizada al usuario mediante el reconocimiento de su voz y sus expresiones
* Desarrollar con el sistema un asistente virtual para el Hospital Infantil Sur (La Colonia) para la atención de las necesidades de médicos y pacientes de esta institución.
* Permitir el auto-ajuste de los asistentes para no tener que crear uno nuevo cada vez que se quiera realizar una actualización.

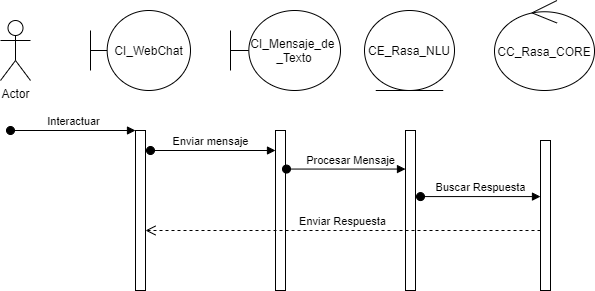
## Diagramas

### Diagrama de Arquitectura del Asistente de Voz



### Diagrama de Secuencia

El diagrama de secuencia nos muestra la ruta que sigue el chatbot para poder generar una respuesta, se observa que primeramente el usuario interactúa con la interfaz web y envía un mensaje de texto el cual llega RASA\_NLU componente que se encargará de tomar el texto, procesarlo, analizarlo y descomponerlo de tal manera que el bot comprenda el contenido del mensaje, luego busque si existe alguna respuesta en RASA\_CORE y la envíe al usuario.

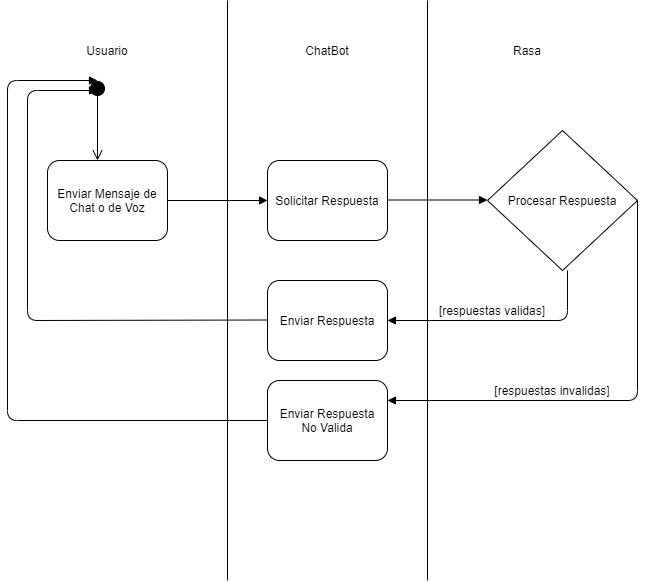


### Diagrama de Actividades del Asistente

El diagrama de actividades representa el flujo de interacción entre el usuario y el chatbot. El proceso comienza cuando el usuario envía un mensaje de chat o voz al chatbot. Luego, el chatbot solicita una respuesta al usuario, lo que indica que está listo para recibir una entrada.

El chatbot procesa la respuesta del usuario utilizando el marco de desarrollo de software RASA NLU, que es una herramienta de procesamiento del lenguaje natural que ayuda a comprender y extraer información relevante de la entrada del usuario. Si RASA NLU encuentra una respuesta válida, el chatbot la envía al usuario como respuesta.

Sin embargo, si RASA NLU no puede encontrar una respuesta válida, el chatbot envía un mensaje al usuario indicando que su respuesta no es válida. Esto puede deberse a que la entrada del usuario no se ajusta a los patrones o intenciones que el chatbot reconoce.



## Consideraciónes personales:

Tareas que faltan por realizar:

* Terminar el backend de la aplicación en cuanto tenga disponible el modelo generador de preguntas y respuestas que usa Duvalón en su algoritmo 1 semana.
* Realizar el diseño de la interfaz del Sistema de creación de asistentes virtuales 3 días
* Someter a una etapa de pruebas de 1 semana a la API del sistema de creación de asistentes luego de que este sea terminado.
* Realizar una estadística de el promedio de tiempo que se tarda el modelo Whisper en transcribir una serie de audios con diferentes duraciones (Esto me lo pidió la profesora guía para la discución de la práctica laboral)
* Realizar un asistente de el dominio del Hospital Infantil Sur para realizarle pruebas de estrés.

# Referencias Bibliográficas:

[1]L. Ajay, “AI chatbots made easy — courtesy RASA,” Medium. Accessed: Oct. 07, 2023. [Online]. Available: https://towardsdatascience.com/ai-chatbots-made-easy-courtesy-rasa-8ecdc2853e66  
   
   
 [2]“Dialogflow de Google cambia su modelo de pricing: ¿cuánto cuesta la IA / NLU? - Planeta Chatbot.” Accessed: Oct. 07, 2023. [Online]. Available: https://planetachatbot.com/dialogflow-google-cambia-modelo-pricing/  
   
   
 [3]“Chatbots Using Python and Rasa,” GeeksforGeeks. Accessed: Oct. 07, 2023. [Online]. Available: https://www.geeksforgeeks.org/chatbots-using-python-and-rasa/  
   
   
 [4]H. Sharma, “Developing Chatbots with RASA- Intuition to Implementation,” DataX Journal. Accessed: Oct. 07, 2023. [Online]. Available: https://medium.com/data-science-community-srm/developing-chatbots-with-rasa-intuition-to-implementation-39c1dd34274c  
   
   
 [5]“¿Qué es y cómo funciona Whisper? | Código Espagueti.” Accessed: Oct. 07, 2023. [Online]. Available: https://codigoespagueti.com/noticias/tecnologia/que-es-como-funciona-whisper/  
   
   
 [6]“He usado Whisper para transcribir una entrevista: es la herramienta que llevaba esperando desde hace años.” Accessed: Oct. 07, 2023. [Online]. Available: https://www.xataka.com/aplicaciones/he-usado-whisper-para-transcribir-entrevista-herramienta-que-llevaba-esperando-hace-anos  
   
   
 [7]M. Terol, “Whisper: optimiza el tiempo a la hora de hacer una transcripción,” Blogthinkbig.com. Accessed: Oct. 07, 2023. [Online]. Available: https://blogthinkbig.com/whisper-transcribir  
   
   
 [8]“Cómo elegir un framework para el backend,” CódigoFacilito. Accessed: Oct. 07, 2023. [Online]. Available: https://codigofacilito.com/articulos/elegir-framework-backend  
   
   
 [9]“Qué es Django y por qué usarlo,” OpenWebinars.net. Accessed: Oct. 07, 2023. [Online]. Available: https://openwebinars.net/blog/que-es-django-y-por-que-usarlo/  
   
   
 [10]P. Team, “Advantages of Django | Disadvantages of Django,” Python Geeks. Accessed: Oct. 07, 2023. [Online]. Available: https://pythongeeks.org/advantages-disadvantages-of-django/  
   
   
 [11]A. J. Condez, “Top 10 Big Companies Using Flask,” Career Karma. Accessed: Oct. 07, 2023. [Online]. Available: https://careerkarma.com/blog/companies-that-use-flask/  
   
   
 [12]“Python Flask: pros and cons,” DEV Community. Accessed: Oct. 07, 2023. [Online]. Available: https://dev.to/detimo/python-flask-pros-and-cons-1mlo  
   
   
 [13]“Características - FastAPI.” Accessed: Oct. 07, 2023. [Online]. Available: https://fastapi.tiangolo.com/es/features/  
   
   
 [14]“¿Qué sigue después de aprender Python? Django vs. Flask vs. FastAPI,” Platzi. Accessed: Feb. 01, 2023. [Online]. Available: https://platzi.com/blog/django-flask-fastapi/  
   
   
 [15]“Explicación de FastAPI en 5 minutos o menos - Geekflare.” Accessed: Oct. 07, 2023. [Online]. Available: https://geekflare.com/es/fastapi-explained/  
   
   
 [16]“TechEmpower Web Framework Performance Comparison,” www.techempower.com. Accessed: Oct. 07, 2023. [Online]. Available: https://www.techempower.com/benchmarks/#section=data-r21&hw=ph&test=query&l=zijzen-sf  
   
   
 [17]“10 principales API de Reconocimiento Facial - Infoteknico.” Accessed: Oct. 07, 2023. [Online]. Available: https://www.infoteknico.com/principales-api-de-reconocimiento-facial/  
   
   
 [18]“Preguntas y respuestas sobre ChatGPT la inteligencia artificial de moda,” www.nationalgeographic.com.es. Accessed: Oct. 07, 2023. [Online]. Available: https://www.nationalgeographic.com.es/ciencia/chatgtp-esta-boca-todos-nadie-sabe-que-es\_19461  
   
   
 [19]“¿Para qué se utiliza el LLaMa-2?” Accessed: Oct. 07, 2023. [Online]. Available: https://textcortex.com/es/post/what-is-llama-2-used-for