
Integración de programas para mantener una conversación con una inteligencia artificial

Juan Emilio Reyes Orantes



UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería



**Integración de programas para mantener una conversación
con una inteligencia artificial**

Trabajo de graduación presentado por Juan Emilio Reyes Orantes para
optar al grado académico de Licenciado en Ingeniería Mecatrónica

Guatemala,

2025

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería




**Integración de programas para mantener una conversación
con una inteligencia artificial**

Trabajo de graduación presentado por Juan Emilio Reyes Orantes para
optar al grado académico de Licenciado en Ingeniería Mecatrónica


Guatemala,

2025


Vo.Bo.:

(f) 
M. Sc. Carlos Esquit

Tribunal Examinador:

(f) 
M.Sc. Carlos Esquit

(f) 
M. Sc. Miguel Enrique Zea Arenales

(f) 
Ing. Kurt Emmanuel Kellner

Fecha de aprobación: Guatemala, 13 de febrero de 2025.

Este trabajo de graduación nació de la motivación por explorar y aprender más sobre la inteligencia artificial, un tema que ha ganado enorme popularidad en la industria. Fue un viaje de aprendizaje y dedicación que me permitió adentrarme en un campo fascinante y desafiante.

Quiero expresar mi profundo agradecimiento a mi asesor, Pedro Joaquín Castillo, cuya guía fue fundamental para completar este proyecto. Su enfoque equilibrado, permitiéndome aprender de forma autónoma mientras me orientaba, fue clave para superar cada obstáculo y llevar a cabo este trabajo de la mejor manera posible.

Agradezco también a las personas que me inspiraron, apoyaron y acompañaron a lo largo de este proceso. En especial, a mi gran mentor, ejemplo a seguir y padre, Ian Reyes, quien no solo me brindó apoyo emocional, profesional y económico, sino que también fue un pilar fundamental para alcanzar esta meta. Su constante fe en mí y su presencia incondicional me dieron la fuerza para llegar hasta aquí. Celebrar este logro junto a él es un privilegio y una bendición que agradezco profundamente.

Asimismo, quiero agradecer a mi guerrera, mi mujer favorita y la persona que me dio la vida, Aída Orantes de Reyes. Este año fue particularmente difícil para ella, pero su apoyo nunca faltó. Su firmeza, amor y enseñanzas fueron mi motor para seguir adelante. Me mostró que, con determinación, todo lo que uno se propone puede lograrse. Su ejemplo de resiliencia y fortaleza ha sido una de las mayores lecciones que llevaré conmigo siempre.

Por último, pero no menos importante, quiero agradecer a Dios por guiarme en este camino, por darme la oportunidad de culminar esta etapa y por todas las bendiciones en mi vida. Sin Su guía y Su presencia, este logro no habría sido posible. Agradezco profundamente tenerlo en mi vida y haber contado con Su fortaleza durante este proceso.

Con gratitud y humildad, dedico este trabajo a todos aquellos que creyeron en mí.

Prefacio	III
Lista de figuras	VIII
Resumen	X
Abstract	XI
1. Introducción	1
2. Antecedentes	2
3. Justificación	7
4. Objetivos	8
4.1. Objetivo general	8
4.2. Objetivos específicos	8
5. Alcance	9
6. Marco teórico	10
7. Preparación y gestión de proyectos con librerías y almacenamiento de datos	18
7.1. Instalación de librerías	18
7.1.1. Librerías de implementación de una interfaz de uso y control para el rostro animatrónico de la Universidad del Valle de Guatemala	18
7.1.2. Librerías de implementación de una inteligencia artificial capaz de crear una conversación en la plataforma del rostro animatrónico	19
7.2. Análisis de proyectos	19
7.2.1. Lectura trabajo escrito	19
7.2.2. Entendimiento de código	20
7.2.3. Entendimiento de funcionamiento	20
7.3. Correcto funcionamiento de proyecto de detección de emociones	20

7.3.1.	Interfaz de uso	20
7.3.2.	Correcto funcionamiento de detección de emociones	21
7.4.	Correcto funcionamiento de proyecto de reentrenamiento	24
7.4.1.	Interfaz de uso	24
7.4.2.	Almacenamiento en un archivo	25
7.4.3.	Archivo tipo .text	26
8.	Configuración y diseño de la integración de la API	27
8.1.	Creación de correo institucional	27
8.2.	Creación de API Key OpenAI	27
8.3.	Adquisición de <i>tokens</i> OpenAI	28
8.4.	Conexión con ChatGPT	28
8.5.	Limitación de <i>tokens</i> para respuestas dadas	29
8.6.	Definición de roles a ChatGPT	30
8.7.	Creación de diseño para interfaz gráfica	31
8.7.1.	Creación de menú	31
8.7.2.	Creación de educa a la inteligencia artificial	32
8.7.3.	Creación de detección de emociones	32
8.7.4.	Creación de información almacenada en la inteligencia artificial	33
8.7.5.	Creación de conversación en vivo	33
8.7.6.	Creación de información extra	34
9.	Desarrollo de la plataforma web e integración de funcionalidades de inteligencia artificial	35
9.1.	Creación de página web	35
9.2.	Creación de páginas aleadas	38
9.2.1.	Educa a la inteligencia artificial	38
9.2.2.	Detección de emociones	42
9.2.3.	Información almacenada	47
9.2.4.	Conversación en vivo	54
9.2.5.	Información extra	65
10.	Entrenamiento de la inteligencia artificial	67
10.1.	Creación de archivo .json con información deseada	67
10.1.1.	Información general de la Universidad del Valle	67
10.1.2.	Información de Ingeniería Electrónica, Mecatrónica y Biomédica	68
10.1.3.	Información detallada de Ingeniería Mecatrónica	68
10.1.4.	Información detallada de Ingeniería Electrónica	71
10.1.5.	Información detallada de Ingeniería Biomédica	74
10.2.	Entrenamiento de redes neuronales de ChatGPT	76
11.	Validación y pruebas de correcto funcionamiento del proyecto	78
11.1.	Conversación con inteligencia artificial, relacionada a la Universidad del Valle, tomando en cuenta la toma de emociones	78
12.	Conclusiones	84
13.	Recomendaciones	85

Lista de figuras

1.	Entrada de texto, para obtener información	3
2.	Proyecto rostro animatrónico	4
3.	Tipos de API	16
4.	Comando ingresado	20
5.	Interfaz de uso	21
6.	Emoción de tristeza detectada	21
7.	Emoción de alegría detectada	22
8.	Emoción de miedo detectada	22
9.	Emoción de disgusto detectada	23
10.	Emoción de sorpresa detectada	23
11.	Emoción de enojo detectada	24
12.	Terminal para ingresar texto	25
13.	Información ingresada, prueba	25
14.	Archivo data.txt	26
15.	Contenido del archivo data.txt	26
16.	Dinero disponible en <i>tokens</i>	28
17.	Verificación de comunicación con el servidor	28
18.	Página web de FastAPI	29
19.	Prueba de respuesta en ChatGPT	30
20.	Max- <i>tokens</i> y su valor	30
21.	Menú para el rostro animatrónico	31
22.	Pestaña donde se educa a la IA	32
23.	Pestaña de detección de emociones	32
24.	Pestaña donde se descarga la información almacenada	33
25.	Pestaña para la conversación en vivo	33
26.	Página web oficial de la Universidad del Valle	34
27.	Página web rostro animatrónico	38
28.	Pestaña donde se educa a la IA	41
29.	Pestaña donde se educa a la IA, texto ingresado	41
30.	Pestaña donde se educa a la IA, información guardada	42

31.	Página principal de rostro animatrónico	42
32.	Pestaña detección de emociones	45
33.	Detección de emociones, enojo	46
34.	Detección de emociones, felicidad	46
35.	Detección de emociones, tristeza	47
36.	Pestaña principal	47
37.	Pestaña información almacenada	52
38.	Pestaña información almacenada adicional	52
39.	Archivo descargado	53
40.	Información que contiene el archivo .json	53
41.	Página principal rostro animatrónico	53
42.	Primera vista de conversación en vivo	62
43.	Pestaña conversación en vivo, conectado a extensión	62
44.	Interacción entre detección de emoción y el texto ingresado	63
45.	Opción de texto a voz activada	63
46.	Selección de archivos en la computadora	64
47.	Archivo seleccionado y agregado en el programa	64
48.	Pestaña principal	65
49.	Página web UVG	66
50.	Información general de la universidad	68
51.	Información extraída de la página web	68
52.	Conversación con inteligencia artificial y detección de emociones	79
53.	Entrada de voz, verificación de transcripción	79
54.	Pregunta relacionada con el Departamento de Mecatrónica	80
55.	Pregunta específica, director de carrera	80
56.	Subtítulos generados de los sonidos emitidos	81
57.	Conversación en otro idioma	82
58.	Pregunta relacionada con el Departamento de Electrónica	83

Un rostro animatrónico es una figura controlada mediante sistemas electrónicos, mecánicos o neumáticos. Esta tecnología permite diversas representaciones, y en este caso específico, se ha desarrollado un rostro con inspiración en las características de un rostro humano.

El rostro animatrónico UVG será utilizado para atraer a diversas personas a que conozcan los distintos proyectos realizados por el Departamento de Ingeniería Mecatrónica, Electrónica y Biomédica. De este modo, será un canal de mercadeo para atraer nuevos estudiantes.

El presente proyecto aborda el desarrollo e integración de programas, incluyendo una interfaz gráfica con detección de emociones y un *chatbot*. Otro programa es un sistema de comunicación aliado con ChatGPT. Este sistema genera respuestas conversacionales coherentes y contextuales basadas en las interacciones con los usuarios, iterando constantemente para estar mejor preparado y completar la conversación de manera efectiva.

Para la realización de este proyecto se hará un análisis exhaustivo de ambos programas, con el fin de entender de forma completa el funcionamiento. Así se procederá con el diseño de una arquitectura de unificación, la cual es el prototipo de la interfaz gráfica, siendo encargada que la experiencia del usuario sea amigable y agradable. Se continuará con la integración de los sistemas y la realización de pruebas unitarias para asegurar la correcta comunicación y funcionamiento entre ambos programas.

Los resultados a obtener del proyecto demostrarán el correcto funcionamiento de la integración de los programas. El usuario final tendrá la posibilidad de establecer una conversación coherente y correcta, así el programa unificado responderá distintas preguntas acerca de la universidad y el departamento. De este modo, contará con distintas áreas de conocimiento para sostener una conversación, dependiendo únicamente del entrenamiento que se le proporcione a la inteligencia artificial. Según el tema en el cual se le eduque, será capaz de responder preguntas relacionadas con ese campo.

En conclusión, la unificación de estos programas, que detectan emociones y sostienen conversaciones coherentes y fluidas mediante inteligencia artificial, contribuye a captar la atención de diversas personas, atrayendo así a nuevos estudiantes. Esto sentará las bases para futuras investigaciones y proyectos en los ámbitos de mecánica, electrónica e inteligencia

artificial.

An animatronic face is a figure controlled by electronic, mechanical, or pneumatic systems. This technology allows for various representations, and in this specific case, a face has been developed inspired by the characteristics of a human face.

The UVG animatronic face will be used to attract people to explore the different projects carried out by the departments of Mechatronics, Electronics, and Biomedical Engineering. Thus, it will serve as a marketing channel to draw new students.

This project focuses on the development and integration of programs, including a graphical interface with emotion detection and a chatbot. Another component is a communication system powered by ChatGPT. This system generates coherent and contextual conversational responses based on user interactions, constantly iterating to improve and effectively complete conversations.

To achieve this project, an in-depth analysis of both programs will be conducted to fully understand their functionality. This will be followed by the design of a unified architecture, represented by the graphical interface prototype, ensuring a user-friendly and enjoyable experience. The integration of systems will continue, along with unit testing to ensure proper communication and functionality between the programs.

The expected outcomes of this project will demonstrate the successful integration of both programs. The end user will be able to engage in coherent and meaningful conversations, and the unified system will answer various questions about the university and its departments. The system will have different areas of knowledge, depending solely on the training provided to the artificial intelligence. Based on the topics it is taught, it will be capable of responding to related questions.

In conclusion, the unification of these programs—enabling emotion detection and coherent, fluid conversations through artificial intelligence—will help capture the attention of a diverse audience, attracting new students. This initiative lays the groundwork for future research and projects in the fields of mechanics, electronics, and artificial intelligence.

CAPÍTULO 1

Introducción

Según pasa el tiempo, la relación entre humanos y máquinas se ha normalizado cada vez más, mostrando un incremento exponencial a través de los años, especialmente desde la creación y familiarización con la inteligencia artificial. Esto ha permitido desarrollar diversos programas, herramientas y aplicaciones efectivas en distintos rubros y contextos, como es el caso del buscador de OpenAI, llamado ChatGPT.

Este proyecto de graduación se enfoca en el diseño e integración de un programa con la capacidad de incorporarse a un rostro animatrónico, permitiendo la detección de emociones y, con base en ello, sostener una conversación personal y continua. Este sistema se utilizará como un canal de interacción para atraer nuevos estudiantes y promover los diversos proyectos desarrollados por el Departamento de Ingeniería Mecatrónica, Electrónica y Biomédica. El proyecto se basa en la unión de dos componentes clave: el reconocimiento de imágenes y la conexión con una inteligencia artificial entrenada con información de la Universidad del Valle de Guatemala, con el objetivo de facilitar una conversación coherente y significativa.

La metodología utilizada comienza con un análisis e investigación profunda de las funciones necesarias para garantizar un conocimiento completo de los sistemas que se integrarían. Otro aspecto clave es asegurar una experiencia de usuario amigable, haciendo que la interfaz final sea intuitiva y fácil de utilizar para el usuario.

Este trabajo no solo evidencia las posibles aplicaciones de la inteligencia artificial y la programación, sino que también resalta la unión de distintos campos que convergen en el desarrollo de este proyecto. La integración de componentes digitales y físicos representa un ejemplo práctico de aplicación robótica, demostrando la colaboración exitosa entre áreas como la mecánica, la electrónica y la inteligencia artificial.

El rostro animatrónico UVG se utilizará para presentar los departamentos de ingeniería mecatrónica, electrónica y biomédica, sirviendo como una herramienta de promoción para atraer a posibles nuevos estudiantes. Su función principal es permitir que los visitantes establezcan una conversación con el rostro animatrónico, lo que les permitirá conocer distintos temas y aplicaciones de estas carreras, tales como la robótica, la electrónica, la mecánica, el diseño y la programación de microcontrolador.

Un microcontrolador es un circuito integrado que en su interior contiene una unidad central de procesamiento (CPU), unidades de RAM y ROM, puertas de entrada, salida y periféricos; así siendo interconectadas entre ellas, conociéndose como una pequeña computadora [1].

El rostro animatrónico es una figura controlada a través de sistemas electrónicos, mecánicos y neumáticos. Esta tecnología permite diversas representaciones y, en este caso específico, se ha desarrollado un rostro inspirado en las características de un rostro humano.

La palabra “animatrónico” surge de la combinación de las palabras “animación” “electrónica”. Este proyecto se hace posible mediante el uso de circuitos, controladores, diseños mecánicos, conexiones y programación, entre otros procesos, con el objetivo de crear una figura animada que impresione y entretenga a los espectadores.

Como dijo Homero: "La unión hace la fuerza". La integración de programas computacionales tiene el fin de unir y mejorar diferentes tecnologías, combinándose para lograr avances significativos. Muchos de los desarrollos actuales son continuaciones de programas ya existentes, mejorándolos para obtener diversos beneficios.

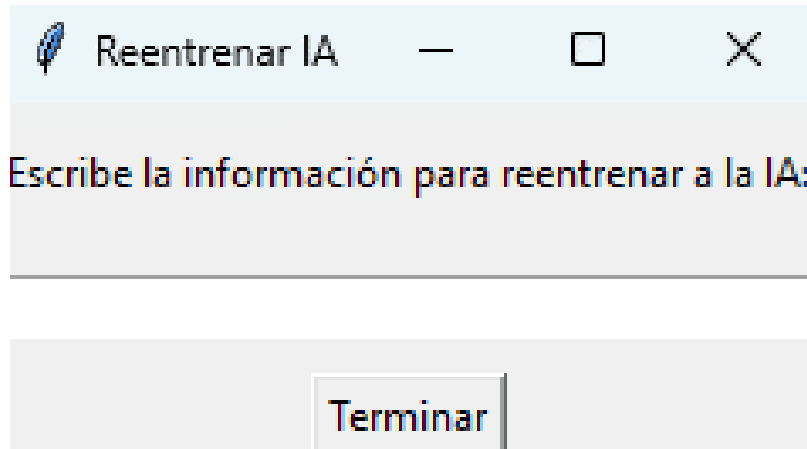


Figura 1: Entrada de texto, para obtener información

Implementación de una inteligencia artificial capaz de crear una conversación en la plataforma del rostro animatrónico

En la Universidad del Valle de Guatemala se desarrolló un sistema de inteligencia artificial que interactúa con los usuarios a través de un rostro animatrónico, logrando una conversación fluida y natural. Este sistema ha sido implementado en varios proyectos de graduación, destacando el titulado *Implementación de una inteligencia artificial capaz de crear una conversación en la plataforma del rostro animatrónico*[2].

La integración de este sistema se logró utilizando diversas herramientas accesibles para cualquier persona con internet. La inteligencia artificial utilizada, llamada Chat GPT, tiene múltiples extensiones que permiten su uso en diferentes proyectos. Para este proyecto, se personalizó una extensión de esta inteligencia artificial para dotarla de la información necesaria para responder preguntas sobre la Universidad del Valle de Guatemala.

El proceso se lleva a cabo de la siguiente manera: el usuario hace una pregunta al rostro animatrónico, que escucha mediante un micrófono. La pregunta se transcribe en un archivo de texto, se vectoriza y se introduce en la herramienta de Chat GPT. La inteligencia artificial genera una respuesta, que se convierte en un archivo de texto y se comunica al usuario a través de bocinas.

Los resultados se organizaron según los objetivos planteados. Primero, se estableció la comunicación con la inteligencia artificial mediante un micrófono. El usuario formula una pregunta, que el rostro animatrónico capta y transcribe. Esta pregunta se envía a la inteligencia artificial, que responde tras consultar su base de datos, permitiendo así una conversación fluida. Además, se entrenó a la inteligencia artificial con datos específicos de la Universidad del Valle de Guatemala. Este entrenamiento se facilitó mediante una interfaz gráfica donde se ingresa la información relevante, que se almacena en un archivo de texto utilizado como base de datos por la inteligencia artificial. Finalmente, se implementó la respuesta de la inteligencia artificial a través de bocinas conectadas al rostro animatrónico, utilizando un lector de voz amigable para mejorar la experiencia de conversación.

En conclusión, el proceso para establecer una conversación fluida con un rostro animatró-

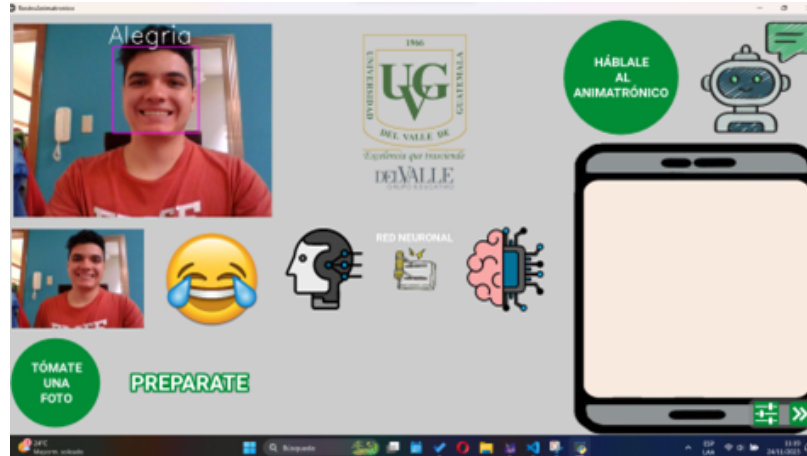


Figura 2: Proyecto rostro animatrónico

El proyecto se completó exitosamente. Este proyecto demostró los pasos y herramientas necesarias para alcanzar el objetivo principal, subrayando la importancia de la integración de tecnologías avanzadas para mejorar la interacción y la educación.

Implementación de una interfaz de uso y control para el rostro animatrónico de la Universidad del Valle de Guatemala

Se desarrolló una interfaz de uso y control para el rostro animatrónico, la cual detectaba las emociones de los usuarios, logrando así una conversación fluida y natural. Esta interfaz ha sido implementada en diferentes proyectos de graduación, destacándose el proyecto titulado *Implementación de una interfaz de uso y control para el rostro animatrónico*.

La metodología utilizada se basó en validaciones, dado que este proyecto ya había sido trabajado anteriormente. Se comenzó validando el funcionamiento de los diferentes segmentos de la cara animatrónica, para luego proceder a implementar una interfaz gráfica que muestra el reconocimiento facial y de voz a través de los motores y sensores del rostro animatrónico.

La interfaz gráfica contaba con detección de emociones y, a través de un patrón de estímulos, podía identificar la emoción del usuario. Esto era posible gracias a una base de datos que almacenaba las distintas emociones posibles y sus correspondientes expresiones faciales, tal como se muestra en la Figura 2.

El objetivo de identificar la emoción del usuario era establecer una conversación basada en esa emoción, aconsejando distintos métodos o hechos para cambiar o mantener su estado de ánimo feliz.

La interfaz gráfica también incluía una función llamada *Háblale al animatrónico*, diseñada para establecer una conversación con el animatrónico mediante un chatbot, el cual ofrecía respuestas inmediatas almacenadas en su base de datos. Esto se diferencia de una inteligencia artificial, que es capaz de generar sus propias respuestas.

Los resultados fueron acertados, cumpliendo con los objetivos planteados. La unificación de las interfaces con el rostro fue exitosa, permitiendo establecer una conversación a través de respuestas predefinidas. Sin embargo, los programas de inteligencia artificial y de detección de emociones no se unificaron, ya que fueron trabajados en paralelo[3].

Brazo robótico de 3 grados de libertad con inteligencia artificial

En el Instituto Tecnológico de Lázaro Cárdenas, un grupo de estudiantes desarrolló un brazo robótico capaz de escribir los números proporcionados por el usuario, integrando así la mecánica con la inteligencia artificial. La inteligencia artificial se utilizó para mejorar la exactitud, rapidez y calidad de la máquina al escribir los números.

La implementación involucró la resolución de los cálculos de la cinemática directa e inversa combinados con redes neuronales, logrando así mover el brazo robótico a una posición específica para iniciar la escritura del número indicado.

Una limitación del proyecto fue que el brazo robótico solo podía manejar un dígito a la vez. Para superar esta limitación, era necesario agregar al menos otro dígito. Además, la iteración para el aprendizaje de la base de datos también impedía manejar múltiples dígitos, ya que se requería almacenar una gran cantidad de números para realizar una comparación efectiva con el número ingresado por el usuario.

En los resultados obtenidos, se observa que el usuario puede ingresar cualquier número del 0 al 9. El programa detecta este número, lo compara con la información en su base de datos y luego registra el número obtenido después de la comparación, permitiendo comprobar si el sistema funciona correctamente[4].

Síntesis de voz y reconocimiento del habla. Implementación en el robot HOAP-3

En la Universidad Carlos III de Madrid, se desarrolló un programa con síntesis de voz y reconocimiento del habla implementado en el Robot HOAP-3, bajo la dirección de Pablo Marín Plaza. El proyecto llevó por título *Síntesis de voz y reconocimiento del habla. Implementación en el robot HOAP-3*.

El objetivo principal del programa era capacitar al robot para imitar al ser humano y así clonar tareas humanas. Este robot fue utilizado para exhibir una nueva fase de la empresa Fujitsu, que buscaba incursionar en el mercado de la robótica. Durante las demostraciones, el robot realizó diversas tareas, incluyendo caminar, trotar, cargar objetos y hablar en distintos idiomas.

El proyecto logró alcanzar todos los objetivos planteados y superar varios desafíos. Durante las demostraciones, se observó que cuando el robot HOAP-3 hablaba en inglés, su voz sonaba un tanto robótica pero era clara y comprensible. En español, aunque el entendimiento era aceptable, requería mayor atención por parte del oyente. Además, se validó el funcionamiento del reconocimiento del habla, mostrando que el robot realizaba un reconocimiento continuo y automático.

La interacción humano-robot es crucial para la automatización de procesos, dado que es necesario proporcionar órdenes claras al robot para que las pueda ejecutar. Este proyecto se enfocó en resolver esta problemática.

La metodología utilizada se dividió en varias etapas: estudio de síntesis y reconocimiento del habla, implementación de síntesis de voz, implementación de reconocimiento del habla y demostración de los sistemas en el robot humanoide. Todos estos procesos fueron trabajados con el fin de integrarlos de manera efectiva. Los programas funcionales se integraron y probaron en las entradas del sistema.

Una de las limitaciones encontradas fue la dificultad para que el robot pudiera trabajar con el idioma español, dado que la mayoría de las librerías disponibles estaban en inglés[5].

Chat GPT

Chat GPT es una aplicación de *chatbot* que cuenta con un modelo de lenguaje por inteligencia artificial llamado GPT-3.5, desarrollado por la empresa OpenAI. Este sistema utiliza redes neuronales con más de 175 millones de parámetros y ha sido entrenado con grandes cantidades de texto para realizar tareas relacionadas con el lenguaje.

Cuando se habla de *entrenar* en este contexto, significa que el sistema realiza iteraciones, es decir, se le introduce información a través de una interfaz gráfica. Esta interfaz se encarga de almacenar la información en un archivo (.text), que a su vez se guarda en la base de datos de la inteligencia artificial.

A medida que se le hacen preguntas y se agrega información a esta inteligencia artificial, Chat GPT realiza correlaciones a lo largo del tiempo, creando conexiones entre sus redes neuronales. Estas conexiones son las que permiten que el sistema proporcione información incluso si no está presente en su base de datos. Cuanta más información tenga, más precisa será su respuesta[6].

En la Universidad del Valle, se busca captar la atención de potenciales estudiantes, especialmente jóvenes, para que consideren estudiar ingeniería mecatrónica. Para este fin, se ha diseñado el proyecto rostro animatrónico, el cual tiene la capacidad de entablar conversaciones y responder preguntas relacionadas con la universidad, dejando una impresión memorable en los visitantes.

Como parte de sus esfuerzos de marketing, la institución realiza diversas actividades para atraer nuevos miembros a la comunidad UVG. En este contexto, el proyecto del rostro animatrónico se presenta al público con el objetivo de atraer a personas hacia los Departamentos de Ingeniería Mecatrónica, Electrónica y Biomédica.

Dado el impacto significativo que la tecnología tiene en la sociedad, así como la importancia de la primera impresión, el rostro animatrónico se convierte en una herramienta clave. Su propósito es impresionar y motivar a los jóvenes para que consideren participar en proyectos similares, actividades y estudiar carreras relacionadas en el departamento correspondiente.

Por lo tanto, la integración de programas es crucial para atraer a nuevos estudiantes y aprovechar esta iniciativa como parte de las estrategias de marketing de la universidad. Además, esta integración será de gran utilidad para aquellas personas que necesiten continuar trabajando o desarrollando temas relacionados con esta iniciativa.

4.1. Objetivo general

Integración de programas para mantener una conversación con una inteligencia artificial.

4.2. Objetivos específicos

- Estudiar el funcionamiento de los programas de inteligencia artificial a integrar.
- Entrenar a la inteligencia artificial con información relacionada de la Universidad del Valle y el Departamento de Ingeniería Electrónica, Mecatrónica y Biomédica.
- Mantener una conversación con la inteligencia artificial relacionada a la Universidad del Valle y el Departamento de Ingeniería Electrónica, Mecatrónica y Biomédica.

En los alcances de esta tesis se aborda un programa para el desarrollo de un sistema interactivo que combina la detección de emociones y la generación de respuestas coherentes relacionadas con la Universidad del Valle de Guatemala, mediante inteligencia artificial. Este sistema tiene como propósito futuro integrarse a un rostro animatrónico, habilitando la posibilidad de que el usuario final mantenga una conversación personal, acompañada de movimientos mecánicos y respuestas adaptadas a sus interacciones.

Uno de los objetivos principales de la unificación de estos programas es utilizar esta tecnología como una herramienta interactiva de mercadeo, atrayendo a nuevos estudiantes y promoviendo los proyectos de innovación desarrollados por el Departamento de Ingeniería Mecatrónica, Electrónica y Biomédica. El proyecto se enfoca en entrenar el sistema de inteligencia artificial con información relacionada con la institución, incluyendo datos generales y detalles del Departamento de Ingeniería, siempre considerando el estado de ánimo del usuario final y ofreciendo una experiencia amigable.

Este trabajo de graduación sienta las bases para el futuro, abriendo camino a nuevos proyectos de robótica con aplicaciones en inteligencia artificial. Además, resalta la unión de diversas disciplinas como mecánica, electrónica, diseño y programación, mostrando el impacto de estas colaboraciones en el desarrollo tecnológico e innovador.

El rostro animatrónico está operativo, pero presenta deficiencias al proporcionar respuestas. Es crucial para la demostración que el programa de inteligencia artificial esté lo suficientemente capacitado para mantener una conversación fluida con el usuario final. Esto se logra mediante un proceso de entrenamiento en el cual se realizan múltiples iteraciones para que el sistema pueda almacenar y generalizar el tipo de respuestas que debe ofrecer.

El funcionamiento del rostro animatrónico será esencial para cumplir con el objetivo de la Universidad, captando la atención de los visitantes.

Durante el desarrollo de esta tesis, se integrarán diversos programas con las siguientes funciones: detectar el estado de ánimo del usuario final y establecer una conexión con ChatGPT para generar respuestas adecuadas, así como iterar con la inteligencia artificial para enriquecer la información contenida en su red neuronal.

Al momento de que el rostro animatrónico efectúa la respuesta hablada lo hace a través de una bocina, se debe de comprobar que esta funcione correctamente, y la calidad del audio transmitido es entendible.

La inteligencia artificial

Según James Fetzer, lo fantástico de la inteligencia artificial es la dificultad que tiene para definir, debido a sus constantes cambios y sus grandes embarcaciones a lo largo de todos los temas. El nombre cuenta con la palabra “artificial”, debido a que las soluciones brindadas cuentan con propiedades especiales, ya que fueron fabricadas, manufacturadas, o creadas de esta manera. La palabra “inteligencia” proviene de la capacidad para resolver problemas. Naciendo de la necesidad de resolver problemas de una manera más eficaz y eficiente, eliminando el error del factor humano. Utilizando sistemas computacionales para entender el pensamiento humano[7].

Los sentimientos limitan a los humanos para poder pensar o aprender con claridad, siendo

un factor de suma importancia a la hora de realizar la comparación entre humano-máquina. A la hora de eliminar el factor sentimiento, no existen distracciones, únicamente siguiendo órdenes dentro de un sistema computacional en el caso de un programa computacional.

Naciendo de la revolución industrial, muchos mecanismos manuales fueron sustituidos por sistemas tecnológicos. Así utilizando la misma lógica con la inteligencia artificial, sustituir, facilitar, agilizar los procesos humanos[8].

Machine learning

El aprendizaje automático es una rama de los algoritmos computacionales que son diseñados para emular la inteligencia humana a través del aprendizaje automático presente en los procesos del día a día. Distintas técnicas basadas en el aprendizaje automático han sido aplicadas en distintas áreas, tales como reconocimiento de patrones, visión computacional, finanzas, entretenimiento, aplicaciones médicas y muchas más.

El aprendizaje se maneja a través de precisión, y para eso depende de sus números de iteraciones o de la cantidad de datos dados a la máquina. Luego de que obtiene los datos el programa, este detecta los posibles patrones. Cuando entiende los patrones, tiene la posibilidad de sugerir predicciones. Dependiendo de la aplicación se dan la libertad de tomar las decisiones o únicamente sugerir; para asegurar su alta efectividad a través de entrenamiento.

El cerebro humano funciona gracias a las neuronas, así recibiendo distintos estímulos, los cuales son recibidos como impulsos nerviosos, pasados a través de las distintas neuronas (lado izquierda de la imagen). En la red neuronal, se observa como va descartando opciones, o *estímulos*, hasta poder llegar a una respuesta clara, sin tener variación[9].

Redes neuronales

Las redes neuronales artificiales son redes interconectadas masivamente en paralelo de elementos simples y con organización jerárquica, las cuales intentan interactuar con los objetos del mundo real del mismo modo que el sistema nervioso biológico. Está siendo una nueva forma de computación, conformada por modelos matemáticos, los cuales cuentan con un gran número de elementos procesales organizados en niveles. Así procesando por medio de su estado dinámico como respuesta a entradas externas.

La meta de una red neuronal es ser lo más parecido posible al cerebro humano. Las ventajas que ofrece una red neuronal son las siguientes: aprendizaje adaptativo, el cual es la capacidad de aprender a realizar tareas basadas en un entrenamiento o un ejemplo. La autoorganización, es la posibilidad de organizar las ideas dadas en un proceso de aprendizaje de forma autónoma. La tolerancia a fallos, aunque la red se someta a daños o a una destrucción parcial de una red, esta acorta la vida útil pero puede continuar con sus funciones, dependiendo del daño. Operación en tiempo real, pueden ser sometidos a cómputos neuronales en paralelo mientras trabajan y seguir funcionando. Fácil inserción dentro de la tecnología existente, tiene una gran flexibilidad para poder adaptar a distintos programas de la actualidad. Así recibiendo distintos chips, los cuales mejorarán su capacidad en ciertas

tareas, así facilitando la integración modular[10].

Chat GPT

Chat GPT es un sistema de chat, el cual cuenta con un sistema llamado modelo de lenguaje por inteligencia artificial GPT-3.5, desarrollado por la empresa OpenAI. Así teniendo redes neuronales, con más de 175 millones de parámetros y ha sido entrenado con grandes cantidades de texto para realizar tareas relacionadas con el lenguaje.

Para realizar la conexión con este sistema se tiene que realizar a través de una interfaz de programación de aplicaciones (API), las cuales son direcciones que permiten a diferentes aplicaciones comunicarse entre sí y compartir información y funcionalidades.

Cuando el programa propio ya se encuentra conectado con Chat GPT, este puede utilizar sus distintas funciones, en este caso se utilizó la parte del chatbot con inteligencia artificial. Con el fin de poder realizar conversaciones fluidas, con respuestas generadas por el programa y no con respuestas definidas.

Las API de Chat GPT son públicas, se pueden encontrar de forma gratuita en su página web. Su forma de cobrar es a través de tokens, los cuales se compran por paquetes, y estos tienen como significado de grupos de caracteres que representan la unidad fundamental del texto, es decir que los tokens son medidos a través de las respuestas dadas por el programa (Chat GPT). Estos son generados por un algoritmo tokenizador que segrega el texto en segmentos más pequeños siguiendo ciertas reglas, como espacios, y otras reglas de puntuación[11].

Unión entre la mecánica y la inteligencia artificial

La inteligencia artificial y la mecánica se fundamentan en múltiples definiciones, así pudiendo realizar los diferentes proyectos/trabajos de una manera más completa. La mecánica arma las máquinas, estructuras, vehículos, objetos, entre más. Y la inteligencia artificial la complementa, a través de diferentes programas; es decir revisa o realiza los planos, ayuda a la simulación, realiza el mantenimiento, o a través del *machine learning* educa y controla a la máquina para realizar el patrón o las instrucciones requeridas.

Python

Python es un lenguaje de programación de código abierto, creado por Guido van Rossum en 1991. Es un lenguaje orientado a objetos, fácil de interpretar y con una sintaxis amigable, ya que es relativamente parecido al inglés. El código en Python se convierte en *bytecode* y luego es ejecutado por la máquina virtual de Python.

A lo largo de los años, Python ha ganado gran popularidad debido a su facilidad de aprendizaje y comprensión. Es un lenguaje de propósito general, lo que permite su uso en una amplia variedad de aplicaciones, a diferencia de otros lenguajes que están más especializados, como aquellos enfocados únicamente en el desarrollo de páginas web. Esta flexibilidad

permite que Python se adapte a múltiples necesidades en la industria.

Python cuenta con comunidades activas, las cuales contribuyen constantemente hacia el aprendizaje. Estas comunidades creando herramientas, librerías y bibliotecas que expanden las capacidades del lenguaje. Tales como Flask, la cual se verá a continuación[12].

Interfaz Gráfica

La interfaz gráfica también conocida como GUI, el cual es un programa informático, el cual funciona para que los usuarios finales interactúen con el respectivo programa. Así siendo amigable para la vista del usuario final, contando con poco texto, imágenes, objetos gráficos y estética correcta[13].

Open CV

La Open Source Computer Vision Library (OpenCV) es una biblioteca de visión por computadora de código abierto diseñada para el público en general. Fue creada con el propósito de proporcionar una infraestructura sólida para el desarrollo de aplicaciones de visión por computadora, facilitando y acelerando la integración de la percepción de máquinas en productos comerciales. OpenCV permite modificar fácilmente el código fuente, lo que fomenta la experimentación y personalización. Actualmente, cuenta con más de 2500 algoritmos optimizados que pueden utilizarse para diversas tareas, como la detección y el reconocimiento de rostros y objetos, la clasificación de elementos o acciones en videos o transmisiones en vivo, la extracción de objetos en tres dimensiones, la transformación de imágenes a alta resolución, la comparación de imágenes, la detección de colores, el seguimiento de objetos, entre muchas otras funcionalidades.

En este proyecto específico, OpenCV se utilizó para la detección de emociones en tiempo real. Este proceso funciona activando la cámara del dispositivo donde se está ejecutando el software, para luego detectar el rostro de la persona. Para identificar las emociones, se definen rangos específicos que describen el movimiento de ciertas partes del rostro humano. El programa se enfoca en puntos clave del rostro y, con base en los cambios observados en estos puntos, determina la emoción de la persona. Es esencial crear y definir estos rangos de manera adecuada, ya que la precisión en la detección de emociones puede variar dependiendo de las características faciales individuales de cada usuario.

OpenCV es una herramienta extremadamente versátil que puede ser utilizada con diferentes lenguajes de programación, como Python, C++, Java, y Matlab. En este caso particular, se implementó en Python para definir cinco emociones: felicidad, sorpresa, tristeza, miedo y enojo. Cada emoción tiene sus propios rangos definidos para mejorar la precisión en la detección[14].

Flask

Flask es un micro *framework* desarrollado en Python y diseñado para facilitar el desarrollo de aplicaciones web siguiendo el patrón MVC. El patrón MVC (modelo-vista-controlador) es una forma de organizar el código que separa la lógica de la interfaz de usuario y el acceso a la base de datos. Para utilizar Flask, es necesario instalarlo previamente, lo que proporcionará todas las herramientas necesarias para crear una aplicación web funcional.

Flask es conocido por su facilidad para extender proyectos, es decir, cuando se desea ampliar el alcance de las aplicaciones web. Así puede crecer de forma orgánica el proyecto. Debido a su facilidad de uso, cuenta con muchas librerías y comunidades, las cuales hacen más sencillo el aprendizaje[15].

Json

Json es un tipo de archivo utilizado para representar, almacenar, acceder intercambiar y manipular información, para que un diccionario en Python lo pueda llamar y así obtener esta información. Esto formando parte del sistema JavaScript, utilizando la misma sintaxis[12].

Se utilizó este tipo de archivo para poder almacenar toda la información de la Universidad del Valle de Guatemala. Así, este documento se le entrega a la inteligencia artificial y este lo almacena[16].

SocketIO

Socket.IO es una biblioteca que proporciona comunicación en tiempo real entre el servidor y los clientes, permitiendo conexiones bidireccionales con respuestas inmediatas. Es ideal para aplicaciones que requieren el envío de datos de forma continua e instantánea, como mensajería instantánea o notificaciones en tiempo real[17].

Socket.IO funciona principalmente a través del protocolo WebSocket, pero también puede utilizar otros medios de comunicación, como HTTP o AJAX. Esta flexibilidad en los métodos de transporte hace que la herramienta sea más amigable para los desarrolladores y garantiza la conectividad en diversos entornos.

Además, Socket.IO ofrece una API robusta que permite utilizar la herramienta de manera personalizada, incluyendo funciones como la reconexión automática. En caso de desconexión, Socket.IO detecta la falta de comunicación y se encarga de restablecer la conexión de forma inmediata, recuperando los datos perdidos. Esto asegura una comunicación confiable y continua durante la transmisión de datos.

Gracias a su facilidad de uso y a la confiabilidad que ofrece en la transmisión de datos en tiempo real, Socket.IO es ampliamente reconocido y utilizado en el desarrollo de aplicaciones web y móviles que requieren comunicación en vivo[18].

HTML

El lenguaje de marcado de hipertexto o mejor conocido como HTML es el código que se utiliza para estructurar y desplegar una página web y sus contenidos. Es considerado como lenguaje de marcado que define la estructura de tu contenido. Así se utiliza para la parte del diseño del proyecto[19].

Este tiene un papel importante en la creación de páginas web, ya que es en donde se define el diseño de esta página. Así se definen elementos tales como los encabezados, los colores, las letras y otros componentes adicionales. Este va de la mano de Javascript, para que este le de la interacción.

Es de las tecnologías más utilizadas en proyectos de páginas web, debido a su facilidad de uso y su alta comunidad de personas que lo utilizan. Así, es muy amigable a la hora de buscar referencias[20].

API

Una interfaz de programación de aplicaciones (API) es un conjunto de funciones y protocolos que permite la comunicación entre diferentes aplicaciones, actuando como un contrato de servicio entre dos o más sistemas. Las API se utilizan para conectar un programa con otro, facilitando la integración y el intercambio de datos entre aplicaciones. Por ejemplo, para implementar un chatbot con inteligencia artificial en una página web, se debe utilizar una API que permita la interacción entre el *chatbot* y la plataforma web, habilitando así esta funcionalidad[21].

Existen distintos tipos de API's. Las cuales son las siguientes:

API pública

Son las API que están abiertas al público para que los clientes que deseen utilizarlas puedan hacerlo. Normalmente, se mantienen públicas porque generan ingresos recurrentes; de esta manera, cualquier persona puede utilizarlas sin que la compañía tenga que vender el producto directamente a cada usuario.

OpenAI, por ejemplo, ofrece sus API públicas para que los desarrolladores se puedan conectar a sus productos, como ChatGPT-4. Así, cada vez que el chatbot de inteligencia artificial genera una respuesta, OpenAI cobra al usuario por el uso del servicio[22].

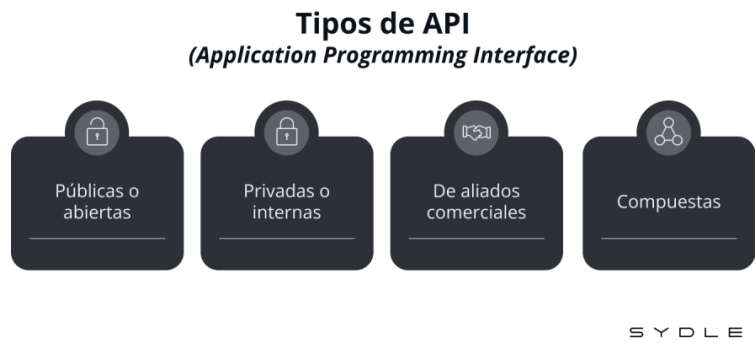


Figura 3: Tipos de API

API privada

Son las API que se utilizan internamente dentro de la empresa propietaria, con el objetivo de permitir que todos los sistemas y empleados se conecten de forma interna, sin que nadie más pueda acceder a esos beneficios. Esto ayuda a optimizar la productividad y la reutilización de servicios.

Un ejemplo de esto es CamiAPP, una herramienta financiera en constante innovación y adaptación al mercado financiero. Dado que muchas personas trabajan simultáneamente realizando cambios y mejoras, una API privada permite que todos se conecten al sistema al mismo tiempo, evitando la necesidad de esperar a que una tarea se complete antes de comenzar otra[23].

API de aliados comerciales

Las API de aliados comerciales son las que se comparten entre compañías aliadas o hermanas, con el fin de que éstas puedan utilizar un programa, un beneficio, a través de su API. Así siendo una API privada, pero expuesta con las compañías deseadas. Así se facilita el trabajo y la integración de la herramienta.

El Banco Industrial de Guatemala tiene API expuestos para aliados comerciales para que así puedan obtener ciertos beneficios, tales como pasarelas de pago, servicios de facturación o pago a cuotas; principalmente para comercios electrónicos[24].

API compuestas

Las API compuestas utilizan distintos datos o diversas API de servicio, así los desarrolladores acceden a todos los proyectos a través de uno[25].

Open AI *tokens*

Los *tokens* para la empresa OpenAI pueden considerarse como unidades que representan caracteres, los cuales componen las respuestas generadas por su herramienta de inteligencia artificial. Estos caracteres incluyen letras, números, signos de puntuación y otros símbolos.

El funcionamiento es el siguiente: primero, se establece una conexión con la API de la herramienta de OpenAI que se desea utilizar y se adquiere una cierta cantidad de *tokens*. Una vez obtenidos los *tokens*, se puede acceder a la herramienta seleccionada.

La extensión más utilizada es ChatGPT, un chatbot que genera respuestas a las consultas del usuario final. Cada respuesta del chatbot incluye múltiples caracteres, y el uso de *tokens* permite medir y gestionar el costo asociado a estas respuestas. En términos generales, cada mil *tokens* equivale a un promedio de 750 palabras[26].

Preparación y gestión de proyectos con librerías y almacenamiento de datos

Según Virage Group, la planificación y el entendimiento de un proyecto pueden determinar si este será exitoso o estará condenado al fracaso. Con este enfoque, se decidió abordar este proyecto, comenzando con una planificación detallada sobre la forma y los pasos a seguir, así como una extensa etapa de aprendizaje para comprender los proyectos anteriores, que serían esenciales para desarrollar una página web capaz de tener una conversación con una inteligencia artificial[27].

Para ello, se utilizaron los trabajos de graduación titulados *Implementación de una interfaz de uso y control para el rostro animatrónico de la Universidad del Valle de Guatemala* e *Implementación de una inteligencia artificial capaz de crear una conversación en la plataforma del rostro animatrónico*. Se revisaron estos trabajos previos desde una perspectiva teórica, con el fin de implementarlos de forma práctica.

7.1. Instalación de librerías

Para el correcto funcionamiento de los proyectos mencionados anterior, se tuvieron que instalar distintas librerías , las cuales serán divididas según el proyecto.

7.1.1. Librerías de implementación de una interfaz de uso y control para el rostro animatrónico de la Universidad del Valle de Guatemala

La primera librería instalada fue json, que habilita el uso de objetos JavaScript Object Notation (JSON), facilitando el envío y recepción de datos de manera rápida y precisa.

La segunda librería instalada fue cv2 (OpenCV), que se utiliza para trabajar con imáge-

nes en python. Esta librería es fundamental para la detección de emociones, ya que permite capturar y procesar imágenes a través de la cámara del dispositivo, identificando las emociones del usuario en tiempo real.

Al instalar python en el ordenador, es importante asegurarse de descargar todos los paquetes predeterminados, ya que esto garantiza que muchas herramientas asociadas a python estén disponibles de inmediato. Se recomienda utilizar estos proyectos en sistemas operativos Windows, ya que muchos permisos de python y Windows funcionan correctamente juntos, lo que ahorra tiempo al evitar las restricciones de permisos que pueden presentarse en otros sistemas operativos.

7.1.2. Librerías de implementación de una inteligencia artificial capaz de crear una conversación en la plataforma del rostro animatrónico

La primera librería instalada fue langchain, cuya función es simplificar la creación de aplicaciones que utilizan modelos de lenguaje grandes. Esta librería se instaló a través de la herramienta de extensiones de Visual Studio Code.

La segunda librería instalada fue tkinter, que se utiliza para el desarrollo y creación de aplicaciones de escritorio, siendo compatible únicamente con el lenguaje python.

Al instalar python en el ordenador, es importante asegurarse de descargar todos los paquetes predeterminados, ya que esto permite que muchas herramientas asociadas a python estén disponibles de inmediato. Se recomienda utilizar estos proyectos en sistemas operativos Windows, ya que muchos permisos de python y Windows funcionan correctamente juntos, lo que ahorra tiempo al evitar restricciones de permisos que pueden presentarse en otros sistemas operativos.

7.2. Análisis de proyectos

Para el análisis de los proyectos se realizaron las siguientes acciones:

7.2.1. Lectura trabajo escrito

Como primer paso, se revisaron los trabajos escritos, lo que permitió comprender tanto los objetivos de los proyectos como sus diferentes resultados. Esta revisión fue fundamental para obtener una visión general y teórica de todo el panorama, facilitando así la transición al proceso práctico.

Las diversas figuras encontradas en los trabajos resultaron de gran ayuda, ya que, al no haber experiencia previa en la parte práctica, era complicado entender completamente los resultados.

7.2.2. Entendimiento de código

En esta etapa, se realizó una lectura detallada, investigación y comprensión de los códigos existentes, lo cual fue fundamental para entender la parte práctica de los proyectos. Este proceso facilitó la descarga adecuada de las librerías necesarias y, principalmente, garantizó el correcto funcionamiento de los proyectos.

La comprensión profunda del código fue esencial para iniciar la integración de los proyectos, ya que se requería conocer a fondo el desarrollo que se iba a combinar. Este entendimiento también era crucial para asegurar que todos los componentes funcionaran correctamente.

7.2.3. Entendimiento de funcionamiento


En esta etapa, se procedió a la implementación de los proyectos que debían integrarse, analizando cuidadosamente los objetivos, las tareas, los problemas, las funciones, entre otros aspectos clave. Se hizo un énfasis especial en identificar y abordar los problemas que surgieron, ya que estos podían representar grandes inconvenientes durante el proceso de integración.

Gracias a la comprensión previa del código, la corrección de los problemas presentados fue menos complicada. Asimismo, esto permitió una mejor comprensión de las funcionalidades específicas de cada parte del proyecto y lo que cada función debía realizar.

7.3. Correcto funcionamiento de proyecto de detección de emociones

Para ejecutar correctamente el proyecto de detección de emociones, es necesario abrir una terminal y escribir el siguiente comando: Python Main.py, como se muestra en la Figura 4. Es fundamental asegurarse de que la terminal esté ubicada en la misma carpeta donde se encuentra el código del proyecto.

Al ejecutar este comando, el proyecto se iniciará de forma adecuada, abriendo una nueva ventana con la interfaz del proyecto.



```
PS C:\Users\juane\OneDrive\Desktop\TESIS\TESIS\Protocolo\TESIS ANTERIORES\Jorge Lanza\RostroAnmatronico2023-main\RostroAnmatronico2023-main> python main.py
```

Figura 4: Comando ingresado

7.3.1. Interfaz de uso

Deberá aparecer una interfaz como se muestra en la Figura 5:

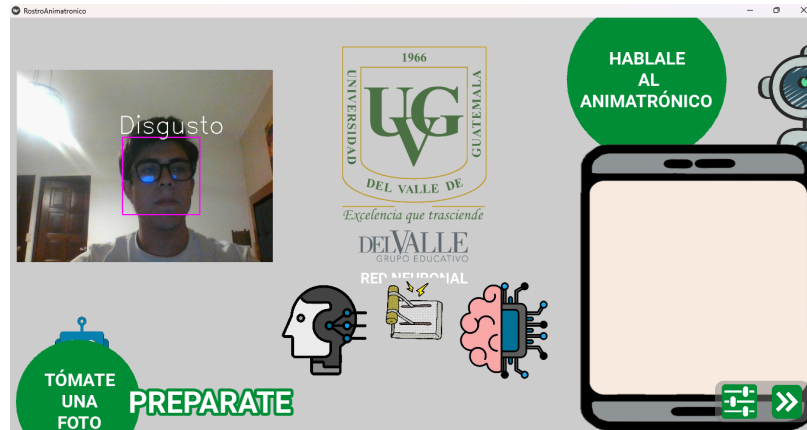


Figura 5: Interfaz de uso

Como se observa se cuenta con la cámara encendida del dispositivo, esto se hace de forma directa e inmediata.

7.3.2. Correcto funcionamiento de detección de emociones

Con la cámara prendida se realizarán distintas emociones, con el fin de que el programa pueda detectarlas y comprobar su correcto funcionamiento. A continuación se observarán distintas emociones puestas a prueba.

Tristeza

Se observa que, al mover ciertas partes del rostro para simular una emoción, el sistema de detección responde correctamente. Por ejemplo, se puede verificar que la emoción de tristeza es identificada de manera precisa.

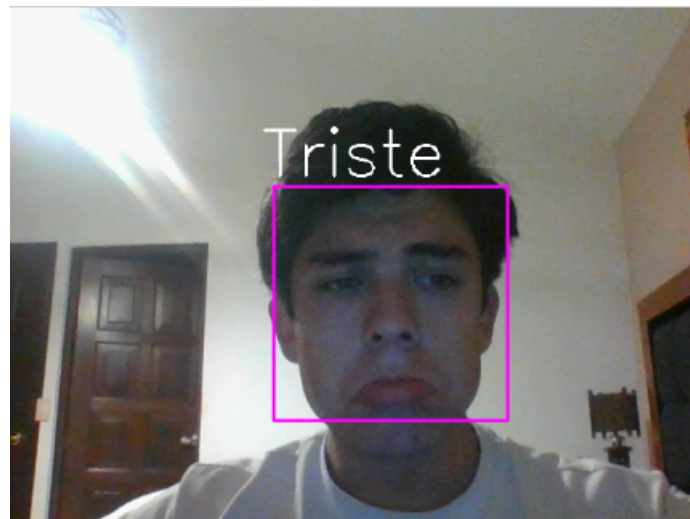


Figura 6: Emoción de tristeza detectada

Alegría

Se observa que, al mover ciertas partes del rostro para simular una emoción, el sistema de detección responde correctamente. Por ejemplo, se puede verificar que la emoción de alegría es identificada de manera precisa.



Figura 7: Emoción de alegría detectada

Miedo

Se observa que, al mover ciertas partes del rostro para simular una emoción, el sistema de detección responde correctamente. Por ejemplo, se puede verificar que la emoción de miedo es identificada de manera precisa.

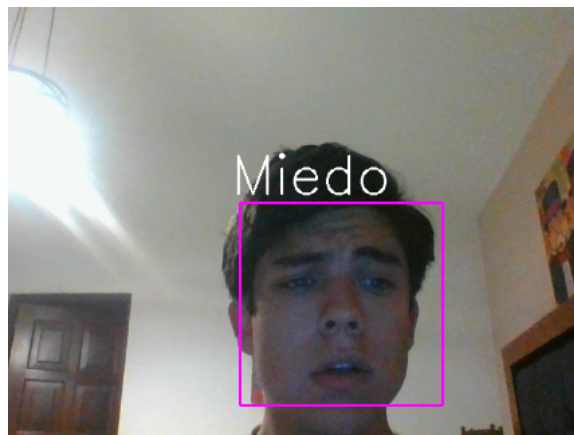


Figura 8: Emoción de miedo detectada

Disgusto

Se observa que, al mover ciertas partes del rostro para simular una emoción, el sistema de detección responde correctamente. Por ejemplo, se puede verificar que la emoción de

disgusto es identificada de manera precisa.



Figura 9: Emoción de disgusto detectada

Sorpresa

Se observa que, al mover ciertas partes del rostro para simular una emoción, el sistema de detección responde correctamente. Por ejemplo, se puede verificar que la emoción de sorpresa es identificada de manera precisa.



Figura 10: Emoción de sorpresa detectada

Enojo

Se observa que, al mover ciertas partes del rostro para simular una emoción, el sistema de detección responde correctamente. Por ejemplo, se puede verificar que la emoción de enojo es identificada de manera precisa.

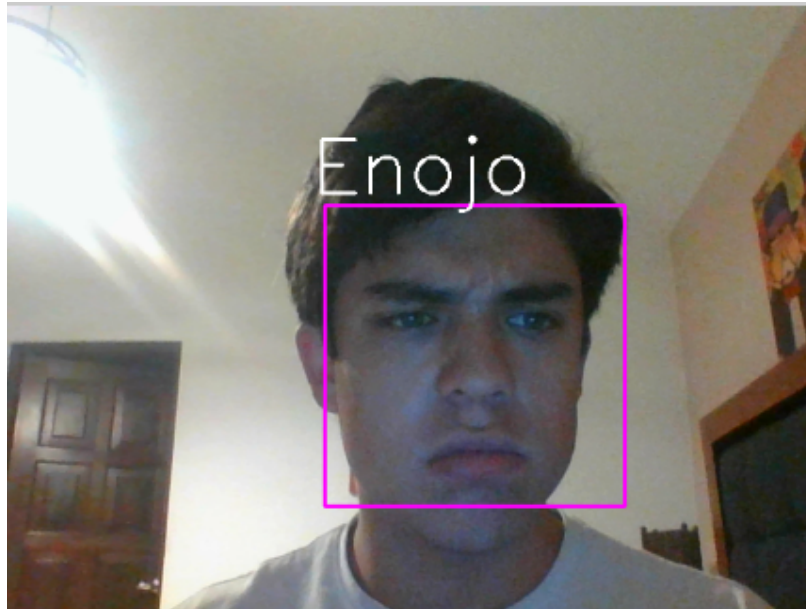


Figura 11: Emoción de enojo detectada

7.4. Correcto funcionamiento de proyecto de reentrenamiento

Este proyecto tenía varios objetivos por cumplir, aunque, para el propósito específico de esta implementación, los más relevantes fueron los objetivos tres y cuatro. Estos objetivos se centraban en reentrenar la inteligencia artificial utilizando un archivo y una interfaz, los cuales se describirán a continuación.

7.4.1. Interfaz de uso

La interfaz de uso para el re-entrenamiento de la inteligencia artificial fue la que se observa en la Figura 12, la cual únicamente tenía ese propósito. Como es el funcionamiento es que adentro de la casilla de texto, se debe de agregar la información que se desea almacenar, así luego seleccionando el botón *Terminar*, y así se estaría almacenando en un archivo .text

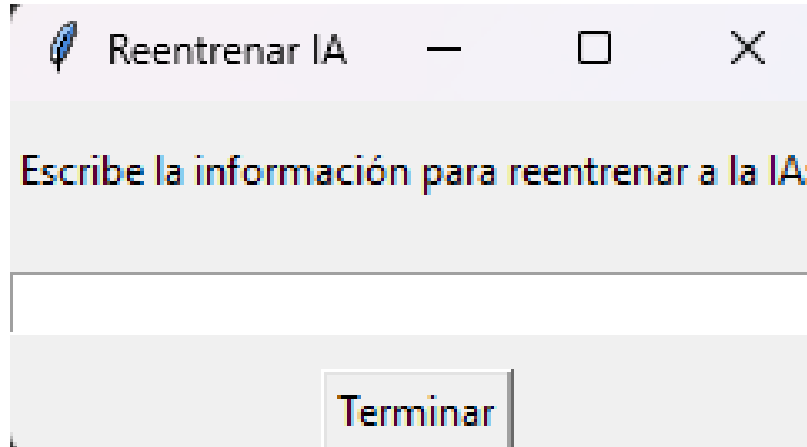


Figura 12: Terminal para ingresar texto

7.4.2. Almacenamiento en un archivo

El proceso para almacenar la información en un archivo es la siguiente:

Primer paso

En primer lugar, se debe escribir en la interfaz la información deseada. Para esta demostración, la información utilizada para el entrenamiento es: El rector de la Universidad del Valle es Roberto Moreno Godoy.

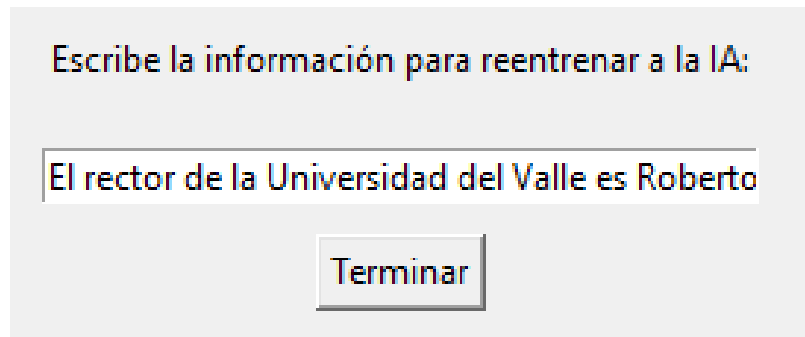


Figura 13: Información ingresada, prueba

Segundo paso

En el segundo paso, se debe seleccionar el botón *Terminar*. Al hacerlo, la interfaz se cerrará automáticamente y, dentro de la carpeta del proyecto, aparecerá un nuevo archivo llamado "data.txt", como se muestra a continuación.

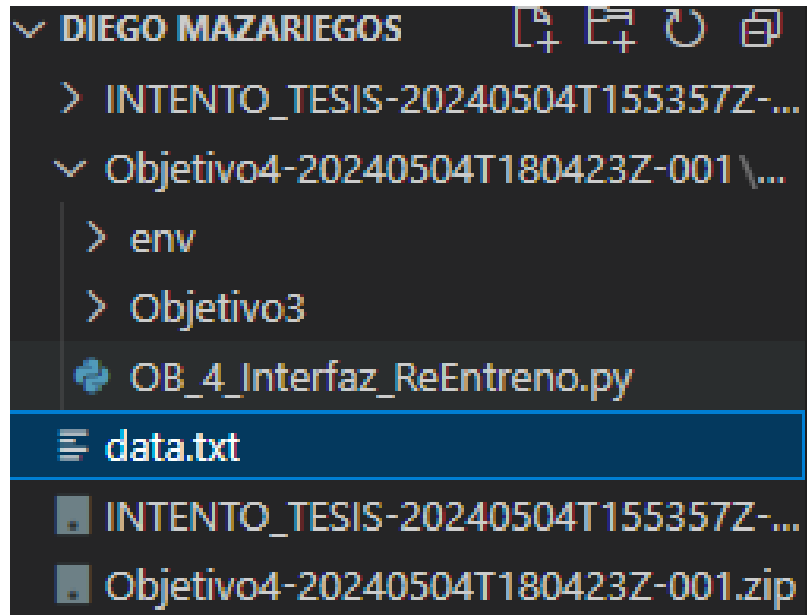


Figura 14: Archivo data.txt

Tercer paso

En el tercer paso, se selecciona el archivo "data.txt" para abrirlo y verificar que su contenido corresponde a la información que se introdujo previamente en la interfaz, como se muestra a continuación.

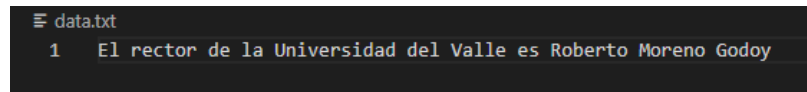


Figura 15: Contenido del archivo data.txt

Cuarto paso

En el último paso, el documento debe ser cargado en la red neuronal de la inteligencia artificial que se está utilizando. Esto proporcionará a la inteligencia artificial la información necesaria para comenzar a iterar y aprender.

7.4.3. Archivo tipo .text

El archivo *.txt* es un archivo de texto compatible con múltiples programas, lo que facilita su uso en diversas aplicaciones. Se eligió este formato por su amplia compatibilidad, permitiendo que pueda ser utilizado independientemente de la inteligencia artificial empleada.

Configuración y diseño de la integración de la API

Se utilizará una extensión de ChatGPT, desarrollada por OpenAI, como la inteligencia artificial para este proyecto. El objetivo es entrenarla con información específica de la Universidad del Valle, enfocándose en los Departamentos de Ingeniería Mecatrónica, Electrónica y Biomédica. Esta información se almacenará en un archivo .json, que se mostrará más adelante.

8.1. Creación de correo institucional

Para una mejor gestión, se creó un correo institucional, con el objetivo de que futuros estudiantes que deseen continuar trabajando en este proyecto de graduación tengan acceso a todos los permisos e información necesarios. Este correo está destinado exclusivamente para su uso y relación con OpenAI, así como para la adquisición de *tokens* asociados con la Universidad del Valle. El correo utilizado se muestra a continuación.

apichatgpt@wvg.edu.gt

8.2. Creación de API Key OpenAI

Se generó una clave de API completamente privada, que se utiliza para conectar la extensión personalizada de ChatGPT, asegurando que sea cien por ciento adaptativa a lo deseado y privada.

Es crucial que esta clave de API no se comparta con nadie, ya que, si alguien tuviera acceso a ella, podría emitir instrucciones, órdenes y proporcionar información para entrenar

la extensión personalizada. Por estas razones, la clave no será divulgada y se mantiene segura dentro del código del archivo principal llamado *App.py*.

8.3. Adquisición de *tokens* OpenAI

La extensión personalizada de OpenAI funciona a través de *tokens*, los cuales se necesitan adquirir a través de la siguiente página web:

<https://openai.com/api/pricing/>

Se adquirieron 40.00 dólares estadounidenses en *tokens*, lo que equivale a cuatro millones de *tokens*, aproximadamente tres millones de palabras. Para consultar el saldo restante de *tokens*, es necesario ingresar a la cuenta de OpenAI utilizando el correo institucional, y se verá el dinero en *tokens* disponible, tal como se muestra en la Figura 16.

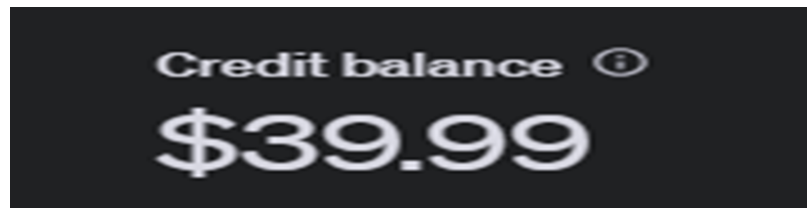


Figura 16: Dinero disponible en *tokens*

El enlace para poder acceder a los *tokens* disponibles es el siguiente:

<https://auth.openai.com/authorize>

8.4. Conexión con ChatGPT

A través de la clave de API (*API key*), se realizó la conexión inicial para verificar la comunicación con el servidor de OpenAI, enviando funciones que solicitaban respuestas de ChatGPT.

En la Figura 17, se puede observar un ejemplo en el que se envía la solicitud "Hola, ¿cómo estás?".^a la inteligencia artificial, esperando una respuesta del servidor.

```
# Prueba de uso

if __name__ == '__main__':
    prompt = "Hola, ¿cómo estás?"
    response = chat_with_server(prompt)
    print(f"Respuesta del servidor: {response}")
```

Figura 17: Verificación de comunicación con el servidor

Para confirmar la correcta conexión con el servidor de OpenAI, se utilizó FastAPI, un marco web para crear API de servicios basados en HTTP en python[28].

Para visualizar las respuestas generadas, se debe acceder al siguiente enlace, que funcionará únicamente si el código se está ejecutando:

<http://localhost:5000/chat>.

Al abrir este enlace, se muestra una página web que incluye la función que estamos utilizando (en este caso, `chat`), proporcionando detalles como los parámetros de la solicitud y un botón para probar la función. Al presionar este botón, en la parte inferior se mostrará la respuesta proporcionada por la inteligencia artificial.

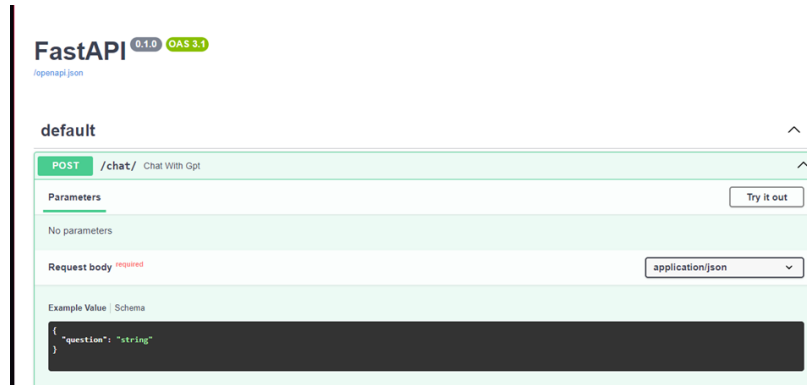


Figura 18: Página web de FastAPI

A través de esa página web, se puede verificar la correcta conexión con la extensión de OpenAI. FastAPI se utilizó inicialmente como un medio para observar el funcionamiento adecuado de la extensión de OpenAI; sin embargo, para el proyecto, se empleó directamente la biblioteca de OpenAI, utilizando *Flask* para mostrar los resultados en una página web.

8.5. Limitación de *tokens* para respuestas dadas

Al hacer una pregunta a ChatGPT, se puede observar que este programa tiende a proporcionar respuestas extensas, tratando de cubrir toda la información relevante. Sin embargo, en muchos casos, la respuesta podría ser más puntual y definirse en solo unas pocas palabras, como se muestra en la Figura 19.

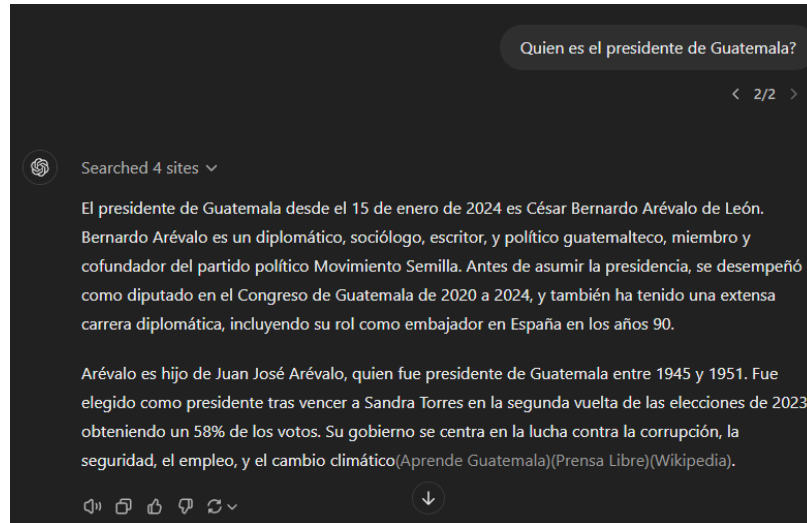


Figura 19: Prueba de respuesta en ChatGPT

Por ejemplo, se le hizo una pregunta puntual: "¿Quién es el presidente de Guatemala?" Sin embargo, el programa respondió de manera extensa, incluyendo información adicional sobre el pasado de esta persona y otros detalles.

Dado que cada respuesta generada por la extensión de OpenAI implica un consumo de *tokens*, se decidió limitar el número de *tokens* por respuesta para obligar a la inteligencia artificial a responder dentro de una cantidad máxima de caracteres. En este caso, se estableció un límite de 150 *tokens* por respuesta, optimizando así el uso de los *tokens*.

Para determinar esta limitación, fue necesario iterar múltiples veces para encontrar la cantidad adecuada de *tokens* que permitiera obtener respuestas correctas, completas y concisas. Finalmente, se definió un límite de 150 *tokens*, como se muestra en la Figura 20. La función utilizada para establecer este valor se llama *Max-Tokens*.

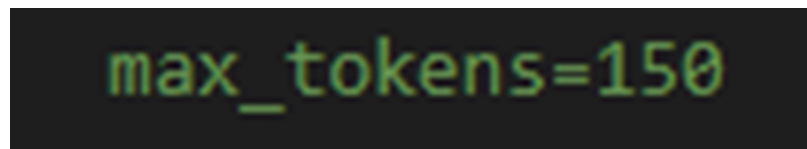


Figura 20: Max-tokens y su valor

8.6. Definición de roles a ChatGPT

Es necesario definir instrucciones iniciales para la extensión de OpenAI, proporcionándole una guía sobre cómo debe comportarse y asignándole un rol específico. En este caso, el rol asignado es el siguiente:

```
role": "system", content": ".Eres un chatbot digital universitario, amigable y capaz de detectar emociones para mejorar el estado de ánimo de los usuarios.", role": "user", content": prompt
```

Esto implica que, para cada decisión que la extensión tome al responder, deberá utilizar estas instrucciones como referencia. Estas instrucciones fueron proporcionadas para que las respuestas sean más personales, permitiendo al usuario final sentirse más cómodo y seguro al interactuar con el programa.

La detección de emociones es un aspecto crucial del proyecto. Por ello, se configuró la extensión de manera que, a través de las instrucciones y roles asignados, sepa que debe esperar una emoción por parte del programa. De esta manera, la emoción detectada se captura por separado, se almacena como una variable y se envía de inmediato a la extensión, para que esta la considere al generar sus respuestas.

8.7. Creación de diseño para interfaz gráfica

Para la creación de la interfaz gráfica, se diseñaron distintas plantillas con el objetivo de hacerla lo más amigable posible para el usuario final, optimizando cada detalle para mejorar la experiencia de usuario.

Para desarrollar estas plantillas, se utilizó un programa llamado Figma, un editor de gráficos vectoriales y una herramienta para la creación de prototipos, especialmente enfocado en diseños basados en páginas web. Figma está disponible para aplicaciones en Windows y MacOS.

8.7.1. Creación de menú

La Figura 21 muestra la página de inicio, con el menú ubicado en el lado izquierdo. En ella, se pueden observar las diferentes opciones disponibles del proyecto. Cabe destacar que estas plantillas fueron creadas de forma interactiva, asignando funciones específicas a cada botón general y a los elementos del menú.



Figura 21: Menú para el rostro animatrónico

8.7.2. Creación de educa a la inteligencia artificial

Cuando se selecciona el botón del menú que dice "Educa a la IA", se accede a una pestaña diseñada para ingresar la información con la que se educará a la extensión de OpenAI. Una vez que se escribe en la caja de texto, se debe presionar el botón "Educar", lo que simula el almacenamiento de la información en un archivo. La plantilla de "Educa a la IA" se muestra a continuación:

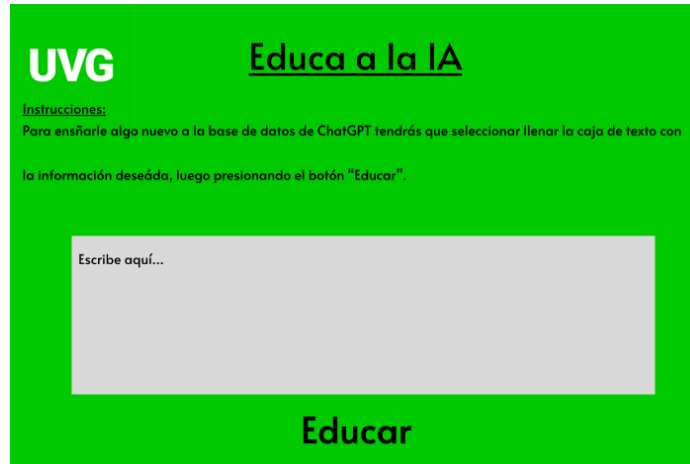


Figura 22: Pestaña donde se educa a la IA

8.7.3. Creación de detección de emociones

La siguiente pestaña creada fue la de "Detección de emociones", diseñada para activar la cámara en tiempo real, permitiendo que el programa detecte las emociones de manera precisa. Esto permite al usuario experimentar cómo el sistema identifica correctamente las emociones captadas. La interfaz de esta pestaña se muestra a continuación:



Figura 23: Pestaña de detección de emociones

8.7.4. Creación de información almacenada en la inteligencia artificial

La siguiente pestaña permite visualizar la información almacenada y con la que se ha entrenado a la inteligencia artificial. Además, ofrece la opción de descargar un archivo en este caso, un Excel que contiene toda la información utilizada en el entrenamiento. Al seleccionar el botón *Descargar*, el archivo se descargará automáticamente con la información solicitada. La interfaz de esta pestaña se presenta a continuación:

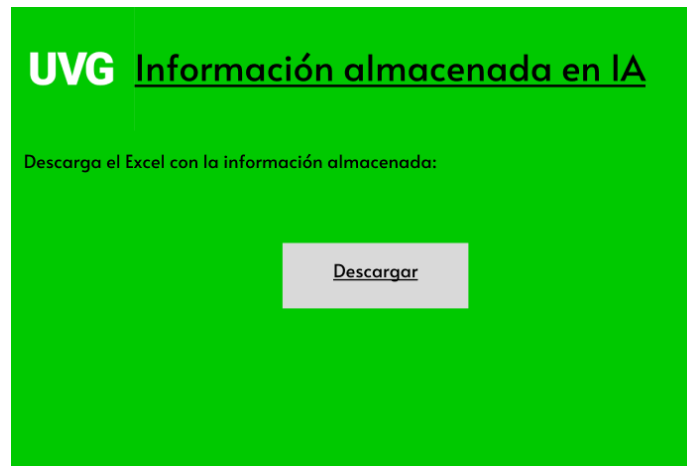


Figura 24: Pestaña donde se descarga la información almacenada

8.7.5. Creación de conversación en vivo

Para la creación de la conversación en vivo, se implementaron dos opciones: "Ver conversación en Chat" "Ver conversación hablada". Estas opciones brindan al usuario la posibilidad de elegir entre mantener una conversación completamente hablada o realizarla como si fuera una conversación por mensajes de texto. La interfaz se presenta de la siguiente manera:



Figura 25: Pestaña para la conversación en vivo

8.7.6. Creación de información extra

Para la última opción del menú, "Información Extra", no se creó una pestaña adicional. Al seleccionar este botón en el menú principal, se configura una re-dirección automática a la página web oficial de la Universidad del Valle. El objetivo es que los usuarios finales puedan obtener toda la información necesaria directamente de la fuente oficial de la universidad. La re-dirección funciona de la siguiente manera:



¿QUÉ ESTÁ PASANDO EN UVG?

Figura 26: Página web oficial de la Universidad del Valle

Esta es la última opción del menú en el prototipo de diseño de interfaz para el proyecto "Integración de programas para mantener una conversación con una inteligencia artificial". En su desarrollo, se han tenido en cuenta diversos factores, como asegurar una experiencia de usuario óptima y facilitar su uso.

Desarrollo de la plataforma web e integración de funcionalidades de inteligencia artificial

En esta etapa del proyecto, centrada en la implementación y el desarrollo, ya se había completado la fase de diseño y planificación. Por lo tanto, se comenzó con el desarrollo de la página web, incluyendo las distintas funciones que se describen a continuación:

9.1. Creación de página web

Para la creación de la página web, se comenzó utilizando Flask y SocketIO, que proporcionaron las bases para el desarrollo a través de las bibliotecas ya incluidas, todo ello desarrollado en el lenguaje python.

Se empleó HTML exclusivamente para el diseño estético de la interfaz, definiendo elementos como colores, tamaño y tipo de letra, entre otros aspectos de diseño. A continuación, se presenta el código del archivo principal, llamado ".App.py":

Código en python

```
from flask import Flask, render_template, request, jsonify, redirect,
    send_file
import openai
import json
from flask_socketio import SocketIO, emit

app = Flask(__name__)
socketio = SocketIO(app)
```

```

def load_json():
with open('uvg_departamento_ingenieria.json', encoding='utf-8') as f:
return json.load(f)

def save_json(data):
with open('uvg_departamento_ingenieria.json', 'w', encoding='utf-8') as f:
json.dump(data, f, ensure_ascii=False, indent=4)

openai.api_key = 'sk-proj-1rW1bTeKWWmbJKga3GRQT3B1bkFJl0KRjCrNbbilAtaleJuc'

@app.route('/')
def index():
return render_template('index.html')

@app.route('/check_connection')
def check_connection():
try:
response = openai.ChatCompletion.create(
model="gpt-3.5-turbo",
messages=[{"role": "system", "content": "Este es un mensaje de prueba."}]
)

upload_data()
return jsonify({'connected': True})
except Exception as e:
print(f"Error al conectar con OpenAI: {e}")
return jsonify({'connected': False})

@app.route('/upload_data', methods=['POST'])
def upload_data():

data_to_upload = load_json()

print("Datos subidos exitosamente")
return jsonify({'status': 'success'})

@app.route('/chat', methods=['POST'])
def chat():
uvg_data = load_json()
user_input = request.form['text']
emotion = request.form['emotion']
response = get_chatgpt_response(user_input, emotion, uvg_data)
return jsonify({'response': response})

def detect_emotions(frame):

pass

def get_chatgpt_response(text, emotion, uvg_data):
context = f"\n\nContexto adicional: {uvg_data['Universidad']['Nombre']} está
ubicada en {uvg_data['Universidad']['Ubicación']}. Fue fundada en {
uvg_data['Universidad']['Fundación']} y su lema es '{uvg_data['
Universidad']['Lema']}'. "

if "Informacion Adicional" in uvg_data:
additional_info = "\n".join(uvg_data["Informacion Adicional"])

```

```

context += f"\nInformación adicional: {additional_info}"

prompt = f"La emoción detectada es {emotion}. Responde de manera adecuada a:
        {text}{context}"

response = openai.ChatCompletion.create(
model="gpt-3.5-turbo",
messages=[
{"role": "system", "content": "Eres un chatbot digital universitario,
        amigable y capaz de detectar emociones para mejorar el estado de ánimo
        de los usuarios."},
{"role": "user", "content": prompt}
]
)
return response['choices'][0]['message']['content'].strip()

@socketio.on('message')
def handle_message(data):
    uvg_data = load_json()
    user_message = data['message']
    emotion = data['emotion']
    response = get_chatgpt_response(user_message, emotion, uvg_data)
    emit('response', {'message': response, 'emotion': emotion})

@app.route('/educa_ia', methods=['GET', 'POST'])
def educa_ia():
    uvg_data = load_json()
    if request.method == 'POST':
        new_info = request.form['new_info']
        if "Informacion Adicional" not in uvg_data:
            uvg_data["Informacion Adicional"] = []
            uvg_data["Informacion Adicional"].append(new_info)
        save_json(uvg_data)
    return render_template('educa_ia.html', success=True)
    return render_template('educa_ia.html', success=False)

@app.route('/deteccion_emociones')
def deteccion_emociones():
    return render_template('deteccion_emociones.html')

@app.route('/informacion_almacenada')
def informacion_almacenada():
    uvg_data = load_json()
    return render_template('informacion_almacenada.html', data=uvg_data)

@app.route('/descargar_informacion')
def descargar_informacion():
    return send_file('uvg_departamento_ingenieria.json', as_attachment=True)

@app.route('/informacion_extra')
def informacion_extra():
    return redirect("https://www.uvg.edu.gt")

@app.route('/conversacion_vivo')
def conversacion_vivo():
    return render_template('conversacion_vivo.html')

if __name__ == "__main__":
    socketio.run(app, debug=True)

```

Resultado práctico

Básicamente, este código se encarga de establecer la conexión con OpenAI, crear el archivo .json que almacenará toda la información utilizada para educar a la inteligencia artificial, y gestionar las diferentes llamadas a cada aplicación con sus respectivas funciones, correspondientes a las distintas pestañas que se estarán utilizando.

Con eso se obtuvo el siguiente resultado de la página web:

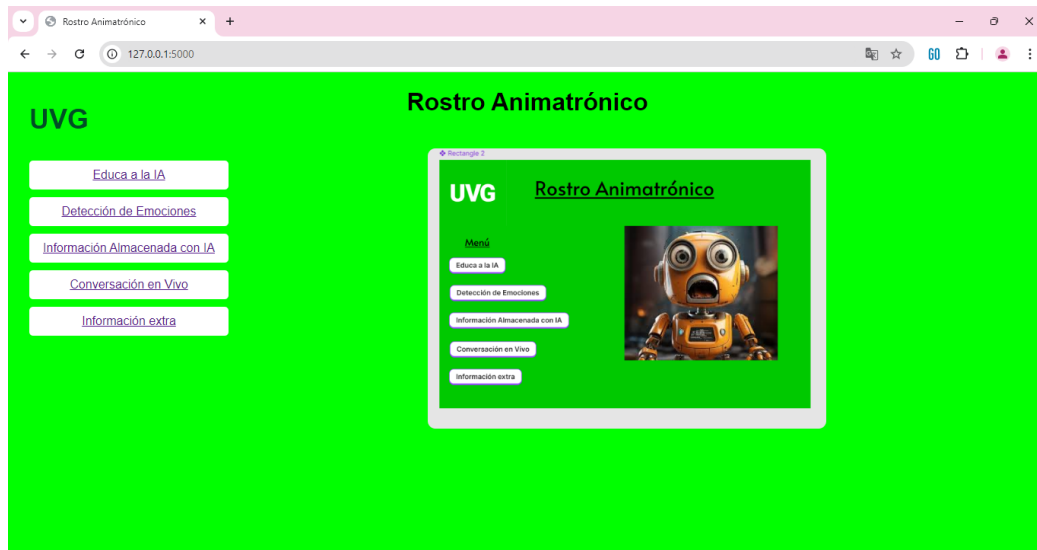


Figura 27: Página web rostro animatrónico

9.2. Creación de páginas aleadas

Se crearon distintas páginas en HTML, las cuales tienen sus distintas funciones. A continuación se observará cada una de ellas, mostrando la función, el código de desarrollo y como se ve presentado.

9.2.1. Educa a la inteligencia artificial

El propósito de esta parte del proyecto es permitir que se ingrese información en la caja de texto, de manera que la inteligencia artificial se mantenga continuamente educada y actualizada con los datos más recientes. El mecanismo funciona al seleccionar el botón "Guardar", que almacena la información proporcionada en un archivo .json. Este archivo centraliza toda la información con la que ha sido educada la extensión de OpenAI.

En el archivo HTML de "Educa a la Inteligencia Artificial", inicialmente se establecen

las instrucciones para definir los colores que aparecerán, así como el tamaño de cada botón y de la caja de texto. También se especifican ciertas características para los botones, como el estilo y la forma en que responden al ser presionados. A continuación, se incluye el logo de la Universidad del Valle de Guatemala y, posteriormente, se asignan las acciones correspondientes a cada botón.

Código en HTML

El código de este HTML es el siguiente:

```
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Educa a la IA</title>
<style>
body {
    background-color: #00FF00;
    font-family: Arial, sans-serif;
    color: #333;
    position: relative;
}
.container {
    max-width: 600px;
    margin: 50px auto;
    padding: 20px;
    background-color: white;
    border-radius: 10px;
    box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.1);
}
h1 {
    text-align: center;
    color: #333;
}
textarea {
    width: 100%;
    height: 100px;
    padding: 10px;
    border-radius: 5px;
    border: 1px solid #ccc;
    margin-bottom: 20px;
}
.buttons {
    display: flex;
    justify-content: space-between;
}
button {
    width: 48%;
    padding: 10px;
    background-color: #4CAF50;
    color: white;
    border: none;
    border-radius: 5px;
    cursor: pointer;
    font-size: 16px;
}
```

```

}
button:hover {
    background-color: #45a049;
}
.back-button {
    background-color: #f44336;
}
.back-button:hover {
    background-color: #d32f2f;
}
.success {
    color: green;
    text-align: center;
    margin-bottom: 20px;
}
/* Estilo para el texto UVG */
.logo-uvg {
    position: absolute;
    top: 20px;
    left: 20px;
    font-family: 'Arial', sans-serif;
    font-size: 36px;
    font-weight: bold;
    color: #004d26;
}
</style>
</head>
<body>
<div class="logo-uvg">UVG</div>

<div class="container">
<h1>Educa a la IA</h1>
{% if success %}
<p class="success">¡Información guardada exitosamente!</p>
{% endif %}
<form action="/educa_ia" method="POST">
<textarea name="new_info" placeholder="Escribe la nueva información aquí..."
    ></textarea>
<div class="buttons">
<button type="submit">Guardar</button>
<a href="/" style="text-decoration: none;">
<button type="button" class="back-button">Volver al Menú</button>
</a>
</div>
</form>
</div>
</body>
</html>

```

Resultado práctico

Desde el punto de vista del diseño, se presenta de la siguiente forma. A continuación, se mostrarán diversas figuras que ilustran los distintos procesos posibles dentro de la pestaña.

La Figura 28 muestra la pestaña de inicio en la sección *Educa a la IA*. Como se puede

observar, no hay ningún texto escrito en la caja de texto.



Figura 28: Pestaña donde se educa a la IA

En la Figura 29, se observa que la caja de texto ya contiene información relacionada con la Universidad del Valle de Guatemala, la cual será almacenada para la demostración.

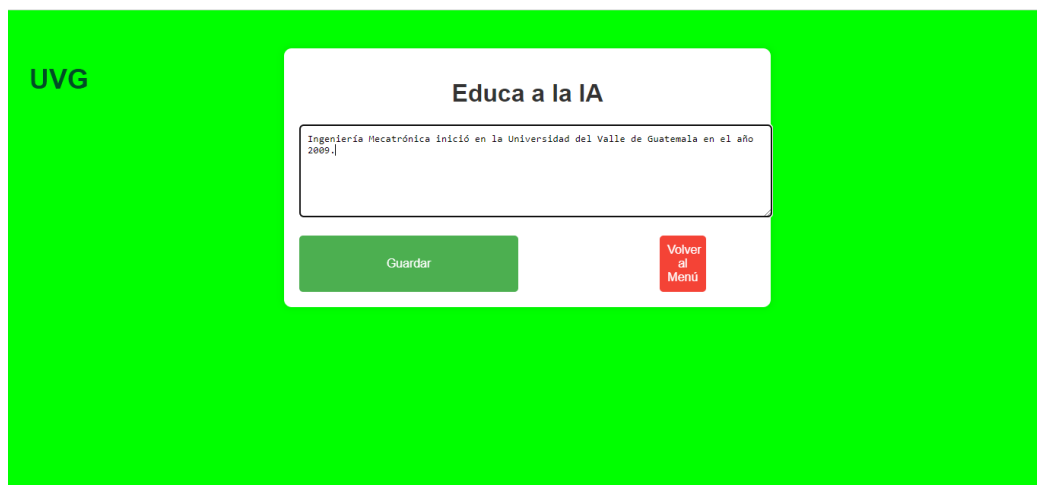


Figura 29: Pestaña donde se educa a la IA, texto ingresado

En la Figura 30, se muestra cómo, al seleccionar el botón *Guardar*, la página genera automáticamente un aviso de "¡Información guardada exitosamente!". Esto permite al usuario confirmar que la información ingresada ha sido almacenada correctamente en el archivo .json.



Figura 30: Pestaña donde se educa a la IA, información guardada

En la Figura 31 de esta sección, se muestra cómo, al seleccionar el botón *Volver al menú*, la página regresa automáticamente al menú principal.

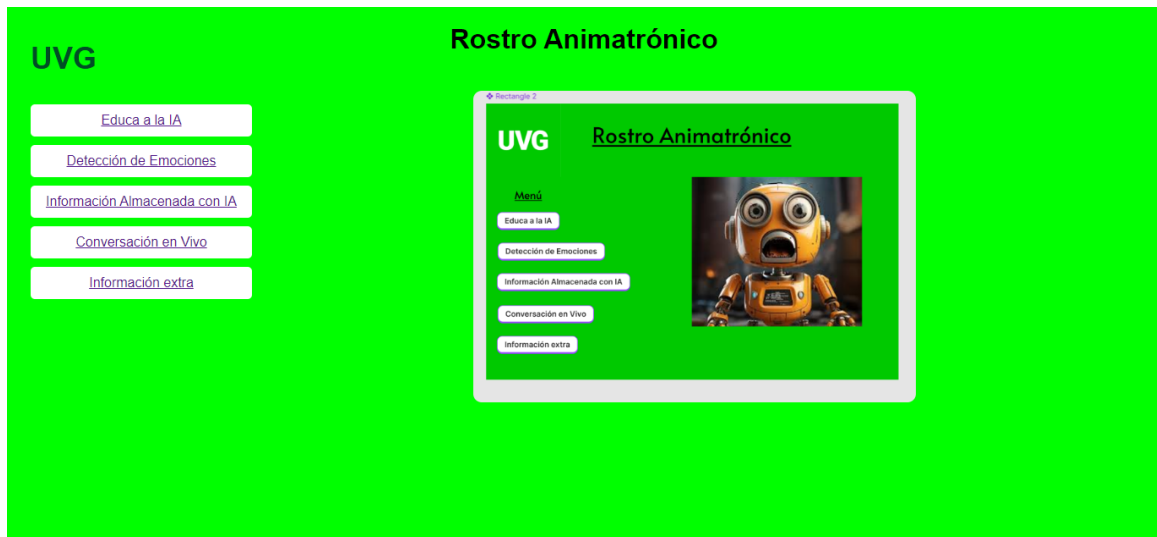


Figura 31: Página principal de rostro animatrónico

9.2.2. Detección de emociones

El propósito de esta parte del proyecto es hacer que la página web sea más interactiva para los usuarios finales, quienes podrían ser posibles estudiantes a los que se desea atraer mediante proyectos innovadores.

El funcionamiento de esta sección permite que, cuando el usuario elija esta opción en el menú principal, se active la cámara y comience a detectar emociones en tiempo real. El usuario final puede mover ciertos rasgos faciales para observar cómo el programa identifica y refleja los cambios de emoción.

En el archivo HTML, se definieron las selecciones de colores, fuentes y otros rasgos visuales. Se añadió un botón para activar la cámara, programando su comportamiento para que, al ser seleccionado, encienda la cámara automáticamente. También se crearon las emociones posibles que el sistema puede detectar: felicidad, sorpresa, tristeza, miedo y enojo. Para finalizar, se implementó programación defensiva, con el objetivo de mostrar avisos en caso de que no se pueda realizar alguna acción o activar la cámara correctamente.

Código en HTML

El código del archivo en HTML es:

```
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Detección de Emociones</title>
<style>
body {
    background-color: #00FF00;
    font-family: Arial, sans-serif;
    display: flex;
    flex-direction: column;
    justify-content: center;
    align-items: center;
    height: 100vh;
    margin: 0;
}
.container {
    width: 600px;
    background-color: #ffffff;
    padding: 20px;
    border-radius: 10px;
    box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.1);
    text-align: center;
    position: relative;
}
h1 {
    color: #333;
    margin-bottom: 20px;
}
video {
    width: 100%;
    border-radius: 10px;
    margin-bottom: 20px;
}
.emotion-display {
    margin-top: 20px;
    font-size: 24px;
    font-weight: bold;
    color: #333;
}
button {
    background-color: #4CAF50;
    border: none;
    color: white;
```

```

padding: 10px 20px;
text-align: center;
text-decoration: none;
display: inline-block;
font-size: 16px;
border-radius: 5px;
cursor: pointer;
}
button:hover {
background-color: #45a049;
}
.back-button {
margin-top: 10px;
background-color: #f44336;
border: none;
color: white;
padding: 10px 20px;
text-align: center;
text-decoration: none;
display: inline-block;
font-size: 16px;
border-radius: 5px;
cursor: pointer;
}
.back-button:hover {
background-color: #d32f2f;
}
</style>
</head>
<body>
<div class="container">
<h1>Detección de Emociones</h1>
<video id="video" autoplay></video>
<div id="emotion-display" class="emotion-display">Emoción: N/A</div>
<button onclick="startEmotionDetection()">Iniciar Detección</button>
<a href="/" class="back-button">Volver al Menú</a>
</div>

<script>
let videoStream;

function startEmotionDetection() {
const videoElement = document.getElementById('video');
const emotionDisplay = document.getElementById('emotion-display');

if (navigator.mediaDevices && navigator.mediaDevices.getUserMedia) {
navigator.mediaDevices.getUserMedia({ video: true }).then(
function(stream) {
videoStream = stream;
videoElement.srcObject = stream;
videoElement.play();

// Simulación de detección de emociones
setInterval(() => {
const emotions = ["Felicidad", "Sorpresa", "Tristeza", "Miedo", "Enojo"];
const detectedEmotion = emotions[Math.floor(Math.random() * emotions.length)];
emotionDisplay.innerText = 'Emoción: ${

```

```
                detectedEmotion}';
            }, 2000); // Cada 2 segundos
        }).catch(function(error) {
            console.error("Error al acceder a la cámara:",
                error);
            alert("No se pudo acceder a la cámara.");
        });
    } else {
        alert("La detección de emociones no es compatible con tu
            navegador.");
    }
}
</script>
</body>
</html>
```

Resultado práctico

En la Figura 32 no aparece nada ya que no se ha seleccionado el botón de *Iniciar detección*.



Figura 32: Pestaña detección de emociones

En la Figura 33 se observa como después de seleccionar el botón de *Iniciar detección*, este activa la cámara del dispositivo y empieza de forma inmediata a detectar emociones.



Figura 33: Detección de emociones, enojo

En la Figura 34 se muestra otra emoción detectada:



Figura 34: Detección de emociones, felicidad

En la Figura 35 se muestra otra emoción detectada:

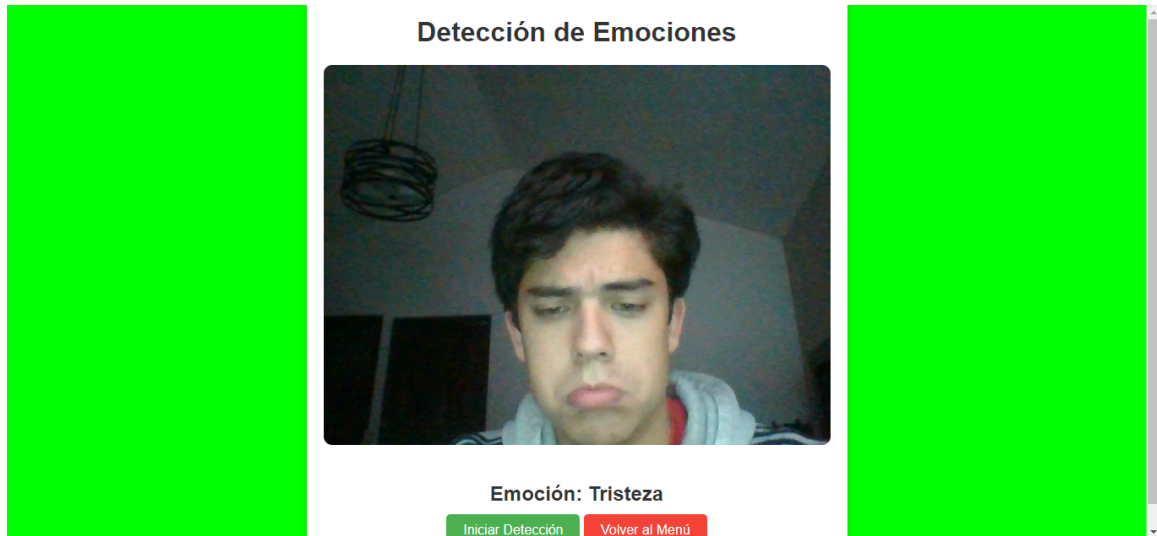


Figura 35: Detección de emociones, tristeza

Por último de esta sección, si el botón de *Volver a Menú* es seleccionado, la página regresa a la página de inicio, la cual es donde se encuentran el menú.



Figura 36: Pestaña principal

9.2.3. Información almacenada

El propósito de esta pestaña es permitir al usuario encontrar la información con la que se ha entrenado a la inteligencia artificial, así como acceder a la sección de información adicional que contiene los datos ingresados en la pestaña *Educa a la IA*.

La pestaña ofrece la opción de descargar el archivo .json para que el usuario final pueda obtener toda la información en su propio dispositivo.

En el código, se definen los colores, las fuentes y otros aspectos estéticos. Además, se

organiza la información en secciones para presentarla de manera visualmente atractiva. Sin esta clasificación, la información aparecería simplemente como está estructurada en el archivo .json.

Código en HTML

El código del archivo en HTML es:

```
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Información Almacenada</title>
<style>
body {
    background-color: #00FF00; /* Fondo verde */
    font-family: Arial, sans-serif;
    color: #333;
    position: relative;
}
.container {
    max-width: 800px;
    margin: 50px auto;
    padding: 20px;
    background-color: white;
    border-radius: 10px;
    box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.1);
}
h1 {
    text-align: center;
    color: #333;
}
.info-section {
    margin-bottom: 20px;
}
.info-section h2, .info-section h3, .info-section h4, .info-section
h5, .info-section h6, .info-section h7 {
    color: #333;
    margin-top: 15px;
    margin-bottom: 5px;
    padding-bottom: 5px;
}
.info-section p, .info-section ul {
    margin: 10px 0;
}
.info-section ul {
    list-style-type: disc;
    margin-left: 20px;
}
.buttons {
    text-align: center;
    margin-top: 20px;
}
button {
    padding: 10px 20px;
```

```

        background-color: #4CAF50;
        color: white;
        border: none;
        border-radius: 5px;
        cursor: pointer;
        font-size: 16px;
    }
    button:hover {
        background-color: #45a049;
    }
    .back-button {
        background-color: #f44336;
        margin-top: 10px;
    }
    .back-button:hover {
        background-color: #d32f2f;
    }
    /* Estilo para el texto UVG */
    .logo-uvg {
        position: absolute;
        top: 20px;
        left: 20px;
        font-family: 'Arial', sans-serif;
        font-size: 36px;
        font-weight: bold;
        color: #004d26;
    }
</style>
</head>
<body>
<div class="logo-uvg">UVG</div>

<div class="container">
<h1>Información Almacenada</h1>

<div class="info-section">
<h2>{{ data['Universidad']['Nombre'] }}</h2>
<p><strong>Ubicación:</strong> {{ data['Universidad']['Ubicación'] }}</p>
<p><strong>Fundación:</strong> {{ data['Universidad']['Fundación'] }}</p>
<p><strong>Lema:</strong> {{ data['Universidad']['Lema'] }}</p>
<p><strong>Sitio Web:</strong> <a href="{{ data['Universidad']['Sitio_Web'] }}" target="_blank">{{ data['Universidad']['Sitio_Web'] }}</a></p>
</div>

<div class="info-section">
<h2>Facultades</h2>
{% for facultad in data['Universidad']['Facultades'] %}
<h3>{{ facultad['Nombre'] }}</h3>
{% for depto in facultad['Departamentos'] %}
<h4>{{ depto['Nombre'] }}</h4>
{% for programa in depto['Programas'] %}
<h5>{{ programa['Grado'] }}</h5>
<p>{{ programa['Descripción'] }}</p>

<h6>Definición:</h6>
{% if programa['Definición'] %}

```

```

<ul>
  {% for definicion in programa['Definición'] %}
  <li>{{ definicion }}</li>
  {% endfor %}
</ul>
{% else %}
<p>No hay información disponible.</p>
{% endif %}

<h6>Áreas de Investigación:</h6>
{% if programa['Áreas_de_Investigación'] %}
<ul>
  {% for area in programa['Áreas_de_Investigación'] %}
  <li>{{ area }}</li>
  {% endfor %}
</ul>
{% else %}
<p>No hay información disponible.</p>
{% endif %}

<h6>Áreas de Excelencia:</h6>
{% if programa['Áreas_de_Excelencia'] %}
<ul>
  {% for area in programa['Áreas_de_Excelencia'] %}
  <li>{{ area }}</li>
  {% endfor %}
</ul>
{% else %}
<p>No hay información disponible.</p>
{% endif %}

<h6>Competencias:</h6>
{% if programa['Competencias'] %}
<ul>
  {% for competencia in programa['Competencias'] %}
  <li>{{ competencia }}</li>
  {% endfor %}
</ul>
{% else %}
<p>No hay información disponible.</p>
{% endif %}

<h6>Áreas de Acción:</h6>
{% if programa['Áreas_de_Acción'] %}
<ul>
  {% for accion in programa['Áreas_de_Acción'] %}
  <li>{{ accion }}</li>
  {% endfor %}
</ul>
{% else %}
<p>No hay información disponible.</p>
{% endif %}

<h6>Desarrollo de Potencial:</h6>
{% if programa['Desarrollo_de_Potencial'] %}
<ul>
  {% for potencial in programa['Desarrollo_de_Potencial'] %}
  <li><strong>{{ potencial['Area'] }}:</strong> {{ potencial['
    Descripcion'] }}</li>

```

```

{% endfor %}
</ul>
{% else %}
<p>No hay información disponible.</p>
{% endif %}

<h6>Perfil del Egresado:</h6>
{% if programa['Perfil_del_Egresado'] %}
<p>{{ programa['Perfil_del_Egresado'] }}</p>
{% else %}
<p>No hay información disponible.</p>
{% endif %}

<h6>Líneas de Investigación y Proyectos:</h6>
{% if programa['Lineas_de_Investigacion_y_Proyectos'] %}
<ul>
{% for proyecto in programa['Lineas_de_Investigacion_y_Proyectos']
  %}
<li>{{ proyecto }}</li>
{% endfor %}
</ul>
{% else %}
<p>No hay información disponible.</p>
{% endif %}

<h6>Plan de Estudios:</h6>
{% if programa['Plan_de_Estudios'] %}
{% for ano, cursos in programa['Plan_de_Estudios'].items() %}
<h7>{{ ano }}</h7>
<ul>
{% for curso in cursos %}
<li>{{ curso }}</li>
{% endfor %}
</ul>
{% endfor %}
{% else %}
<p>No hay información disponible.</p>
{% endif %}
{% endfor %}
{% endfor %}
</div>

{% if 'Informacion Adicional' in data %}
<div class="info-section">
<h2>Información Adicional</h2>
{% for info in data['Informacion Adicional'] %}
<p>{{ info }}</p>
{% endfor %}
</div>
{% endif %}

<div class="buttons">
<a href="/descargar_informacion" style="text-decoration: none;">
<button>Descargar Información</button>
</a>
<a href="/" style="text-decoration: none;">
<button class="back-button">Volver al Menú</button>
</a>

```

```
</div>
</div>
</body>
</html>
```

Resultado práctico

En la Figura 37 se muestra la información inicial del documento, tal como aparece en la pestaña al momento de ingresar.

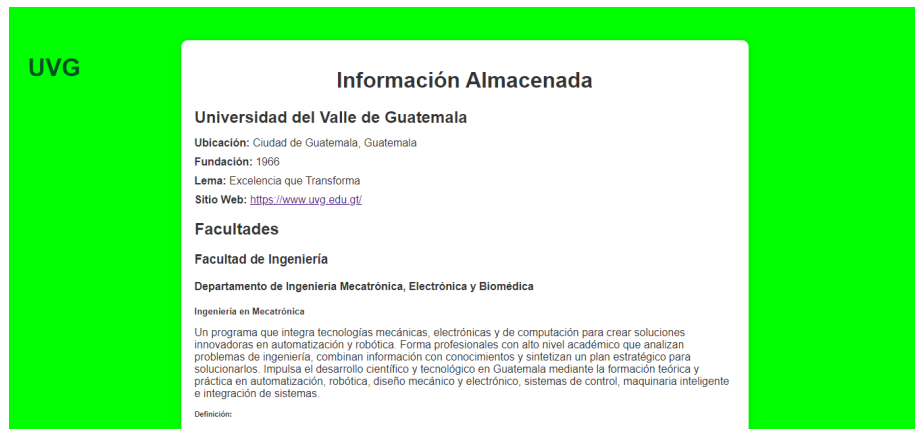


Figura 37: Pestaña información almacenada

En la Figura 38 se muestra la parte inferior de la pestaña, después de haber desplazado toda la información. Se puede observar que la última sección se titula "Información Adicional", donde se encuentra la información ingresada manualmente en la sección "Educa a la IA".

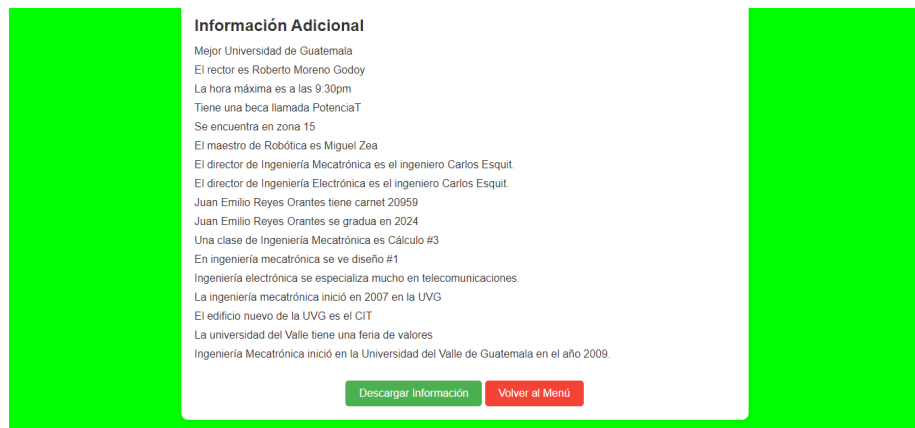


Figura 38: Pestaña información almacenada adicional

Se puede observar el botón *Descargar Información*, el cual, al ser seleccionado, descarga automáticamente el archivo .json que contiene toda la información visualizada. Tal como se observa en la Figura 39.

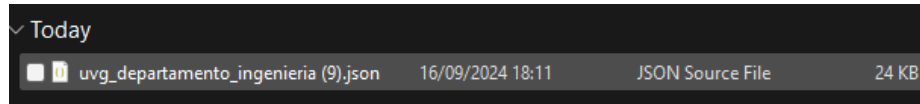


Figura 39: Archivo descargado

Como se muestra en la Figura 40, así es como se ve el archivo .json, que contiene toda la información utilizada para entrenar a la inteligencia artificial.

```
1 {
2   "Universidad": {
3     "Nombre": "Universidad del Valle de Guatemala",
4     "Ubicación": "Ciudad de Guatemala, Guatemala",
5     "Fundación": 1966,
6     "Lema": "Excelencia que Transforma",
7     "Sitio_Web": "https://www.uvg.edu.gt/",
8     "Facultades": [
9       {
10        "Nombre": "Facultad de Ingeniería",
11        "Departamentos": [
12          {
13            "Nombre": "Departamento de Ingeniería Mecatrónica, Electrónica y Biomédica",
14            "Programas": [
15              {
16                "Grado": "Ingeniería en Mecatrónica",
17                "Descripción": "Un programa que integra tecnologías mecánicas, electrónicas y de",
18                "Definición": [
19                  "Es un campo científico y tecnológico que combina diseño electrónico con dis",
20                ],
21                "Áreas de Investigación": [
22                  "Robótica y Automatización",
23                  "Sistemas Embebidos",
24                  "Instrumentación Biomédica",
25                  "Procesamiento de Señales",
26                  "Sistemas de Control"
27                ],
28                "Áreas de Excelencia": [
29                  "Automatización",
30                  "Diseño mecánico y electrónico",
31                  "Sistemas de control",
32                  "Maquinaria inteligente",
```

Figura 40: Información que contiene el archivo .json

En la Figura 41 de esta sección, se muestra cómo, al seleccionar el botón *Volver al menú*, la página regresa automáticamente al menú principal.

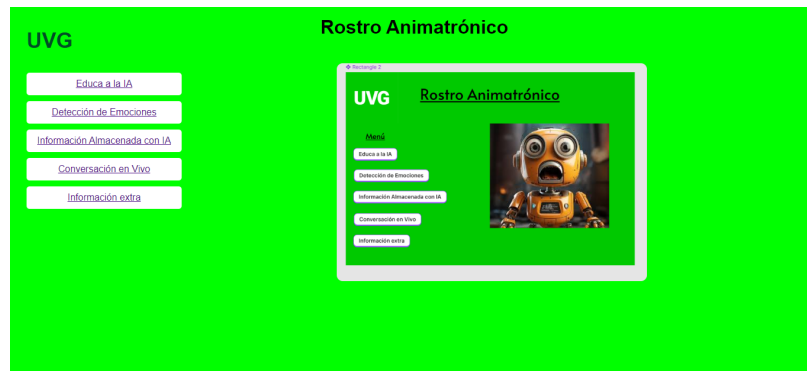


Figura 41: Página principal rostro animatrónico

9.2.4. Conversación en vivo

El propósito de esta parte del proyecto es integrar los programas de detección de emociones y la inteligencia artificial personalizada. De esta forma, la inteligencia artificial utiliza las emociones detectadas para proporcionar respuestas más personalizadas. Su objetivo es ofrecer respuestas alentadoras, buscando que la persona se sienta feliz, mientras se mantiene enfocada en brindar información sobre la Universidad del Valle de Guatemala.

En el archivo HTML, se configuran inicialmente los elementos estéticos, como los colores, la tipografía, el tamaño de letra, entre otros. Luego, se habilita el botón *Conectar con la Inteligencia Artificial*, que envía un comando a la extensión de ChatGPT para verificar la conexión, mostrando un aviso de conectado si recibe una respuesta.

La página web también solicita permisos para activar la cámara, que se enciende automáticamente al acceder a la pestaña "Conversación en Vivo". A continuación, se configura un interruptor para seleccionar si la conversación será hablada o escrita. Si se elige la opción de "Texto a voz", las respuestas se emitirán en voz; de lo contrario, solo aparecerán como texto.

Se configura una caja de texto donde se mostrarán todas las respuestas de la conversación, y otra caja de texto donde el usuario final podrá escribir. Además, se personaliza un botón de micrófono que permite al usuario hablar en lugar de escribir.

Luego, se definen las emociones que el sistema puede detectar y se asigna un emoji a cada una para hacer la experiencia más amigable. También se implementa programación defensiva en caso de que no se logre conectar con la inteligencia artificial.

Finalmente, se define que, al conectarse con la extensión de OpenAI, se enviará automáticamente el archivo .json que contiene toda la información necesaria para educar a la inteligencia artificial.

Código en HTML

El código del archivo en HTML es:

```
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Chat en Vivo con la IA</title>
<style>
body {
  background-color: #00FF00;
  font-family: Arial, sans-serif;
  display: flex;
  flex-direction: column;
  justify-content: center;
  align-items: center;
  height: 100vh;
  margin: 0;
  position: relative;
```

```

}
.chat-container {
  width: 600px;
  background-color: #ffffff;
  padding: 20px;
  border-radius: 10px;
  box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.1);
  display: flex;
  flex-direction: column;
  align-items: center;
}
.status-message {
  text-align: center;
  margin-bottom: 10px;
  font-size: 16px;
}
.emoji-display {
  text-align: center;
  font-size: 50px;
  margin-bottom: 20px;
}
.chat-box {
  flex-grow: 1;
  margin-bottom: 20px;
  overflow-y: auto;
  border: 1px solid #ddd;
  padding: 10px;
  border-radius: 5px;
  height: 200px;
  background-color: #f9f9f9;
  width: 100%;
}
.camera-container {
  margin-bottom: 20px;
  display: flex;
  justify-content: center;
  align-items: center;
  width: 100%;
}
video {
  border-radius: 10px;
  width: 150px;
  height: 100px;
  object-fit: cover;
}
.message {
  margin-bottom: 10px;
  padding: 5px 10px;
  border-radius: 5px;
  background-color: #e1f5fe;
  color: #333;
  font-size: 14px;
}
.user-message {
  background-color: #c8e6c9;
  text-align: right;
}
.bot-message {
  background-color: #e1f5fe;
}

```

```

}
.input-group {
    display: flex;
    align-items: center;
    width: 100%;
}
input[type="text"] {
    flex-grow: 1;
    padding: 10px;
    border: 1px solid #ccc;
    border-radius: 5px;
    font-size: 16px;
}
button {
    padding: 10px;
    margin-left: 10px;
    background-color: #4CAF50;
    color: white;
    border: none;
    border-radius: 5px;
    cursor: pointer;
    font-size: 16px;
}
button:hover {
    background-color: #45a049;
}
.connect-button {
    background-color: #f44336;
    color: white;
    margin-bottom: 10px;
}
.connect-button.connected {
    background-color: #4CAF50;
}
.mic-button {
    margin-left: 10px;
    background-color: #FF5722;
    border-radius: 50%;
    padding: 10px;
    display: flex;
    align-items: center;
    justify-content: center;
}
.mic-button:hover {
    background-color: #E64A19;
}
.file-input {
    display: none;
}
.file-label {
    margin-left: 10px;
    background-color: #FF5722;
    border-radius: 5px;
    padding: 10px;
    cursor: pointer;
    color: white;
}
.file-label:hover {
    background-color: #E64A19;
}

```

```

}
.back-button {
    margin-top: 10px;
    background-color: #f44336;
    border: none;
    color: white;
    padding: 10px 20px;
    text-align: center;
    text-decoration: none;
    display: inline-block;
    font-size: 16px;
    border-radius: 5px;
    cursor: pointer;
}
.back-button:hover {
    background-color: #d32f2f;
}
}
.switch-container {
    display: flex;
    align-items: center;
    justify-content: center;
    margin-bottom: 20px;
}
}
.switch-label {
    margin-right: 10px;
    font-size: 16px;
    font-weight: bold;
}
}
.switch {
    position: relative;
    display: inline-block;
    width: 60px;
    height: 34px;
}
}
.switch input {
    opacity: 0;
    width: 0;
    height: 0;
}
}
.slider {
    position: absolute;
    cursor: pointer;
    top: 0;
    left: 0;
    right: 0;
    bottom: 0;
    background-color: #ccc;
    transition: .4s;
    border-radius: 34px;
}
}
.slider:before {
    position: absolute;
    content: "";
    height: 26px;
    width: 26px;
    left: 4px;
    bottom: 4px;
    background-color: white;
    transition: .4s;
}

```

```

        border-radius: 50%;
    }
    input:checked + .slider {
        background-color: #4CAF50;
    }
    input:checked + .slider:before {
        transform: translateX(26px);
    }
    .logo-uvg {
        position: absolute;
        top: 20px;
        left: 20px;
        font-family: 'Arial', sans-serif;
        font-size: 36px;
        font-weight: bold;
        color: #004d26;
    }
</style>
</head>
<body>
<div class="logo-uvg">UVG</div>

<div class="chat-container">
<button id="connect-button" class="connect-button" onclick="checkConnection
    ()">Conectar con la IA</button>
<div class="status-message" id="status-message">Estado: Desconectado</div>
<div id="emoji-display" class="emoji-display">:</div>

<div class="camera-container">
<video id="video" autoplay muted></video>
</div>

<div class="switch-container">
<span class="switch-label">Texto a Voz:</span>
<label class="switch">
<input type="checkbox" id="text-to-speech-toggle">
<span class="slider"></span>
</label>
</div>

<div class="chat-box" id="chat-box">
<!-- Mensajes se mostrarán aquí -->
</div>
<div class="input-group">
<input type="text" id="user-input" placeholder="Escribe tu mensaje aquí..."
    disabled>
<button onclick="sendMessage()" disabled>Enviar</button>
<button class="mic-button" onclick="startRecognition()">Microfono</button>
<input type="file" id="file-input" class="file-input" onchange="
    handleFileUpload()">
<label for="file-input" class="file-label" clip-label>
</div>
<a href="/" class="back-button">Volver al Menú</a>
</div>

<script src="https://cdn.socket.io/4.0.0/socket.io.min.js"></script>
<script>
const socket = io();
let detectedEmotion = "neutral"; // Valor predeterminado

```

```

const statusMessage = document.getElementById('status-message');
const emojiDisplay = document.getElementById('emoji-display');
const userInput = document.getElementById('user-input');
const sendButton = document.querySelector('button[onclick="sendMessage()"]')
    ;
const connectButton = document.getElementById('connect-button');
const textToSpeechToggle = document.getElementById('text-to-speech-toggle');
const videoElement = document.getElementById('video');

const emojiMap = {
    "Felicidad": "😊",
    "Sorpresa": "😮",
    "Tristeza": "😞",
    "Miedo": "😱",
    "Enojo": "😡",
    "neutral": "😐"
};

function checkConnection() {
    fetch('/check_connection')
        .then(response => response.json())
        .then(data => {
            if (data.connected) {
                statusMessage.textContent = "Estado: Conectado";
                connectButton.classList.add('connected');
                connectButton.textContent = "Conectado";
                userInput.disabled = false;
                sendButton.disabled = false;
                uploadData(); // Subir los datos al conectarse con
                    éxito
            } else {
                statusMessage.textContent = "Estado: No se logró
                    conectar con la IA";
            }
        })
        .catch(error => {
            statusMessage.textContent = "Estado: Error al conectar";
            console.error('Error al verificar la conexión:', error);
        });
}

function uploadData() {
    fetch('/upload_data', {
        method: 'POST',
        headers: {
            'Content-Type': 'application/json'
        },
        body: JSON.stringify({ data: 'Información almacenada en la
            IA' })
    })
        .then(response => response.json())
        .then(data => {
            console.log('Datos subidos con éxito:', data);
        })
        .catch(error => {
            console.error('Error al subir los datos:', error);
        });
}

```

```

function sendMessage() {
  const message = userInput.value.trim();
  if (message !== "") {
    appendMessage("Tú", message, "user-message");
    socket.emit('message', { message, emotion: detectedEmotion
    });
    userInput.value = "";
  }
}

function appendMessage(sender, message, messageType) {
  const chatBox = document.getElementById('chat-box');
  const messageElement = document.createElement('div');
  messageElement.classList.add('message', messageType);
  messageElement.innerHTML = '<strong>${sender}</strong> ${message}';
  chatBox.appendChild(messageElement);
  chatBox.scrollTop = chatBox.scrollHeight;
}

socket.on('response', data => {
  appendMessage("IA", data.message, "bot-message");
  updateEmoji(data.emotion);
  if (textToSpeechToggle.checked) {
    speakMessage(data.message);
  }
});

function updateEmoji(emotion) {
  emojiDisplay.innerText = emojiMap[emotion] || emojiMap["neutral"];
}

function speakMessage(message) {
  const voices = window.speechSynthesis.getVoices();
  if (!voices.length) {
    window.speechSynthesis.onvoiceschanged = function() {
      speakMessage(message);
    };
    return;
  }

  const speech = new SpeechSynthesisUtterance(message);
  speech.lang = 'es-ES';
  speech.voice = voices.find(voice => voice.lang === 'es-ES') ||
  voices[0];
  window.speechSynthesis.speak(speech);
}

function startRecognition() {
  const recognition = new (window.SpeechRecognition || window.
  webkitSpeechRecognition)();
  recognition.lang = 'es-ES';

  recognition.onresult = function(event) {
    const transcript = event.results[0][0].transcript;
    userInput.value = transcript;
    sendMessage();
  };

  recognition.onerror = function(event) {

```

```

        console.error('Error en el reconocimiento de voz:', event.
            error);
    };

    recognition.start();
}

function handleFileUpload() {
    const fileInput = document.getElementById('file-input');
    const file = fileInput.files[0];
    if (file) {
        const reader = new FileReader();
        reader.onload = function(e) {
            appendMessage("Tú", 'Archivo subido: ${file.name}',
                "user-message");
        };
        reader.readAsDataURL(file);
    }
}

function simulateEmotionDetection() {
    const emotions = ["Felicidad", "Sorpresa", "Tristeza", "Miedo", "
        Enojo"];
    detectedEmotion = emotions[Math.floor(Math.random() * emotions.
        length)];
    updateEmoji(detectedEmotion);
}

function startCamera() {
    navigator.mediaDevices.getUserMedia({ video: true })
        .then(stream => {
            videoElement.srcObject = stream;
        })
        .catch(err => {
            console.error("Error al acceder a la cámara:", err);
        });
}

setInterval(simulateEmotionDetection, 5000);
startCamera();
</script>
</body>
</html>

```

Resultado práctico

En la Figura 42 se observa a primera vista la pestaña de Conversación en Vivo"

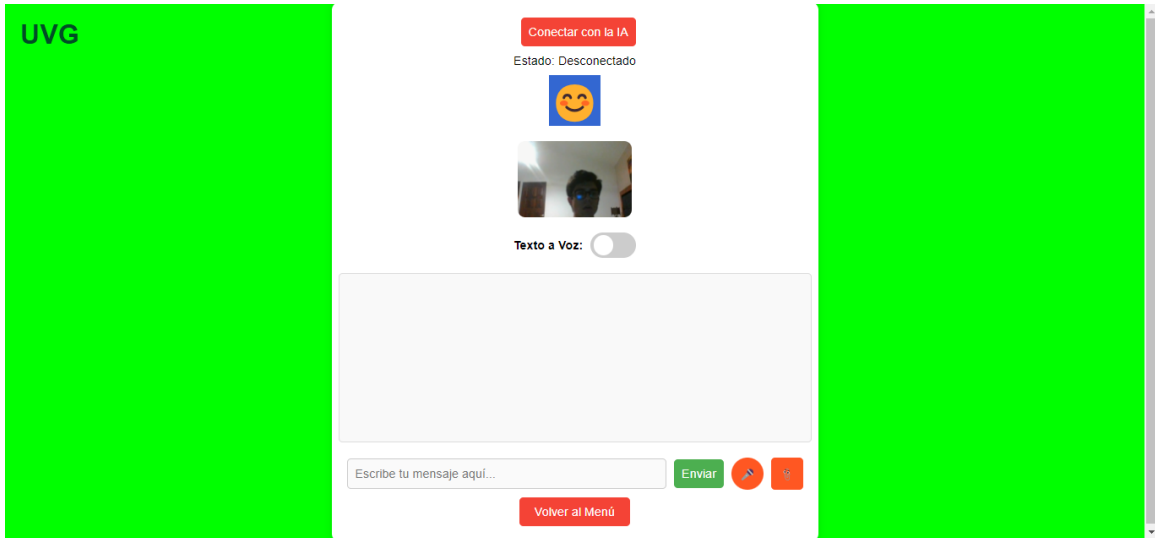


Figura 42: Primera vista de conversación en vivo

En la Figura 43 se observa que ya se conecto con la extensión de ChatGPT.

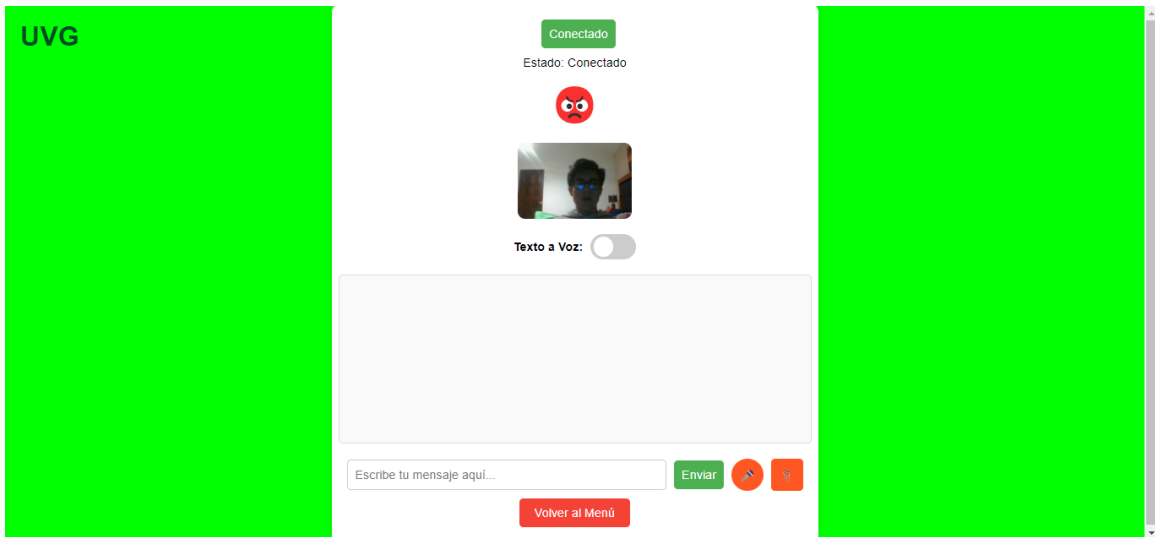


Figura 43: Pestaña conversación en vivo, conectado a extensión

En la Figura 44 se observa como el usuario le hizo una consulta, y el programa le respondió tomando en cuenta la emoción del usuario final y la información de la Universidad del Valle de Guatemala.

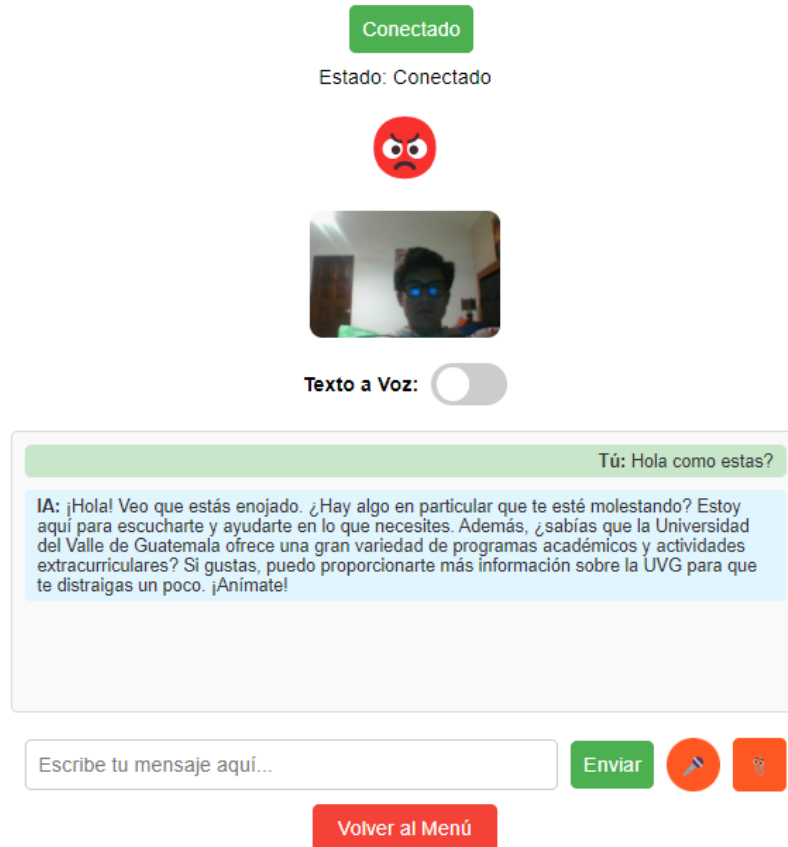


Figura 44: Interacción entre detección de emoción y el texto ingresado

En la Figura 45 se observa que esta activada la opción de "Texto a Voz", y la respuesta fue dada de forma hablada.

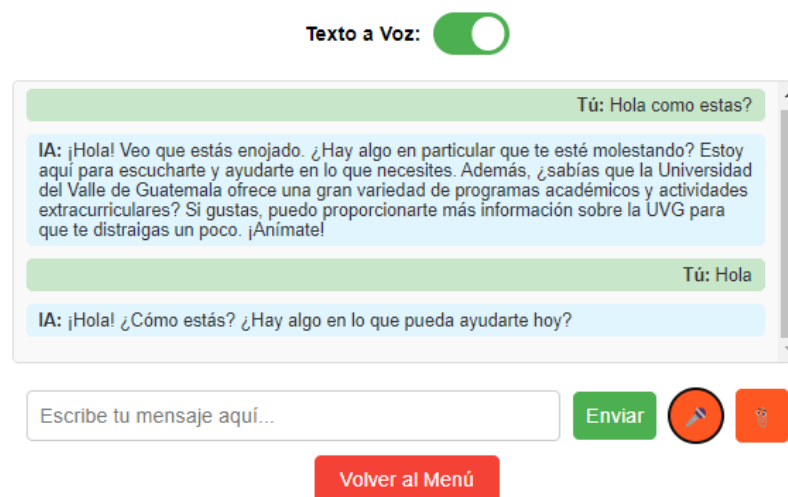


Figura 45: Opción de texto a voz activada

En la Figura 46, se muestra que es posible subir documentos al seleccionar el ícono de "Clip". Al hacer esto, se abrirá la ventana de exploración de documentos del dispositivo.

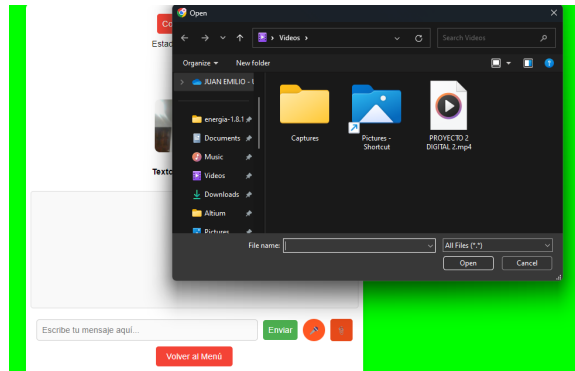


Figura 46: Selección de archivos en la computadora

En la Figura 47 se observa como un archivo ya fue seleccionado y este se envió por medio del chat.

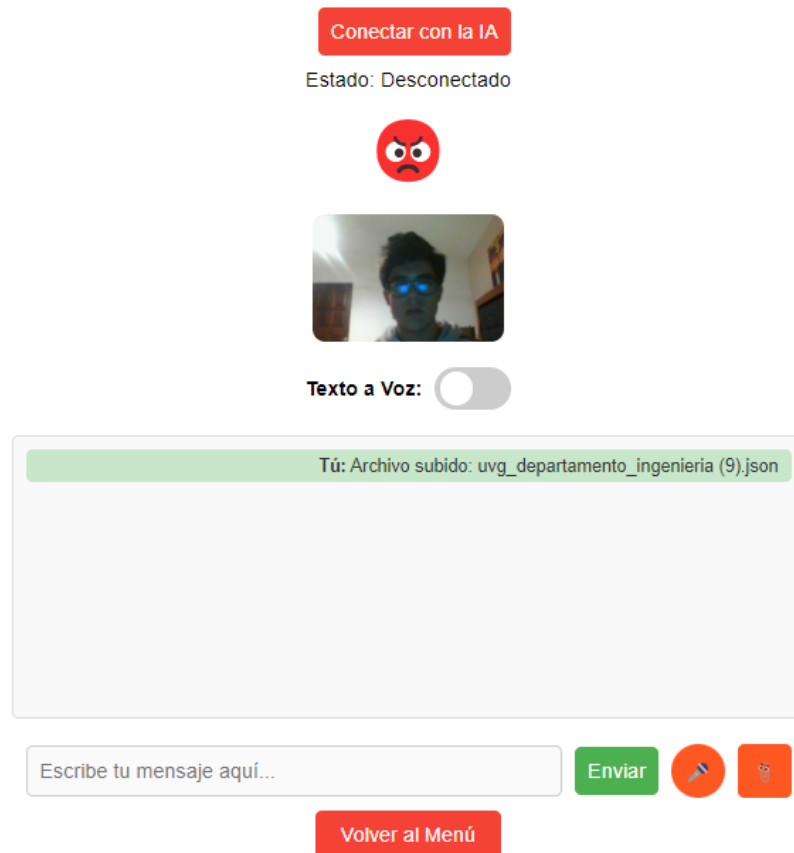


Figura 47: Archivo seleccionado y agregado en el programa

Por último de esta sección, si el botón de *Volver a Menú* es seleccionado, la página

regresa a la página de inicio, la cual es donde se encuentran el menú.

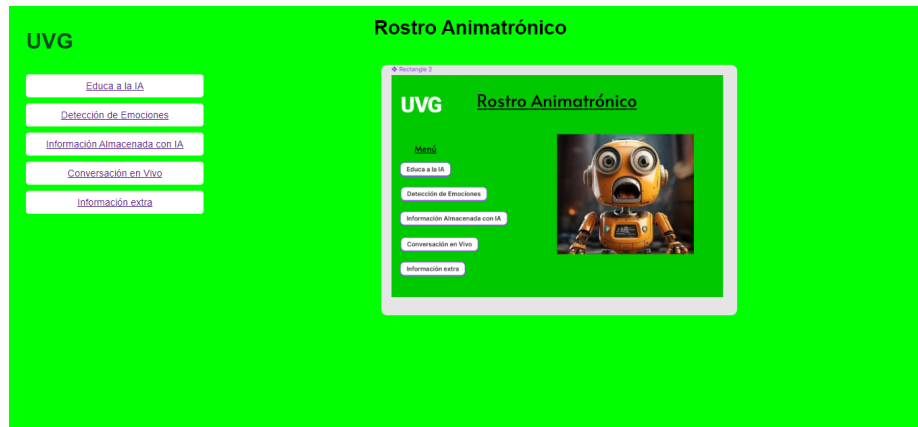


Figura 48: Pestaña principal

9.2.5. Información extra

El propósito de esta pestaña es redirigir al usuario hacia la página oficial de la Universidad del Valle de Guatemala, permitiendo a los usuarios finales obtener más información sobre la institución directamente desde su sitio web oficial.

En este caso, no se utilizó un archivo HTML, ya que la redirección se realiza directamente desde el archivo principal, `.App.py`.

Código en Python

El código del archivo en Python es:

```
@app.route('/informacion_extra')
def informacion_extra():
    return redirect("https://www.uvg.edu.gt")
```

Resultado práctico

En la Figura 49 se observa lo que aparece cuando el botón que dice *Información Extra* es seleccionado.

The image shows the homepage of the Universidad del Valle de Guatemala (UVG) website, specifically for the 2024 race registration. The header includes the UVG logo and navigation links for ESTUDIANTES, COLABORADORES, EGRESADOS, DONANTES, and PADRES. Below the header is a green navigation bar with links for NOSOTROS, ADMISIONES, ACADÉMICO, VIDA ESTUDIANTIL, IMPACTO, INVESTIGACIÓN, and SERVICIOS. The main banner features the text "CARRERA UVG 2024" and "Transformando sueños en oportunidades Circuito La Reforma". It also displays the registration date as "Domingo 22 de septiembre" at "6:00 a.m." with a registration fee of "Q150". A prominent "INSCRÍBETE" button is present. A circular badge indicates "10 HORAS ADICIONALES" for those who register. Below the banner, there is a section for "PRUEBA DE ADMISIÓN PRESENCIAL" with a photo of a student in a lab. To the right, it states "Fecha límite de inscripción: SÁBADO, 28 DE SEPTIEMBRE".

UVG UNIVERSIDAD DEL VALLE DE GUATEMALA

ESTUDIANTES COLABORADORES EGRESADOS DONANTES PADRES

ACTUALIDAD UVG EDUCACION TRABAJA EN UVG OPTIMIZADO POR Google

NOSOTROS ADMISIONES ACADÉMICO VIDA ESTUDIANTIL IMPACTO INVESTIGACIÓN SERVICIOS

CARRERA UVG 2024

Transformando sueños en oportunidades Circuito La Reforma

Domingo 22 de septiembre 6:00 a.m. Inscripción: Q150

INSCRÍBETE

10 HORAS ADICIONALES

5 HORAS ADICIONALES por cada persona que inscribas *No aplica para horas de extensión

PRUEBA DE ADMISIÓN PRESENCIAL

Fecha límite de inscripción: SÁBADO, 28 DE SEPTIEMBRE

Fecha de la prueba:

Figura 49: Página web UVG

Entrenamiento de la inteligencia artificial

Para el entramiento de la inteligencia artificial se creó un archivo tipo .json, el cual contiene información general de la Universidad del Valle de Guatemala y detallada del departamento de ingeniería mecatrónica, electrónica y biomédica.

El primer paso fue crear un archivo .text, y almacenar toda la información adentro de el.

10.1. Creación de archivo .json con información deseada

El proceso de trabajo consistió en crear un archivo .json en la aplicación Visual Studio Code, donde se empezó a almacenar toda la información. Como la información original estaba en un archivo .txt, fue necesario cambiar la sintaxis para que fuera compatible con el formato .json.

10.1.1. Información general de la Universidad del Valle

La información general de la Universidad del Valle de Guatemala fue la siguiente:

```

"Universidad": {
  "Nombre": "Universidad del Valle de Guatemala",
  "Ubicación": "Ciudad de Guatemala, Guatemala",
  "Fundación": 1966,
  "Lema": "Excelencia que Transforma",
  "Sitio_Web": "https://www.uvg.edu.gt/",
}

```

Figura 50: Información general de la universidad

Información extra la cual ha sido directamente desde el página web, es la siguiente:

```

"Información Adicional": [
  "Mejor Universidad de Guatemala",
  "El rector es Roberto Moreno Godoy",
  "La hora máxima es a las 9:30pm",
  "Tiene una beca llamada PotenciaT",
  "Se encuentra en zona 15",
  "El maestro de Robótica es Miguel Zea",
  "El director de Ingeniería Mecatrónica es el ingeniero Carlos Esquit.",
  "El director de Ingeniería Electrónica es el ingeniero Carlos Esquit.",
  "Juan Emilio Reyes Orantes tiene carnet 20959",
  "Juan Emilio Reyes Orantes se gradua en 2024",
  "Una clase de Ingeniería Mecatrónica es Cálculo #3",
  "En ingeniería mecatrónica se ve diseño #1",
  "Ingeniería electrónica se especializa mucho en telecomunicaciones.",
  "La ingeniería mecatrónica inició en 2007 en la UVG",
  "El edificio nuevo de la UVG es el CIT",
  "La universidad del Valle tiene una feria de valores",
  "Ingeniería Mecatrónica inició en la Universidad del Valle de Guatemala en el año 2009."
]

```

Figura 51: Información extraída de la página web

10.1.2. Información de Ingeniería Electrónica, Mecatrónica y Biomédica

La información esta dividida según la carrera específica, incluyendo únicamente estas tres siguientes carreras:

10.1.3. Información detallada de Ingeniería Mecatrónica

La información detallada de de Ingeniería Mecatrónica es la siguiente:

```

{
  "Grado": "Ingeniería en Mecatrónica",
  "Descripción": "Un programa que integra tecnologías mecánicas, electrónicas y de computación para crear soluciones innovadoras en automatización y robótica. Forma profesionales con alto nivel académico que analizan problemas de ingeniería, combinan información con conocimientos y sintetizan un plan estratégico para solucionarlos. Impulsa el desarrollo científico y tecnológico en Guatemala mediante la formación teórica y práctica en automatización, robótica, diseño mecánico y electrónico."
}

```

```

sistemas de control , maquinaria inteligente ue integración de
sistemas. ",
"Definición": [
"Es un campo científico y tecnológico que combina diseño electrónico
con diseño mecánico para crear integraciones de los denominados
sistemas Meca-Trónicos. Tiene mucha relación con programación,
principalmente a nivel de máquinas y circuitos, pues muchos de
los sistemas mecatrónicos requieren inteligencia que les permita
ser autónomos o realizar alguna acción específica como
respuesta a su entorno. El programa tiene fuerte orientación a
robótica y automatización. La mecatrónica cambia el mundo día a
día a través de la creación de tecnologías que todos los seres
humanos utilizamos."
],
"Áreas de Investigación": [
"Robótica y Automatización",
"Sistemas Embebidos",
"Instrumentación Biomédica",
"Procesamiento de Señales",
"Sistemas de Control"
],
"Áreas de Excelencia": [
"Automatización",
"Diseño mecánico y electrónico",
"Sistemas de control",
"Maquinaria inteligente",
"Integración de sistemas"
],
"Competencias": [
"Pensamiento crítico y analítico",
"Comunicación efectiva",
"Trabajo en equipo",
"Resolución efectiva de problemas",
"Uso adecuado de la tecnología",
"Ejercicio de liderazgo",
"Actuación ética",
"Investigación en diferentes áreas del conocimiento",
"Aprendizaje autónomo",
"Actitud emprendedora e innovadora"
],
"Áreas de Acción": [
"Investigación, planteamiento, planeación, desarrollo, simulación y
manufactura de sistemas para optimizar procesos.",
"Servicio público, instituciones de investigación y docencia,
industria en general, o como profesional independiente.",
"Diseño de componentes mecánicos y electrónicos de integración de
ambos.",
"Sistemas de control.",
"Maquinaria inteligente.",
"Mantenimiento Industrial."
],
"Desarrollo de Potencial": [
"Instrumentación y automatización industrial: Rama de la ingeniería
civil que se ocupa del diseño y cálculo de la parte estructural
en elementos y sistemas estructurales tales como edificios,
puentes, muros, presas, túneles y otras obras civiles.",
"Diseño Electrónico: Diseñar y optimizar sistemas automatizados de
producción industrial con base a sensores y actuadores electró
nicos. Esto implica programación de equipos y dispositivos de

```

```

control como computadores y PLCs, así como parametrización y configuración de electrónica de campo.",
"Diseño mecánico: Análisis y diseño de piezas mecánicas y mecanismos para sistemas robóticos. Aquí se analiza el comportamiento estático y dinámico de las piezas que conforman diseños mecánicos, los cuales combinados con diseño electrónico, producen sistemas mecatrónicos vanguardistas."
],
"Perfil_del_Egresado": "El ingeniero en Mecatrónica es un profesional que diseña e implementa sistemas mecánicos y electrónicos (mecatrónicos) útiles en cualquier industria. Analiza, diseña y optimiza sistemas mecatrónicos, con competencias que permiten desempeñarse en diferentes áreas de la industria e investigación, tanto a nivel nacional como internacional.",
"Plan_de_Estudios": {
  "Año_1": [
    "Requisito Inglés",
    "Algoritmos y Programación Básica",
    "Pensamiento Cuantitativo",
    "Comunicación Efectiva",
    "Introducción a la Ingeniería Electrónica y Mecatrónica",
    "Coaching para la Excelencia",
    "Química General",
    "Ciudadanía Global",
    "Dibujo Mecánico",
    "Cálculo 1",
    "Física 1",
    "Álgebra Lineal 1",
    "Guatemala en el Contexto Mundial",
    "Retos Ambientales y Sostenibilidad"
  ],
  "Año_2": [
    "Estadística 1",
    "Cálculo 2",
    "Física 2",
    "Circuitos Eléctricos 1",
    "Máquinas Herramienta",
    "Ciencia de los Materiales",
    "Cálculo 3",
    "Ecuaciones Diferenciales 1",
    "Circuitos Eléctricos 2",
    "Ecología para Todos",
    "Electrónica Digital 1",
    "Mecánica 1: Estática"
  ],
  "Año_3": [
    "Investigación y Pensamiento Científico",
    "Curso Selectivo (Formación General)",
    "Programación de Microcontroladores",
    "Mecánica 2: Dinámica",
    "Resistencia de Materiales 1",
    "Teoría Electromagnética 1",
    "Curso Selectivo",
    "Electrónica Analógica 1",
    "Simulación de Circuitos y Fabricación de PCB",
    "Ecuaciones Diferenciales 2",
    "Emprendimiento e Innovación",
    "Resistencia de Materiales 2"
  ],
}

```

```

"Año_4": [
  "Instrumentación_y_Automatización_Industrial_1",
  "Procesamiento_de_Señales",
  "Diseño_de_Ingeniería_Mecánica_1",
  "Instalaciones_Eléctricas",
  "Electrónica_Digital_2",
  "Termodinámica_y_Mecánica_de_Fluidos",
  "Sistemas_de_Control_1",
  "Instrumentación_y_Automatización_Industrial_2",
  "Diseño_de_Ingeniería_Mecánica_2",
  "Mecanismos",
  "Temas_Especiales_de_Ingeniería_Electrónica_y_Mecatrónica_1"
],
"Métodos_Numéricos_1"
],
"Año_5": [
  "Introducción_a_Máquinas_Elécricas",
  "Sistemas_Hidráulicos_y_Neumáticos",
  "Economía_de_Empresas",
  "Sistemas_de_Control_2",
  "Robótica_1",
  "Práctica_profesional",
  "Diseño_e_Innovación_en_Ingeniería_1",
  "Robótica_2",
  "Ingeniería_Económica",
  "Introducción_a_Diseño_y_Manufactura_CAD_CAM",
  "Gestión_de_Proyectos",
  "Diseño_e_Innovación_en_Ingeniería_2",
  "Trabajo_de_graduación"
]
}

```

10.1.4. Información detallada de Ingeniería Electrónica

La información detallada de de Ingeniería Electrónica es la siguiente:

```

"Grado": "Ingeniería_en_Electrónica",
"Descripcion": "La_Ingeniería_Electrónica_es_el_campo_del_conocimiento_que_diseña_la_esencia_de_todas_las_tecnologías_que_hacen_funcionar_al_mundo_en_la_actualidad. Es_el_campo_que_ha_provocado_la_tercera_revolución_industrial_(la_actual). Incorpora_los_fundamentos_del_diseño_electrónico_analógico_y_digital_con_diversas_tecnologías_vanguardistas, análisis_y_diseño_de_sistemas_de_telecomunicaciones_cableadas_e_inalámbricas, y_diseño_de_sistemas_nanoeléctricos.",
"Campo_de_Servicio": "La_ingeniería_electrónica_es_en_realidad_un_campo_de_absoluto_servicio_a_las_demás_disciplinas, pues_representa_al_productor_de_tecnologías_que_crean_herramientas_para_la_operación, investigación_y_desarrollo_en_todas_las_áreas_del_conocimiento, mejorando_la_calidad_de_vida_mediante_sistemas_de_telecomunicaciones, equipos_biomédicos_y_equipos_electrónicos_de_consumo.",
"Enfoque_del_Programa": "El_programa_de_Ingeniería_Electrónica_está_diseñado_con_un_enfoque_intenso_tanto_en_teoría_como_en_práctica, razón_por_la_cual_consiste_en_diversas_áreas_curriculares_y_ejes_transversales_desarrollados_a_través_de_iniciativas_acadé

```

```

micas que incluyen laboratorios, trabajo en equipo y metodología
orientada a proyectos y realización de diseños de ingeniería
funcionales.",
"Desarrollo_de_Potencial": [
{
  "Area": "Redes y Telecomunicaciones",
  "Descripcion": "Infraestructura y tecnología que soporta a
  todos los sistemas de comunicaciones digitales en el
  mundo actual, abarcando desde la temática de redes de
  datos en su nivel fundamental hasta llegar a las
  tecnologías y plataformas de telecomunicaciones, en sus
  diversas generaciones, utilizadas en todo el mundo."
},
{
  "Area": "Diseño Electrónico",
  "Descripcion": "Desarrollo de hardware, firmware y software
  con el fin de proveer soluciones a una amplia variedad
  de problemáticas. Abarca desde el conocimiento de
  componentes electrónicos y sus diversas interconexiones
  hasta la programación del conjunto de instrucciones que
  regirán el comportamiento de un sistema electrónico
  completo, fabricación de tarjetas electrónicas y el uso
  de tecnología nanoelectrónica."
},
{
  "Area": "Instrumentación y Automatización Industrial",
  "Descripcion": "Se refiere a todo el equipo electrónico dise-
  ñado e implementado para el control automático de
  procesos industriales de gran escala. Incluye mediciones
  de variables físicas para digitalizarlas e
  interconectarlas con controladores, optimizando el
  desempeño y la estabilidad del proceso."
}
],
"Lineas_de_Investigacion_y_Proyectos": [
"Diseño del primer nanochip en la historia de Guatemala (de hecho el
primero para casi toda Latinoamérica).",
"Diseño de neuroestimulador para pacientes con epilepsia (en
conjunto con el Centro de Epilepsia y Neurocirugía Funcional -
HUMANA).",
"Modelaje matemático para aplicaciones de emulación y procesamiento
de señales de audio."
],
"Perfil_del_Egresado": [
"Formar profesionales con alto nivel académico que analicen un
problema de ingeniería, combinen la información del problema con
sus conocimientos, y aplicando sus competencias, sintetizan un
plan estratégico que le dé solución.",
"Formar profesionales con sólida base que les permita fácil
integración a universidades extranjeras para estudios de
posgrado en las diversas áreas de la ingeniería.",
"Impulsar el desarrollo científico y tecnológico en Guatemala a travé-
s de la formación teórica y práctica de profesionales en este
campo pilar de la era digital.",
"Formar ingenieros competentes para administrar eficientemente los
recursos electrónicos de Guatemala y aportar al desarrollo de
nuevas tecnologías.",
"Brindar al profesional egresado los fundamentos intelectuales y
morales que le permitan ejercer liderazgo técnico y humano en la

```

```

    sociedad_guatemalteca_y_extranjera."
  ],
  "Plan_de_Estudios": {
    "Año_1": [
      "Requisito_Inglés",
      "Algoritmos_y_Programación_Básica",
      "Pensamiento_Cuantitativo",
      "Comunicación_Efectiva",
      "Coaching_para_la_Excelencia",
      "Química_General",
      "Introducción_a_la_Ingeniería_Electrónica_y_Mecatrónica",
      "Ciudadanía_Global",
      "Cálculo_1",
      "Física_1",
      "Dibujo_Mecánico",
      "Álgebra_Lineal_1",
      "Ciencia_de_Materiales_Electrónicos_1",
      "Guatemala_en_el_Contexto_Mundial"
    ],
    "Año_2": [
      "Cálculo_2",
      "Investigación_y_Pensamiento_Científico",
      "Estadística_2",
      "Física_2",
      "Circuitos_Eléctricos_1",
      "Retos_Ambientales_y_Sostenibilidad",
      "Circuitos_Eléctricos_2",
      "Ecuaciones_Diferenciales_1",
      "Cálculo_3",
      "Electrónica_Digital_1",
      "Ecología_para_Todos",
      "Mecánica: Estática_y_Dinámica"
    ],
    "Año_3": [
      "Emprendimiento_e_Innovación",
      "Curso_Selectivo_(Formación_General)",
      "Electrónica_Analógica_1",
      "Simulación_de_Circuitos_y_Fabricación_de_PCB",
      "Ecuaciones_Diferenciales_2",
      "Programación_de_Microcontroladores",
      "Teoría_Electromagnética_1",
      "Electrónica_Analógica_2",
      "Electrónica_Digital_2",
      "Métodos_Numéricos_1",
      "Teoría_Electromagnética_2"
    ],
    "Año_4": [
      "Economía_de_Empresas",
      "Instrumentación_y_Automatización_Industrial_1",
      "Procesamiento_de_Señales",
      "Electrónica_Digital_3",
      "Interconexión_de_Dispositivos_para_Red",
      "Temas_Especiales_en_Ingeniería_Electrónica_y_Mecatrónica_1",
      "Instalaciones_Eléctricas",
      "Arquitectura_de_Computadores",
      "Sistemas_de_Telecomunicaciones_1",
      "Sistemas_de_Control_1",
      "Instrumentación_y_Automatización_Industrial_2"
    ]
  }
}

```

```

"Nanoelectrónica_1"
],
"Año_5": [
"Redes de Telefonía Inalámbrica",
"Sistemas de Telecomunicaciones_2",
"Sistemas de Control_2",
"Temas Especiales de Ingeniería Electrónica y Mecatrónica_2"
,
"Nanoelectrónica_2",
"Práctica Profesional",
"Diseño de Innovación en Ingeniería_1",
"Sistemas de Telecomunicaciones_3",
"Ingeniería Económica",
"Gestión de Proyectos",
"Introducción a Máquinas Eléctricas",
"Diseño de Innovación en Ingeniería_2",
"Trabajo de graduación"
]

```

10.1.5. Información detallada de Ingeniería Biomédica

La información detallada de de Ingeniería Biomédica es la siguiente:

```

"Grado": "Ingeniería Biomédica",
"Descripción": "Un programa que aplica principios de ingeniería al campo médico, enfocándose en el desarrollo de tecnologías para el cuidado de la salud y dispositivos médicos.",
"Definición": [
"La Ingeniería Biomédica forma profesionales con un alto nivel académico, teórico y práctico, capaces de analizar problemas de ingeniería, combinar información del problema con sus conocimientos y, a través de sus competencias, proporcionar soluciones innovadoras y efectivas."
],
"Áreas de Investigación": [
"Fisiología: Estudio de los diversos mecanismos fisiológicos y anatómicos desde una perspectiva cuantitativa, lo que permite comprenderlos y modelarlos matemáticamente. Esto facilita el diseño de dispositivos electrónicos o robóticos para capturar información o interactuar con dichos mecanismos, comprendiendo su comportamiento e interacción, así como las características de las señales que pueden capturar.",
"Robótica Médica: Diseño de robots con aplicaciones médicas asistenciales, desarrollando herramientas robotizadas basadas en las necesidades y requerimientos de diagnóstico, tratamiento, cirugía o rehabilitación. Conocer los mecanismos fisiológicos es crucial para crear diseños robóticos efectivos y adecuados.",
"Instrumentación y Dispositivos Médicos: Diseño y fabricación de instrumentos y dispositivos electrónicos para capturar información fisiológica, ya sea para procesamiento o análisis posterior o para estimular los mecanismos fisiológicos mediante señales de corriente, voltaje, frecuencia, campos eléctricos o magnéticos, entre otros, logrando influir en los procesos médicos según sea necesario."
],
"Áreas de Excelencia": [
"Ingeniería de Rehabilitación: Desarrollo de tecnologías para mejorar la calidad de vida de personas con discapacidades, incluyendo prótesis, ortesis y dispositivos de asistencia.",

```

```

"Imágenes Médicas: Innovación en técnicas de captura y procesamiento de imágenes para diagnóstico y tratamiento médico.",
"Biomecánica: Aplicación de principios de mecánica a los sistemas biológicos para estudiar la función y el movimiento del cuerpo humano."
],
"Competencias": [
"Análisis crítico y resolución de problemas complejos.",
"Diseño y desarrollo de dispositivos biomédicos.",
"Integración de tecnologías emergentes en aplicaciones médicas.",
"Comunicación efectiva y trabajo en equipos multidisciplinarios.",
"Conocimiento en normativas y regulación de dispositivos médicos."
],
"Áreas_de_Acción": [
"Centros de Investigación Biomédica.",
"Industria de Dispositivos Médicos.",
"Hospitales y Centros de Salud.",
"Consultoría en Ingeniería Biomédica.",
"Docencia y Capacitación."
],
"Desarrollo_de_Potencial": [
"Formación en innovación y emprendimiento tecnológico.",
"Participación en proyectos interdisciplinarios y colaboraciones internacionales.",
"Acceso a prácticas profesionales en hospitales y empresas de tecnología biomédica.",
"Capacitación en uso de herramientas y técnicas de última generación."
],
"Plan_de_Estudios": {
"Año_1": [
"Requisito Inglés",
"Algoritmos y Programación Básica",
"Pensamiento Cuantitativo",
"Ciencias de la Vida",
"Coaching para la Excelencia",
"Química 1",
"Introducción a la Ingeniería Biomédica",
"Ciudadanía Global",
"Comunicación Efectiva",
"Dibujo Mecánico",
"Cálculo 1",
"Física 1",
"Química 2",
"Estadística 1"
],
"Año_2": [
"Cálculo 2",
"Álgebra Lineal 1",
"Biomecánica",
"Física 2",
"Investigación y Pensamiento Científico",
"Química Orgánica 1",
"Guatemala en el Contexto Mundial",
"Circuitos Eléctricos 1",
"Ecuaciones Diferenciales 1",
"Cálculo 3",
"Retos Ambientales y Sostenibilidad",
"Biomateriales"
],
"Año_3": [

```

```

"Curso_Selectivo_(Formación_General)",
"Circuitos_Elctricos_2",
"Electrnica_Digital_1",
"Fisiología_Cuantitativa_1",
"Mecánica_Estática_y_Dinámica",
"Diseño_Mecánico_y_Mecanismos",
"Electrnica_Analógica_1_(Biomédica)",
"Emprendimiento_e_Innovación",
"Fenómenos_de_Transporte_Biológico_e_Ingeniería_de_Tejidos",
"Fisiología_Cuantitativa_2",
"Bioética,_Legislación_y_Normas_de_Calidad",
"Electrnica_Digital_2_(Biomédica)"
],
"Año_4": [
"Instrumentación_Biomédica",
"Análisis_de_Datos_Biomédicos",
"Curso_Selectivo",
"Electrnica_Analógica_2_(Biomédica)",
"Procesamiento_de_Señales_(Biomédica)",
"Simulación_de_Circuitos_y_Fabricación_de_PCB_(Biomédica)",
"Economía_de_Empresas",
"Regulación_y_Certificación_de_Dispositivos_Médicos",
"Introducción_a_Imágenes_Médicas",
"Curso_Selectivo",
"Gestión_de_Proyectos",
"Sistemas_de_Control_1_(Biomédica)"
],
"Año_5": [
"Tópicos_Avanzados_de_Ingeniería_Biomédica",
"Ingeniería_Económica",
"Práctica_Profesional_1",
"Sistemas_de_Control_2_(Biomédica)",
"Robótica_Médica_1",
"Diseño_e_Innovación_en_Ingeniería_1",
"Curso_Selectivo",
"Diseño_e_Innovación_en_Ingeniería_2",
"Práctica_Profesional_2",
"Trabajo_de_Graduación",
"Instalaciones_Elctricas_(Biomédica)",
"Robótica_Médica_2"
]
}

```

10.2. Entrenamiento de redes neuronales de ChatGPT

El proceso para entrenar las redes neuronales de la extensión de ChatGPT comienza cuando se establece la conexión con la inteligencia artificial. Para hacerlo, es necesario presionar el botón **Conectar con la IA**, ubicado en la pestaña **Conversación en Vivo**. Al presionar este botón, la información almacenada en el archivo llamado `üvg-departamento-ingenieria.json` se entrega automáticamente a la extensión de OpenAI, permitiendo que se eduque y base sus respuestas en la información proporcionada.

Cada vez que se presiona el botón **Conectar con la IA**, el archivo se sube nuevamente

para asegurar que la inteligencia artificial se mantenga actualizada con la información más reciente que el usuario haya ingresado a través de la pestaña "Educa a la IA".

Validación y pruebas de correcto funcionamiento del proyecto

A lo largo de los distintos capítulos, se ha demostrado el funcionamiento completo del proyecto, dividido en diferentes fases.

11.1. Conversación con inteligencia artificial, relacionada a la Universidad del Valle, tomando en cuenta la toma de emociones

Uno de los principales objetivos del proyecto es lograr una conversación fluida relacionada con la Universidad del Valle de Guatemala. A continuación, se presentarán varias figuras que demuestran cómo se mantiene esta conversación continua, tomando en cuenta las emociones del usuario final y utilizando la información con la que se ha entrenado el sistema. Estas interacciones incluyen preguntas en formato escrito y hablado, para mostrar el correcto y completo funcionamiento del sistema.

En la Figura 52, se muestra cómo el sistema, al detectar enojo en el usuario, generó una respuesta relacionada con la Universidad del Valle de Guatemala. Al mismo tiempo, intentó personalizar la respuesta ofreciendo ayuda para mejorar el estado de ánimo del usuario.

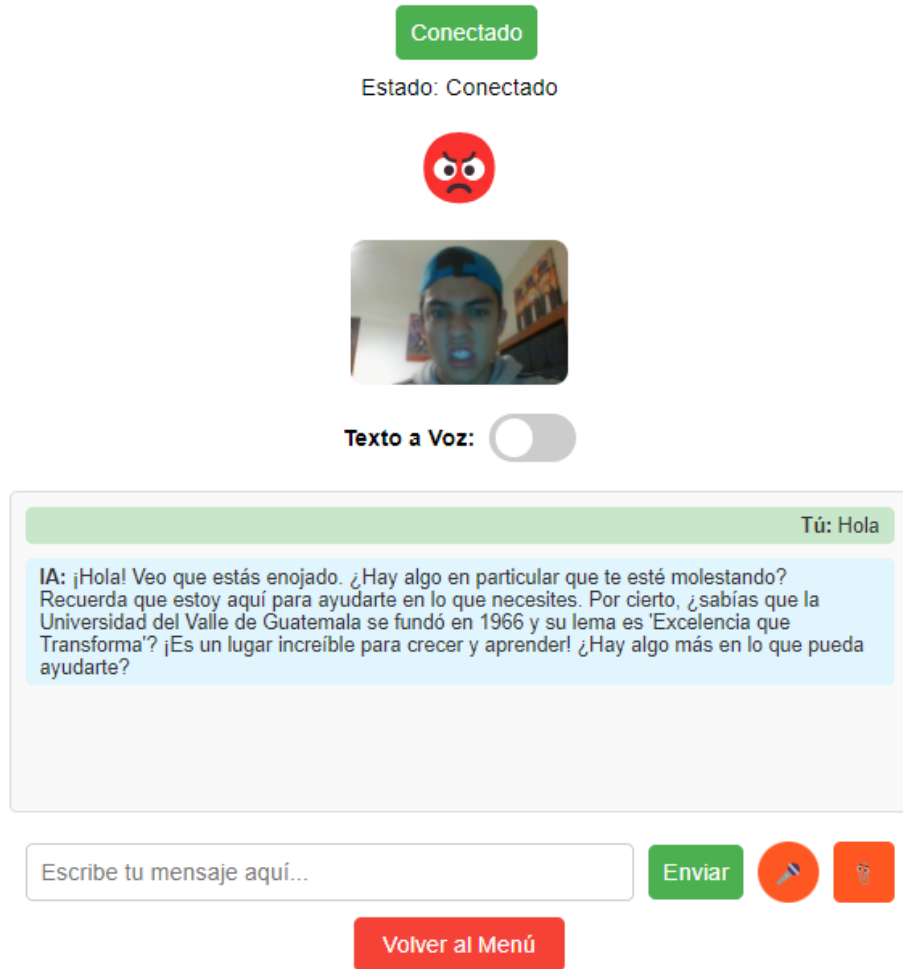


Figura 52: Conversación con inteligencia artificial y detección de emociones

En la Figura 53, se presenta una pregunta específica sobre las clases de la carrera de Ingeniería Mecatrónica, con el propósito de conocer los temas que se cubren en el quinto año de la carrera. La pregunta se realiza de forma hablada para observar cómo el sistema transcribe la consulta.

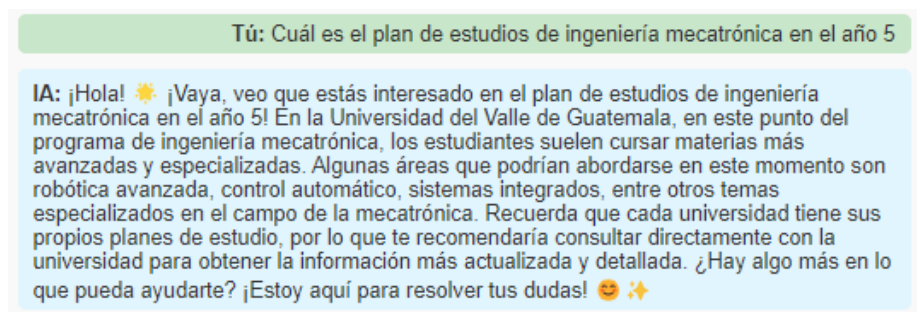


Figura 53: Entrada de voz, verificación de transcripción

Se formulará una pregunta que permita obtener una respuesta específica sobre las áreas de acción de la Ingeniería Mecatrónica.

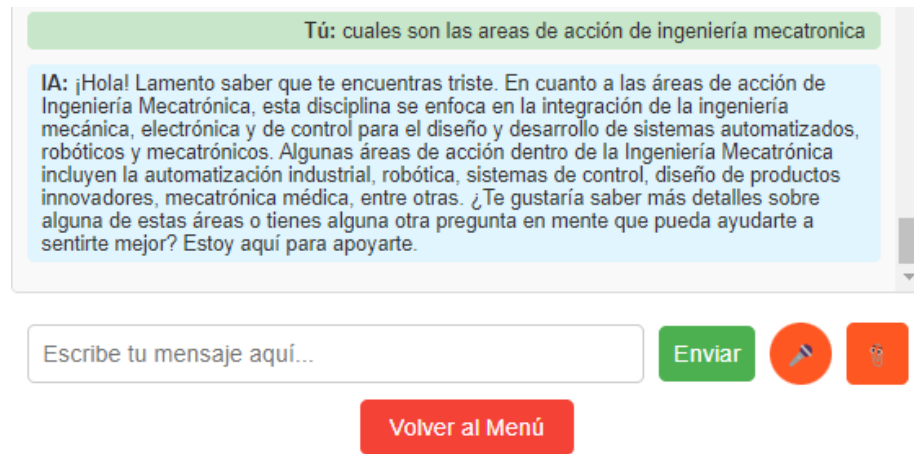


Figura 54: Pregunta relacionada con el Departamento de Mecatrónica

Ahora se hará una pregunta específica, tanto la pregunta como la respuesta se realizarán de forma hablada. La pregunta realizada fue: "¿Quién es el director de la carrera de Ingeniería Mecatrónica?". El resultado se puede observar en la Figura 55:

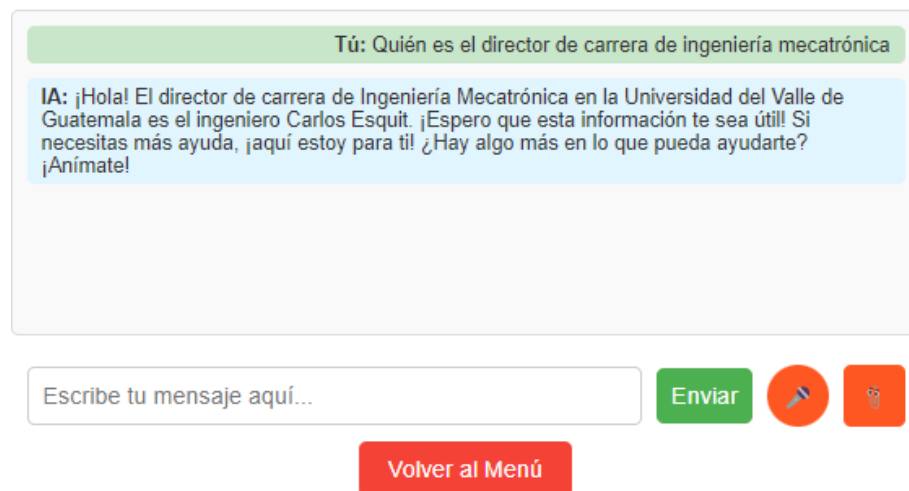


Figura 55: Pregunta específica, director de carrera

Se utilizará la herramienta de Windows, para poner subtítulos de los sonidos emitidos por el dispositivo, así se puede observar la correcta forma de hablar del programa.



Figura 56: Subtítulos generados de los sonidos emitidos

Como se puede observar, la herramienta comenzó a transcribir la respuesta proporcionada por la extensión de ChatGPT. A continuación, se demostrará cómo funciona la conversación en otro idioma.

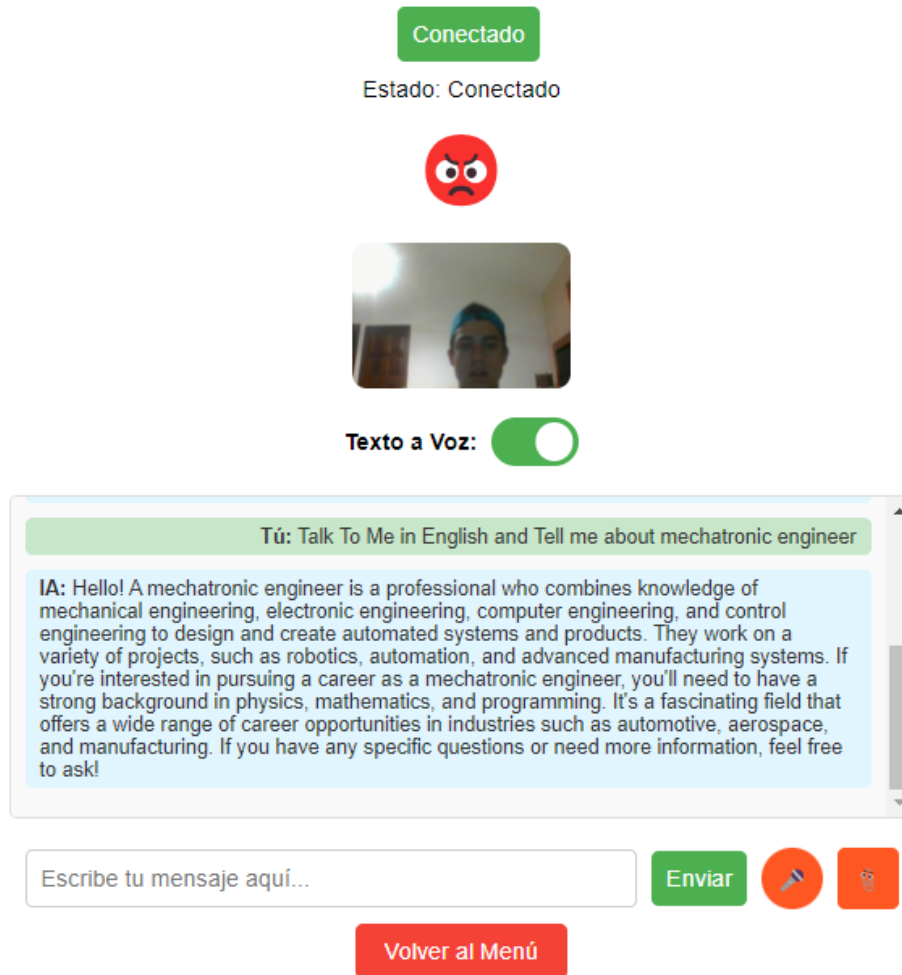


Figura 57: Conversación en otro idioma

Por último, se realizó una pregunta general con el objetivo de obtener una respuesta relacionada con la Ingeniería Electrónica.



Figura 58: Pregunta relacionada con el Departamento de Electrónica

Se observa cómo, tras múltiples preguntas dirigidas a la inteligencia artificial, esta respondió a todas utilizando tanto la emoción presentada por el usuario como la información almacenada con la que fue entrenada, además de los conocimientos base que posee la herramienta. Se puede concluir que el funcionamiento conjunto es el correcto.

- Tras una evaluación exhaustiva, se determinó que Google Chrome es el navegador más adecuado para este proyecto. Su conjunto de características y permisos habilitados garantizan un funcionamiento óptimo y completo de todas las funcionalidades del sitio web.
- Para un rendimiento óptimo de la función de 'conversación en vivo', se recomienda un entorno de trabajo tranquilo y bien iluminado. Estas condiciones favorecen la concentración y permiten una interacción más efectiva entre el usuario y el sistema.
- La implementación de la detección de emociones en las conversaciones ha demostrado ser una estrategia efectiva para ofrecer una experiencia de usuario más personalizada y empática. Al superar las respuestas genéricas típicas de los chatbots, se establecen interacciones más naturales y satisfactorias.
- La función 'Conectar con IA' garantiza una actualización continua de la extensión de ChatGPT, asegurando que siempre disponga de la última información puesta en el archivo .json.
- La capacidad del programa de admitir tanto español como inglés amplía significativamente su alcance, permitiendo su uso por un público más diverso, incluyendo visitantes extranjeros.

- Se recomienda utilizar el sistema operativo Windows para garantizar una óptima compatibilidad entre Python y el entorno de trabajo.
- Adoptar un enfoque parcial en la instalación de Python, instalando únicamente las librerías estrictamente necesarias, ofrece mayor flexibilidad y eficiencia. Esto evita conflictos de versiones y reduce el consumo de recursos del sistema.
- Se recomienda utilizar Google Chrome como navegador para la ejecución de este proyecto, ya que la compatibilidad con diversas tecnologías y su configuración por defecto permiten aprovechar al máximo todas las funcionalidades del sistema.
- Verificar la instalación correcta de todas las librerías requeridas para garantizar un funcionamiento adecuado del proyecto.
- Ante la presencia de errores relacionados con las librerías, se recomienda verificar la compatibilidad entre las versiones de Python y las librerías utilizadas. En caso de incompatibilidad, se sugiere actualizar a la segunda versión más reciente de la librería, evitando siempre la versión más reciente para minimizar posibles problemas de estabilidad.

-
- [1] E. Estudio, *¿Qué es un microcontrolador?* Micro, 2024.
 - [2] D. S. M. Guzmán, *Implementación de una inteligencia artificial capaz de crear una conversación en la plataforma del rostro animatrónico*, Tesis de licenciatura, Universidad del Valle de Guatemala, 2023.
 - [3] J. F. L. Salguero, *Implementación de una interfaz de uso y control para el rostro animatrónico de la Universidad del Valle de Guatemala*, Tesis de licenciatura, Universidad del Valle de Guatemala, 2023.
 - [4] G. Ledezma y grupo, *Brazo Robótico de 3 Grados de Libertad con Inteligencia Artificial*, Tesis de licenciatura, Instituto Tecnológico de Lázaro Cárdenas, 2016.
 - [5] P. Marín, *Síntesis de voz y reconocimiento del habla. Implementación en el robot HOAP-3*, Licenciatura en Ingeniería de Sistemas y Automática, Universidad Carlos III de Madrid, 2012.
 - [6] Y. Fernández, *ChatGPT: qué es, cómo usarlo y qué puedes hacer con este chat de inteligencia artificial GPT*, Investigación sobre ChatGPT, 2024.
 - [7] J. Fetzer, *What is Artificial Intelligence?* IAFetzer, 1990.
 - [8] F. Anjila, *Artificial Intelligence*, IA, 2021.
 - [9] J. Murphy e I. Naqa, *What Is Machine Learning?* Machine Learning, 2015.
 - [10] D. Matich, *Redes Neuronales: Conceptos Básicos y Aplicaciones*, RedNeu, 2001.
 - [11] G. Workspace, *Comprender los tokens GPT de OpenAI: Una guía completa*, Definición de Tokens en Chat GPT, 2024.
 - [12] E. C. Navone, *Python leer archivo JSON*, JSON1, 2022.
 - [13] Ionos, *¿Qué es una interfaz gráfica de usuario (GUI)?* IntGra, 2021.
 - [14] O. team, *About OpenCV*, OPENCV, 2024.
 - [15] J. D. Muñoz, *About OpenCV*, FLASK, 2017.
 - [16] M. Coppola, *JSON para principiantes*, JSON2, 2022.
 - [17] C. Merchán, *Introducción a socket.io*, SCK1, 2015.

- [18] Socket.IO, *Socket.IO. Bidirectional and low-latency communication for every platform*, SCK2, 2024.
- [19] mdn, *Conceptos básicos de HTML*, HTML1, 2024.
- [20] Vadavo, *¿Qué es HTML y para qué sirve?* HTML2, 2023.
- [21] AWS, *What is an API (Application Programming Interface)?* API1, 2024.
- [22] O. AI, *OpenAI API*, OPENAI11, 2020.
- [23] GSI, *CAMI APP Test*. CAMI, 2023.
- [24] B. Industrial, *bind api banco*, BIGT, 2024.
- [25] SYDLE, *¿Qué es API? Ejemplos, ventajas y tipos*, Tipos_APIs, 2023.
- [26] OpenAI, *What are tokens and how to count them?* TOKENS, 2024.
- [27] V. Group, *¿Por qué es importante la planificación de proyectos?* Frase1, 2023.
- [28] FastAPI, *FastAPI*, FastAPI, 2024.