

Asistente multifunción con IA Oreo Bot

Multifunctional assistant with AI Oreo Bot

Fabricio Rolando Rivadeneira Zambrano  ORCID, Pedro Fernando Valdez Delgado, 
ORCID, Moise Julián Zambrano Loo  ORCID, Braulio Yovanny Valencia Zambrano 
ORCID, Marcos Tulio Zambrano Zambrano  ORCID

Universidad Laica Eloy Alfaro de Manabí Extensión Chone, Manabí, Ecuador

RESUMEN

La inteligencia artificial (IA), nace como una tecnología emergente que ha revolucionado e impactado en las actividades que comúnmente realizamos. La IA se encuentra en la actualidad desde los teléfonos inteligentes hasta infraestructuras de las empresas, aportando de forma significativa en la automatización de procesos y ayuda a la toma de decisiones, por su capacidad de aprender y mejorar con el tiempo. Para la realización del proyecto se planteó como objetivo Construir un asistente virtual multifunción con IA, para lograr el objetivo propuesto se utilizó la metodología SCRUM, la cual permitió organizar los procesos y ejecutar las actividades de forma óptima. OREO BOT, cuenta con tres funciones: Asistente virtual, traducción en tiempo real y geolocalización, para lo cual se utilizó la API de Openai para llevar a cabo la función de Asistente, además se usaron otras APIs como la de Google Translate y Google Maps para las otras dos funciones de OREO BOT.

PALABRAS CLAVES: inteligencia artificial, asistente virtual, metodología SCRUM, traducción en tiempo real, geolocalización

ABSTRACT

Artificial intelligence (AI) was born as an emerging technology that has revolutionized and impacted the activities we commonly carry out. AI is currently found from smartphones to company infrastructures, contributing significantly to the automation of processes and helps decision making, due to its ability to learn and improve over time. To carry out the project, the objective was to build a multifunction virtual assistant with AI. To achieve the proposed objective, the SCRUM methodology was used, which allowed the processes to be organized and the activities to be executed optimally. OREO BOT has three functions: Virtual Assistant, real-time translation and geolocation, for which the OpenAI API was used to carry out the Assistant function, in addition other APIs such as Google Translate and Google Maps were used to the other two functions of OREO BOT.

KEY WORDS: artificial intelligence, virtual assistant, SCRUM methodology, real-time translation, geolocation

INTRODUCCIÓN

En la era de la inteligencia artificial (IA), los asistentes virtuales representan una convergencia innovadora de tecnología avanzada y accesibilidad humana. Este artículo explora las capacidades y aplicaciones de un asistente virtual de IA, diseñado para interactuar de manera natural y eficiente con los usuarios. A través del aprendizaje automático y el procesamiento del lenguaje natural, estos sistemas no solo responden a consultas, sino que también anticipan necesidades y adaptan sus respuestas en función del contexto y las preferencias individuales. El asistente virtual OREO BOT es un prototipo que promete transformar la forma en que interactuamos con la información y gestionamos nuestras tareas diarias, marcando un hito significativo en la evolución de la interfaz hombre-máquina.

Para el desarrollo del Oreo Bot se utilizó la metodología Scrum por ser un marco ágil para la gestión de proyectos que se centra en entregar resultados de alta calidad de manera rápida y eficiente. Scrum permitió transformar la forma del trabajo en equipo, mejorando la productividad, la calidad del producto. La revisión y retroalimentación constante en cada sprint ayudaron a identificar y corregir problemas de calidad rápidamente.

Oreo Bot es un asistente virtual que utiliza varias APIs avanzadas para proporcionar una experiencia interactiva y funcional. Para implementar la función de asistente virtual, se utilizó la API de OpenAI. Además, se integró la API de Google Translate, lo que permitió la traducción en línea a varios idiomas. Para la función de geolocalización, se utilizó la API de Google Maps. El proyecto también incluyó la creación de un avatar para mejorar la interfaz de interacción, que se implementó en una placa Raspberry Pi con una pantalla táctil GeekPI de 10,1 pulgadas, permitiendo el desarrollo de la parte lógica del proyecto.

En la parte de robótica, se emplearon componentes como un chasis de coche de robot de metal inteligente con ruedas mecanum de 3.89 pulgadas y una placa Arduino para la programación del sistema de movimiento. Un sensor de proximidad se integró para evitar obstáculos. Al chasis se le añadió un brazo de metal con una base para instalar una pantalla y el case de la placa Raspberry Pi, completando así la estructura del Oreo Bot.

MARCO TEÓRICO

La Inteligencia artificial es una de las tecnologías emergentes, si bien no es de ahora, sus funciones y exploración en nuevos campos es lo que lo hace una tendencia, (Martínez-Comesaña et al., 2023) agrega que, en el campo educativo, la IA se centra en potenciar en gran medida elementos o herramientas educativos, como el aprendizaje personalizado, las evaluaciones adaptables, los sistemas de tutorías inteligentes, la calificación automatizada, la realizada virtual y la realidad aumentada en la educación. (Medina Gamero, 2024) agrega en el campo de la medicina que, la integración de la inteligencia artificial, la realidad virtual y el big data en la enfermería conduce a un entorno más personalizado, eficaz, predictivo, preventivo y participativo.

En la investigación, el uso de esta herramienta comprende grandes desafíos, (Gimeno-Ballester & Trigo-Vicente, 2024) explican que, si bien las herramientas de IA prometen eficiencia y rapidez en la generación de artículos, también conllevan riesgos significativos, por su parte (Galiana, Gudino, & González, 2024) menciona que la aplicabilidad en el desarrollo ético de IA asegura que los sistemas de inteligencia artificial se alineen con los valores humanos y principios éticos, finalmente (Quirós-Fons & García-Ull, 2023) complementa que, la dependencia excesiva de la tecnología, es un desafío ético importante con la integración de IA en la investigación.

Las redes neuronales artificiales, según (J.F. Avila-Tomás, Mayer-Pujadas, & Quesada-Varela, 2020) están compuestas por elementos que se comporten de forma similar a la neurona biológica en sus funciones más comunes y que se denominan elemento procesador. Añadiendo, (Basález & Mora, 2022) sustenta que, la búsqueda heurística se basa en representar el conocimiento implícito o procedimental que poseen los seres humanos de forma explícita. (Mayer, 2023) agrega que, en el 2022, la empresa OpenIA puso a disposición de todos los usuarios un sistema de chat basado en el modelo de lenguaje por inteligencia artificial llamado GPT-3.5 o ChatGPT.

Los modelos de lenguaje a gran escala, según (Haque, Brito, & Frade, 2024) son modelos generativos preentrenados, los GPT utilizan el aprendizaje no supervisado en grandes volúmenes de datos textuales, por su parte (Santos, Fernandes, Lima, Linhares, & Palha, 2024) agrega explicando que, este principio es fundamental en la aplicación de asistentes virtuales, ya que estimula una interacción bidireccional donde no solo se recibe información, sino que también se interactúa. Sintetizando (Martinez, 2024) difiere del libro “Deep Learning” que lo autores (Heaton et al., 2018) definen la inteligencia artificial como un campo de la informática que se centra en el diseño de sistemas que pueden aprender de los datos.

La potencial integración de la IA según (Alcántara, 2024), tienen la posibilidad de personalizar la experiencia del aprendizaje, adaptándola bien a las necesidades individuales para brindar retroalimentación instantánea y personalizada. De acuerdo con el autor anterior, (Ponce, 2024) argumenta que, la visualización de datos basadas en IA puede ser útil para comunicar hallazgos de manera clara y efectiva. (Briva-Iglesias, 2023) evidencia dos tipos de inteligencia artificial: IA débil o narrow IA, está diseñada para hacer una tarea específica o un conjunto de tareas relacionadas, por su parte la IA general o IAG pretende copiar las habilidades cognitivas de los seres humanos.

El uso de algoritmos permite que estos modelos o lenguajes logren más asertividad con sus resultados, explican (Hernández & Herrera, 2021) que, el uso de un algoritmo permitirá describir de forma sucesiva un sistema complejo de relaciones a partir de operaciones matemáticas, complementando que las redes neuronales artificiales se asemejan a las redes neuronales biológicas al utilizar nodos, mismas que se usan en la implementación de sistemas de inteligencia artificial. (C. R. Domínguez Mayorga, Ramos Velasco, Ramos Fernández, & Hernández, 2012) agrega un algoritmo denominado wavelets donde la primera es una red de tres capas donde las neuronas de la entrada y salida son elementos lineales y la función de activación de la capa oculta es una wavelet madre, lo cual obtienen gradualmente el número de neuronas.

Un aspecto importa lo mencionan (Ruiz & Velásquez, 2023), los datos juegan un papel central para lograr un entrenamiento correcto, pues si no están limpios y consolidados se verá afectado el algoritmo y se verá afectado su capacidad de aprendizaje. (Gallardo, 2023) difiere que ChatGPT es capaz de retener más información y centrarse en lo más importante de un texto a través de una capa de su algoritmo desarrollado en 2017 llamado transformer, es por ello por lo que la IA ha crecido sustancialmente debido a su capacidad algorítmica para hacer recomendaciones, predicciones, decisiones ya prender en diferentes contextos.

En contextos generales, (Gutiérrez & López, 2022) indican que la información son datos interpretados por algún sujeto. De la interpretación se conforma el conocimiento, que es la base del actuar inteligente. (Glück & Yokoyama, 2023) destaca que, los lenguajes lógicos superan la direccionalidad de los lenguajes convencionales al no dar preferencia a las direcciones de computación y al especificar entre la entrada y salida. Es por ello por lo que (Rubín, 2024) refiere que, la IA está transformando al mundo tal y como lo conocíamos hasta ahora, creando un futuro que impactara a la sociedad humana de una manera que antes solo podíamos imaginar.

La relación entra la tecnología y humanos han dado origen a fronteras inquebrantables (Barraca-Mairal, 2024) indica que, la IA y la tecnología han de acertar, pensar y promover nuestra sensibilidad moral más básica, la ética, mas no a suplantarla. (Pearson, 2023) expone que ChatGPT tiene una opción que permite evaluar textos generados por personas para identificar datos erróneos o incongruentes, así mismo (Alvarez, 2023) difiere que la utilización de ChatGPT para corregir errores o generar ideas puede liberar tiempo para que las personas se concentren en el pensamiento crítico y el análisis de conceptos complejos.

No todo es beneficioso, (Jesús, 2024) expone varias limitación o desventajas que presenta esta inteligencia artificial, una de ellas es la generación de texto incoherente en ciertas situaciones, otra es no mantener la fluidez ya que no siguen el hilo de la conversación, (Reyes, 2023) de acuerdo con el autor agrega que otras de las limitaciones de esta IA es que su base del conocimiento ciñe hasta el 2021, por ello el mismo autor remienda a los usuarios contrastar y contextualizar la información que reciben. A diferencia de los autores anteriores, (Soage, 2022) agrega resultados favorables como los escritos de alta calidad, la realización de tareas de razonamiento y contenido de mayor longitud, gracias al correcto entrenamiento de los datos.

MATERIALES Y MÉTODOS

Para la realización de Oreo Bot, se emplearon tanto materiales de hardware como de software. La metodología utilizada fue Scrum, según (Mayo-Álvarez, y otros, 2024) se basa en ciclos cortos, interactivos y colaborativos llamados sprint, en los cuales los productos se desarrollan de manera incremental y se prioriza a través de eventos, artefactos y roles. Lo que conllevó una planificación, en su etapa inicial se definió los requisitos y funcionalidades para la implementación de OREO BOT, seguidamente se realizaron Sprint con una duración de dos semanas, al final de cada Sprint se producía un incremento en funcionalidades de OREO BOT. Se realizaron Daily Stand-ups Lo que permitió identificar obstáculos y planificar el trabajo del día siguiente, al final de cada sprint se realizaban revisiones de las funcionalidades y obtener una retroalimentación.

LOS MATERIALES DEL HARDWARE:

Raspberry Pi: Raspberry Pi 4 Model B, con un procesador Broadcom BCM2711, su arquitectura está compuesta por un Quad-core Cortex-A72 (ARM v8) 64 bit SoC @ 1.5GHz, una memoria RAM de 8GB LPDDR4-3200 SDRAM, dispone de conectividad de red Ethernet “Gigabit Ethernet”, un Wifi disponible de 2.4 GHz y 5.0 GHz estandarizado por la IEEE 802.11ac, adicional un bluetooth 5.0, BLE, dispone de varios puertos que son: 2 USB 3.0 y 1 USB 2.0, dos entradas para video micro HDMI con soporte hasta 4K, así mismo, una salida de audio estéreo y video compuesto. En su almacenamiento, disponemos de una microSD kingstong de 64 GB clase 10, los gráficos que proporciona en su GPU, un VideoCore VI 3D con OpenGL ES 3.0 graphics.

Arduino: Mega 25-60

Chasis del Robot: La plataforma de movimiento del chasis de este coche utiliza ruedas mecanum, permitiendo un movimiento omnidireccional flexible. Es ideal para proyectos de enseñanza científica y competiciones de robots. Las características incluyen una placa de aluminio resistente y artística, y compatibilidad con la mayoría de los controladores. El chasis, hecho de aleación de aluminio, mide 13 x 11.42 x 3.14 pulgadas. Las ruedas mecanum de 97 mm están hechas de plástico de alta calidad y goma, con un peso de 150 g cada una, un diámetro exterior de 3.82 pulgadas, un grosor de 1.77 pulgadas y una capacidad de carga de 0-10 kg para las cuatro ruedas. El motor de CC de 12V tiene una potencia de 4.8W, una corriente nominal de 360mA, una velocidad de 110000 rpm y una relación de reducción de 1:30. La lista de envío incluye un marco de metal, cuatro ruedas mecanum de 97 mm, cuatro motores de CC de 37, y un juego de tornillos.

Brazo de Metal: Une la base y la pantalla a una altura de 1,20 metros y en su parte superior una base para colocar la pantalla.

Pantalla: Cuenta con una resolución de 1024 * 600 pixeles, el tipo de pantalla es IPS lo que le proporciona colores brillantes y amplios ángulos de visión, dispone de una tasa de frecuencia de 60Hz lo que asegura una visualización fluida y sin interrupciones, la pantalla es compatible con Raspberry Pi 5, 4B, 3B+, 3B, B+, Zero, 400; Jetson Nano; Banana Pi M5/M2 Zero; VisionFive. En sus entradas disponemos de un HDMI para video, USB/C para la alimentación. El paquete incluye la pantalla LCD de 10.1 pulgadas, 1 cable USB a USB-C, 1 cable HDMI completo, 1 adaptador HDMI a Micro HDMI y 2 soportes para la pantalla.

Sensores: Sensores de proximidad ultrasónicos para la detección de obstáculos permitiendo la libre movilidad de la base del robot y actuadores

Otros Componentes: Para la interacción de los usuarios se usó unos auriculares Pulz 70w de Scorpion, entre sus modos de conexión tenemos: Cableado, Bluetooth a 2.4G inalámbrico, es compatible con PC, consolas de videojuegos. Su altavoz es de 40 mm, su tipo de sonido estéreo con graves potentes y su frecuencia es de 20 Hz–20 KHz, cubre el rango completo de audición humana, proporcionando una calidad de sonido clara y nítida, con un micrófono omnidireccional desmontable de alta sensibilidad, su ergonomía tiene un diseño ligero para una experiencia cómoda, también una iluminación RGB que da un aspecto estético atractivo.

FALTA (Baterías, cables, motores, etc) A NIVEL DE ESTRUCTURA

LOS MATERIALES DEL SOFTWARE:

Sistema Operativo: Raspberry Pi OS utilizada en el sistema operativo oficial, con una distribución en Debian optimizada para el hardware de la Raspberry.

Lenguajes de Programación: Python, C++ (u otros).

APIs: OpenAI API: Entre las APIs utilizadas para este proyecto tenemos la de OpenIA en sus modelos GPT-3 y GPT-4 cada una con diferentes versiones, donde se detalla:

- Versiones de la API
 - GPT-4 Lanzada en 2023, mejorado en comprensión y generación de texto. En esta versión disponemos de dos modelos:
 - GPT-4-8K: Con un límite de 1024 tokens por solicitud
 - GPT-4-32K: Con un límite de 32768 tokens por solicitud, diseñado para tareas que requieren contextos largos

En cuanto a la funcionalidad se utilizó llamadas POST usadas obtener las respuestas y luego transformar el texto (respuesta) en audio para la salida logrando una mejor interacción del sistema. Para lograr la llamada de la API, se creó un entorno en Python, que en el CMD del sistema se colocaron las siguientes líneas de comandos:

- `python -m venv myenv`: Crea el entorno virtual
- `Myenv\Scripts\activate`: Activa el entorno virtual

Así mismo, el uso de la librería de OpenIA es fundamental para ello en la terminal se colocó “`pip install request openia`”

Para implementar la funcionalidad de traducción en tiempo real en un asistente virtual utilizando la API de Google Translate en Python, se requirieron varios materiales de software. A continuación, se detallan los principales:

Google Cloud SDK

- **Descripción:** Es un conjunto de herramientas que permite interactuar con los servicios de Google Cloud desde la línea de comandos.

Google Translate API

- **Descripción:** Es un servicio proporcionado por Google Cloud que permite realizar traducciones automáticas de textos entre múltiples idiomas. La API fue el núcleo de la funcionalidad de traducción en tiempo real.

Biblioteca os de Python

- **Descripción:** Es una biblioteca estándar de Python que proporciona una forma de interactuar con el sistema operativo, como el manejo de variables de entorno.

Archivo de Credenciales JSON

- **Descripción:** Este archivo es generado por Google Cloud y contiene la información necesaria para autenticar y autorizar las solicitudes de la aplicación Python a la API de Google Translate.

La API de Google Maps es un conjunto de servicios web que permite a los desarrolladores integrar diversas funcionalidades de Google Maps en sus aplicaciones y sitios web. Aquí hay una descripción detallada de las principales características y funcionalidades de la API de **Google Maps:**

Características Principales

Maps JavaScript API: Permite incrustar mapas interactivos en sitios web.

Ofrece una amplia gama de controles de mapa (zoom, tipos de mapa, capas personalizadas). Soporta la creación de marcadores, infowindows, overlays, y más.

Incluye funciones de geocodificación (conversión de direcciones en coordenadas geográficas y viceversa).

Places API: Permite a las aplicaciones buscar lugares y negocios cercanos.

Proporciona información detallada sobre lugares, como reseñas, fotos, y horarios de apertura. Soporta la búsqueda de lugares mediante texto, autocompletado de direcciones y búsqueda por coordenadas geográficas.

Geocoding API: Convierte direcciones en coordenadas geográficas (geocodificación) y coordenadas en direcciones (geocodificación inversa).

Útil para obtener ubicaciones precisas a partir de direcciones de texto.

Directions API: Proporciona rutas detalladas para conducir, caminar, andar en bicicleta y transporte público.

Incluye información sobre el tráfico en tiempo real y rutas alternativas.

Distance Matrix API: Calcula las distancias y tiempos de viaje entre múltiples orígenes y destinos. Útil para aplicaciones que requieren planificación de rutas y optimización logística.

Elevation API: Proporciona datos de elevación para ubicaciones especificadas.

Útil para aplicaciones que requieren información topográfica.

Time Zone API: Devuelve la zona horaria para cualquier ubicación geográfica especificada.

Incluye información sobre el horario de verano y diferencias horarias.

Roads API:

Permite ajustar ubicaciones de GPS a la carretera más cercana y proporciona datos sobre el tramo de carretera.

Útil para aplicaciones de seguimiento de vehículos y navegación.

Usos Comunes

Desarrollo de Aplicaciones de Navegación: Integrar mapas y direcciones en aplicaciones de conducción, transporte público, y rutas a pie o en bicicleta.

Aplicaciones de Localización de Negocios: Mostrar ubicaciones de negocios y puntos de interés cercanos, junto con información detallada.

Optimización Logística: Planificar y optimizar rutas para la entrega de bienes y servicios.

Aplicaciones de Seguimiento: Monitorizar vehículos y personal en tiempo real.

Aplicaciones de Turismo: Proporcionar información sobre lugares de interés y atracciones turísticas.

Requisitos

Clave de API: Para utilizar la mayoría de los servicios de Google Maps API, se requiere una clave de API que se obtiene a través de Google Cloud Platform.

Facturación: Aunque muchas funcionalidades tienen un nivel gratuito, el uso extensivo puede generar costos que se facturan a través de la cuenta de Google Cloud.

La API de Google Maps es una herramienta poderosa y versátil para desarrolladores que buscan integrar funcionalidades avanzadas de mapas y localización en sus aplicaciones y sitios web.

Otros recursos: Bibliotecas y frameworks: La lista de las librerías utilizadas son:

- Request 2.31.0
- Pygame 2.5.2
- Speech_recognition 3.10.4
- Googletrans 4.0.0rc1
- Movie 1.0.3
- Otras adicionales pero que no requieren instalación son: threading, time, io y subprocess

Documentación y Manuales: Para realizar este trabajo, se usó como guía la documentación que proporciona la página de donde obtenemos la API que es OpenIA así mismo para la APIs de Google en conjunto de su documentación.

PARA LA REALIZACIÓN DE OREO BOT SE SIGUIERON LAS SIGUIENTES FASES

CONFIGURACIÓN DEL HARDWARE:

Montaje del Chasis y Brazo: Se unió la base que realiza el movimiento y por su parte en la parte superior del brazo, un espacio para la pantalla, en la parte interna del brazo, tendrá el cableado necesario.

Conexión de Componentes: La comunicación de estos será a través de una línea de código en donde mandará una señal al Arduino y este lo recibirá por los pines libres, para así limitar el movimiento del robot y permitir su avance

DESARROLLO DEL SOFTWARE:

Configuración del Entorno de Desarrollo: Primero se descargó la imagen de Raspbian desde el sitio oficial de Raspberry Pi. Luego, se utilizó la herramienta Raspberry Pi Imager para grabar la imagen en una tarjeta microSD. Se insertó la tarjeta microSD en la Raspberry Pi, se conectaron un teclado, un ratón, un monitor y la fuente de alimentación. Al encender la Raspberry Pi, se siguieron las instrucciones en pantalla para completar la configuración inicial del sistema operativo, como la configuración de la red y la actualización del software.

Integración de APIs: Para la integración de la API proporcionada de OpenIA se usó una estructura de código la cual se puede hallar en la documentación de la API, donde da una idea de cómo se puede implementar su servicio con las llamadas POST para obtener los recursos necesarios. Para instalar la API de Google Maps en Python, primero, se creó un proyecto en Google Cloud y se habilitó la API de Google Maps. Luego, se procedió a generar una clave de API en la sección de credenciales de Google Cloud Console. Luego se Instaló la biblioteca google maps en el entorno de Python utilizando el comando pip install google maps. Finalmente, se importó la biblioteca google maps en el script de Python y seguido la

configuración de la clave de la API para comenzar a realizar solicitudes a los servicios de Google Maps.

Programación del Movimiento del Robot: El movimiento del Robot se permite usando 4 motores colocados en las llantas que permiten el arranque; los cuatro motores están conectados a un módulo y al Arduino, donde se envía la señal para que funcionen sincronizadamente.

Desarrollo de la Interfaz de Usuario: La interfaz del usuario es permitida por la visualización de una ventana la cual contiene el avatar, para ello se usó dos archivos .mp4, uno donde simule que habla y otro en movimiento, dentro de la estructura del código, existe un identificador que intercala los videos según se vaya interactuando.

Pruebas y Validación

Pruebas de Funcionalidad: Según los Sprint, se revisaba cada dos semanas de haber implementado algún mejora o funcionalidad, con el objetivo de verificar si se cumplían o no las especificaciones indicadas

Pruebas de Movimiento: Se realizó pruebas donde se evaluaba si la dirección de las llantas iba en varios sentidos, con el objetivo de que, en la puesta de acción pueda realizar los respectivos movimientos.

RESULTADOS

Para la integración del avatar, usamos dos recursos multimedia, que hacen el cambio según el estado de la API, es por ello por lo que estos videos están dentro de los archivos del programa:

```

1 video_mouth_closed = VideoFileClip('video_mouth_closed.mp4').resize((screen_width, screen_height)) # type: ignore
2 video_mouth_open = VideoFileClip('video_mouth_open.mp4').resize((screen_width, screen_height)) # type: ignore

```

Para la llamada a la API tenemos la siguiente estructura, la cual está compuesta por varios factores, uno de ellos es el uso de URL que hacen la llamada POST:

```

1 url = "https://api.openai.com/v1/chat/completions"
2 url = "https://api.openai.com/v1/audio/speech"

```

La integración de diversas APIs fue fundamental para dotar a Oreo Bot de sus capacidades avanzadas. La API de OpenAI fue crucial para el desarrollo del asistente virtual, proporcionando respuestas precisas y coherentes a las consultas de los usuarios. La API de Google Translate permitió traducciones multilingües con una alta precisión, y la API de Google Maps ofreció geolocalización precisa y en tiempo real, es por ello que se muestra la estructura para usar la API de OpenIA y la forma en la que tiene que hacer la llamada POST:

```

1  api_key = "sk-proj-sKZKYivzEThXqmYEV22aT3B1bkFJt91jfhUW65WfHh0Rae30"
2  headers = {
3      'Authorization': f'Bearer {api_key}',
4      'Content-Type': 'application/json'
5  }

```

Una vez se logra la llamada de la API, tenemos que obtener las respuestas que se logra con la siguiente estructura de código, es importante destacar que se usan en relaciona las URL ya que una es encargada de las respuestas en texto y la otra URL se encarga de crear y generar el audio, para que salga como respuesta hablada juntamente con los archivos multimedia cargados en la parte superior y así intercalar cuando están activos y desactivado el asistente.

```

1  def enviar_a_chat_gpt(input_text):
2      url_chat = "https://api.openai.com/v1/chat/completions"
3      model = "gpt-4"
4      headers = {
5          "Content-Type": "application/json",
6          "Authorization": f"Bearer {api_key}"
7      }
8      data = {
9          "model": model,
10         "messages": [
11             {"role": "user", "content": input_text},
12             {"role": "assistant", "content": ""}
13         ]
14     }
15     try:
16         response = requests.post(url_chat, headers=headers, json=data)
17         if response.status_code == 200:
18             return response.json()["choices"][0]["message"]["content"]
19         else:
20             print(f"Error al realizar la solicitud: {response.status_code} - {response.text}")
21             return None
22     except requests.exceptions.RequestException as e:
23         print(f"Error de conexión: {e}")
24         return None

```

Así mismo el audio, luego de obtener la respuesta de la API, ese texto se transforma a audio y obtenemos:

```

1 def generar_audio(text, model='tts-1', voice='nova', speed=0.95, format='opus'):
2     headers = {
3         "Content-Type": "application/json",
4         "Authorization": f"Bearer {api_key}"
5     }
6     data = {
7         'model': model,
8         'input': text,
9         'voice': voice,
10        'speed': speed,
11        'response_format': format
12    }
13    time.sleep(2)
14    response = requests.post(url, headers=headers, json=data)
15    if response.status_code == 200:
16        if response.headers['Content-Type'] == 'audio/opus':
17            print("Longitud de la respuesta:", len(response.content))
18            return io.BytesIO(response.content)
19        else:
20            print("Contenido inesperado:", response.headers['Content-Type'])
21    else:
22        print(f"Error al generar el audio: {response.status_code} - {response.text}")
23    return None

```

Para la integración del Api de Google Maps una codificación la cual realiza una operación inversa (obtener la dirección a partir de las coordenadas geográficas).

```

1 from geopy.geocoders import GoogleV3

```

Se procede a la llamada del API de Google Maps agregando así la clave correspondiente se generó.

```

1 def get_location_info(location):
2     geolocator = GoogleV3(api_key='AIzaSyA0l8z4-0L6pI7GXCJNUXT60oq_au53K0')
3     location = geolocator.geocode(location)
4     if location:
5         return location.address, (location.latitude, location.longitude)
6     else:
7         return None, None

```

DESAFÍOS

SOFTWARE

La integración de múltiples APIs resultó en problemas de latencia, afectando significativamente el tiempo de respuestas del sistema. Se trató de reducirlas mediante la optimización del código y la arquitectura del sistema. Sin embargo, sigue siendo un área crítica que requiere mejoras continuas para asegurar una experiencia de usuario fluida y eficiente, y a su vez evitar que los tiempos de respuestas sea más corta y precisas.

La dependencia externa esta anexada a la latencia, ya que es clave identificar las limitaciones que cada uno de los servicios nos ofrece. Si estos servicios experimentan interrupciones o cambios en su funcionamiento, pueden afectar el desempeño y la fiabilidad del sistema. Es esencial desarrollar estrategias de mitigación, como redundancia y alternativas locales, para minimizar estos riesgos.

La creación de un avatar personalizable y adaptable a las respuestas de la API es un desafío considerable. La capacidad de personalizar el avatar no solo mejora la experiencia del usuario, sino que también requiere una integración avanzada con la API para garantizar respuestas coherentes y relevantes.

HARDWARE

Uno de los principales desafíos es asegurar una tracción y adherencia adecuadas en diversas superficies. Diferentes tipos de terreno pueden afectar la capacidad del robot para moverse eficientemente, especialmente en superficies resbaladizas o irregulares. Mantener la estabilidad y el equilibrio del robot, especialmente en terrenos accidentados o durante maniobras complejas, es crucial. La distribución del peso y el diseño del chasis juegan un papel importante en este aspecto.

Las llantas y otros componentes mecánicos están sujetos a desgaste y daños, especialmente en entornos exigentes. Asegurar la durabilidad de estos componentes y facilitar el mantenimiento son desafíos importantes. Los robots con llantas deben interactuar con su entorno, incluyendo superar obstáculos, subir pendientes y adaptarse a cambios en el terreno. Desarrollar sistemas de suspensión y control avanzados es crucial para enfrentar estos desafíos.

CONCLUSIONES

La inteligencia artificial (IA) proporciona beneficios clave en diversos campos, destacándose la automatización de tareas repetitivas, lo que incrementa la eficiencia y reduce errores humanos, permitiendo a las organizaciones enfocar recursos en actividades estratégicas. Además, la IA facilita el análisis de grandes volúmenes de datos en tiempo real, proporcionando insights valiosos para decisiones estratégicas informadas. Este análisis avanzado permite identificar tendencias y oportunidades ocultas. Asimismo, la IA impulsa la innovación y el desarrollo tecnológico al resolver problemas complejos de manera eficiente, fomentando la creación de nuevos productos y servicios y abriendo nuevas posibilidades para futuras aplicaciones tecnológicas.

Los asistentes virtuales potenciados por IA aumentan la productividad al gestionar múltiples tareas simultáneamente, permitiendo a los usuarios enfocarse en actividades de

mayor valor. Su disponibilidad 24/7 garantiza asistencia continua, trascendental en atención al cliente y soporte técnico. Así mismo la personalización avanzada, adaptándose a las preferencias del usuario y proporcionando respuestas más relevantes, lo que mejora la experiencia del usuario y optimiza la eficiencia operativa al anticipar necesidades y ofrecer recomendaciones proactivas.

La combinación de Scrum con tecnologías avanzadas ha demostrado ser una estrategia eficaz para el desarrollo de Oreobot. Los desafíos enfrentados y superados durante el proyecto proporcionan valiosas lecciones para futuros desarrollos en el campo de la robótica y la inteligencia artificial. A medida que la tecnología avanza, la integración de metodologías ágiles y herramientas avanzadas seguirá siendo crucial para abordar los complejos desafíos de la innovación tecnológica.

REFERENCIAS

- Alcántara, F. C. (2024). IA en la Educación: Desafíos de Implementación y Oportunidades de Transformación. *Revista Ilustração*, 5337-5358. doi:10.37811/cl_rcm.v8i2.10947
- Alvarez, J. (2023, Septiembre 18). *¿Es Usar Chat GPT una Trampa o una Herramienta Educativa? El Debate Desentrañado*. Retrieved from Alvarez Joseph Blog: <https://www.alvarezjoseph.com/blog/chat-gpt-etica-educativa/>
- Barraca-Mairal, J. (2024, Junio). Fronteras entre IA y humanos. *ResearchGate*. doi:https://www.researchgate.net/publication/381260695
- Basáez, E., & Mora, J. (2022). Salud e inteligencia artificial: ¿cómo hemos evolucionado? *Revista Médica Clínica Las Condes*, 33, 556-561. doi:10.1016/j.rmclc.2022.11.003
- Briva-Iglesias, V. (2023). Inteligencia artificial (IA). *Fundación Universitaria Oberta de Catalunya (FUOC)*. doi:10.13140/RG.2.2.25455.41127
- C. R. Domínguez Mayorga, M. A., Ramos Velasco, J. C., Ramos Fernández, E., & Hernández, E. (2012). Algoritmos Wavenet con Aplicaciones en la Aproximación de Señales: un Estudio Comparativo. *Revista Iberoamericana de Automática e Informática Industrial*, 347-358. doi:10.1016/j.riai.2012.09.001
- Galiana, L. I., Gudino, L. C., & González, P. M. (2024, Marzo). Ethics and artificial intelligence. *Revista Clínica Española, Volume 224, Issue 3*, 178-186. doi:https://doi.org/10.1016/j.rce.2024.01.007
- Gallardo, R. (2023, Abril 13). *¿Qué es Chat GPT? Académicos explican el funcionamiento de esta Inteligencia Artificial y sus aplicaciones*. Retrieved

from Universidad de Chile: <https://uchile.cl/noticias/204091/que-es-chat-gpt-academicos-explican-su-funcionamiento-y-aplicaciones>

- Gimeno-Ballester, V., & Trigo-Vicente, C. (2024). El rol de la inteligencia artificial en la publicación científica: perspectivas desde la farmacia hospitalaria. *Farmacia Hospitalaria*(100942). doi:10.1016/j.farma.2024.06.002
- Glück, R., & Yokoyama, T. (2023). Reversible computing from a programming language perspective. *Theoretical Computer Science*, 113-429. doi:10.1016/j.tcs.2022.06.010
- Gutiérrez, C., & López, M. (2022). La salud en la era digital. *Revista Médica Clínica Las Condes*, 562-567. doi:10.1016/j.rmclc.2022.11.001
- Haque, R. R., Brito, I. M., & Frade, S. (2024). Copilot e ChatGPT: Como utilizar ferramentas de IA na análise de dados. *Caderno de Laboratório, I*, 96-106. doi:10.15847/LAPSOCadLAb2024/ia
- Hernández, F., & Herrera, F. (2021). Identificación Inteligente de un Proceso Fermentativo Usando el Algoritmo GMDH Modificado. *Revista Iberoamericana de Automática e Informática Industrial*, 9 , 3-13. doi:10.1016/j.riai.2011.11.001
- J. Trujillano, J., March, M. B., & Rodríguez, A. S. (2003). Aplicación de las redes neuronales artificiales para la estratificación de riesgo de mortalidad hospitalaria. *Gaceta Sanitaria*, 17(6), 504-511. doi:10.1016/S0213-9111(03)71798-1
- J.F. Avila-Tomás, M., Mayer-Pujadas, V., & Quesada-Varela. (2020). La inteligencia artificial y sus aplicaciones en medicina I: introducción antecedentes a la IA y robótica. *Atención Primaria*, 52, 778-784. doi:10.1016/j.aprim.2020.04.013
- Jesús. (2024, Mayo 12). *Desventajas de Chat GPT*. Retrieved from Dongee Blog: <https://www.dongee.com/tutoriales/desventajas-de-chat-gpt/>
- Martinez, C. R. (2024, Marzo). Introdução ao conceito de Tecnologia Assistiva e modelos de abordagem da deficiência. *Universidad Católica de El Salvador*, 1-34. doi:10.13140/RG.2.2.12352.71681
- Mayer, M. A. (2023). Inteligencia artificial en atención primaria: un escenario de oportunidades y desafíos. *Atención Primaria*, 55, 1-3. doi:10.1016/j.aprim.2023.102744
- Medina Gamero, A. (2024). Formación en enfermería: tecnología avanzada con inteligencia artificial, realidad virtual y big data. *Educación Médica*, 25(100942). doi:10.1016/j.edumed.2024.100942

- Miguel Martínez-Comesaña, X. R.-D.-J., Martínez-Torres, J., Ocarranza-Prado, I., & Kreibel, D. (2023). Impact of artificial intelligence on assessment methods in primary and secondary education: Systematic literature review. *Revista de Psicodidáctica*, 93-103. doi:10.1016/j.psicoe.2023.06.002
- Pearson. (2023, Mayo 23). *Chat GPT: Su uso ético en la educación superior*. Retrieved from Pearson Latinoamérica Blog: <https://blog.pearsonlatam.com/educacion-del-futuro/chat-gpt-su-uso-etico-en-la-educacion-superior>
- Ponce, L. R. (2024). La Inteligencia Artificial en la formación de las competencias investigativas en estudiantes de educación superior. *Trabajo IA y NTIC competencias investigativas 2024 II*. doi:https://www.researchgate.net/publication/380733992
- Quirós-Fons, A., & García-Ull, F. J. (2023). La IA educativa: dilemas y perspectivas éticas. *Preprint*. doi:10.13140/RG.2.2.11073.53600
- Reyes, J. L. (2023, Junio 9). *¿Cuáles son sus alcances y limitaciones?* Retrieved from UNAM Global: https://unamglobal.unam.mx/global_revista/chat-gpt-cuales-son-sus-alcances-y-limitaciones/
- Rubín, C. N. (2024). La IA en la administración de negocios actual. *Cuadernos del CIMBAGE*, 61-76. doi:https://orcid.org/0009-0001-8074-1743
- Ruiz, R. B., & Velásquez, J. D. (2023). Inteligencia artificial al servicio de la salud del futuro. *Revista Clínica Las Condes*, 84-91. doi:10.1016/j.rmcl.2022.12.001
- Santos, R. N., Fernandes, A. B., Lima, A. B., Linhares, J. R., & Palha, V. S. (2024). Assistentes virtuais inteligentes na educação especial a distância: potencializando o suporte e a orientação acadêmica com IA. *Revista Ilustração*, 5, 37-51.
- Soage, S. (2022, Diciembre 28). *ChatGPT: proceso de formación, ventajas y limitaciones*. Retrieved from Aivo Blog: <https://es.aivo.co/blog/chatgpt-training-process-advantages-and-limitations>
- Mayo-Alvarez, L., Del-Aguila-Arcenales, S., Alvarez-Risco, A., Sekar, M. C., Davies, N. M., & Yáñez, J. A. (2024). Innovation by integration of Drum-Buffer-Rope (DBR) method with Scrum-Kanban and use of Monte Carlo simulation for maximizing throughput in agile project management. *Journal of Open Innovation: Technology, Market, and Complexity*, 10, 100-228.