

CORPORACION UNIVERSITARIA AUTÓNOMA DEL CAUCA



CHAT GPT 2.0

ENTREGABLE N°8 TALLER

CRISTIAN CAMILO MARTINEZ CORDOBA

ELECTIVA II

INGENIERÍA ELECTRONICA

MAYO 2025

En el presente informe, se observará un análisis detallado sobre el proceso de creación de un modelo de inteligencia artificial similar a GPT 2.0, basado en un video instructivo que guía a través de cada etapa del desarrollo utilizando HTML y JavaScript. Todo esto con el fin de proporcionar una comprensión clara de cómo se puede integrar la inteligencia artificial en aplicaciones web modernas, destacando su accesibilidad y versatilidad.

Primero que todo el video presenta un enfoque innovador para desarrollar un modelo de inteligencia artificial similar a ChatGPT que se ejecuta completamente en el navegador, utilizando JavaScript puro. Este enfoque elimina la necesidad de un backend o APIs externas, lo que permite que el modelo funcione de manera local y privada. La capacidad de ejecutar el modelo en el navegador no solo reduce costos, sino que también mejora la seguridad de los datos del usuario, ya que no se envían a servidores externos. Esto abre la puerta a una amplia gama de aplicaciones, desde herramientas de asistencia en línea hasta sistemas de gestión de contenido que pueden beneficiarse de la inteligencia artificial.

Uno de los conceptos clave discutidos es el uso de WebGPU, una API que permite acceder a la potencia de la GPU del dispositivo. Esto es fundamental para mejorar el rendimiento del modelo de inteligencia artificial, ya que permite realizar cálculos complejos de manera más eficiente. Al aprovechar la GPU, el modelo puede procesar datos más rápidamente, lo que resulta en respuestas más ágiles y una mejor experiencia de usuario. Este enfoque es especialmente relevante en aplicaciones que requieren un procesamiento intensivo, como el aprendizaje automático y la generación de texto.

El video también introduce el concepto de Web Workers, que permite la ejecución de scripts en un hilo separado del hilo principal de JavaScript. Esto es crucial para evitar que la interfaz de usuario se congele durante operaciones intensivas, como la carga y ejecución del modelo de IA. Al delegar tareas pesadas a un Web Worker, se mejora la interactividad de la aplicación, permitiendo que los usuarios sigan interactuando con la interfaz mientras se procesan los datos en segundo plano. Esta técnica es esencial para mantener una experiencia de usuario fluida y receptiva.

Además, se menciona la dependencia de Web LLM, que facilita la carga de modelos de inteligencia artificial preentrenados directamente en el navegador. Esta biblioteca permite a los desarrolladores integrar modelos de IA sin necesidad de configuraciones complicadas o dependencias adicionales. La posibilidad de utilizar modelos de código abierto y ejecutarlos localmente democratiza el acceso a la inteligencia artificial, permitiendo que más desarrolladores y empresas implementen soluciones innovadoras sin incurrir en altos costos.

De la misma manera se enfatiza la importancia de la programación multihilo en JavaScript a través de Web Workers. Tradicionalmente, JavaScript es un lenguaje de programación monohilo, lo que significa que todas las operaciones se ejecutan en un solo hilo. Esto puede llevar a problemas de rendimiento, especialmente en aplicaciones que requieren procesamiento intensivo, como los modelos de inteligencia artificial. Al utilizar Web Workers, los desarrolladores pueden delegar tareas pesadas a hilos separados, lo que permite que la interfaz de usuario permanezca receptiva.

Esta capacidad de manejar múltiples tareas simultáneamente es crucial para crear aplicaciones modernas que ofrezcan una experiencia de usuario fluida y sin interrupciones.

Otro concepto relevante es la carga de modelos de inteligencia artificial en el navegador. El autor del video explica cómo los modelos se descargan y almacenan en caché localmente, lo que significa que una vez que un modelo ha sido descargado, no es necesario volver a cargarlo en futuras interacciones. Esto no solo mejora la velocidad de respuesta, sino que también reduce la dependencia de la conectividad a Internet. Los usuarios pueden acceder a la inteligencia artificial de manera rápida y eficiente, incluso en entornos con conectividad limitada. Este enfoque es especialmente valioso para aplicaciones que requieren un acceso constante a la IA, como asistentes virtuales o herramientas de análisis de datos.

Además, se menciona la versatilidad de los casos de uso que surgen al implementar un modelo de IA localmente. Por ejemplo, los desarrolladores pueden integrar este modelo en sistemas de gestión de contenido para ayudar a los usuarios a generar texto, corregir errores o proporcionar sugerencias contextuales. También se pueden crear extensiones de navegador que utilicen la inteligencia artificial para mejorar la productividad del usuario. La posibilidad de personalizar y adaptar el modelo a diferentes contextos y necesidades específicas amplía enormemente su aplicabilidad en diversas industrias.

El enfoque de ejecutar inteligencia artificial en el navegador también plantea cuestiones interesantes sobre la privacidad y la seguridad de los datos. Al mantener el procesamiento de datos en el dispositivo del usuario, se minimizan los riesgos asociados con la transmisión de información sensible a servidores externos. Esto es especialmente relevante en un momento en que la privacidad de los datos es una preocupación creciente para los usuarios y las empresas. La capacidad de ofrecer soluciones de IA que respeten la privacidad del usuario puede ser un diferenciador clave en el mercado.

Finalmente, se destaca una reflexión sobre el futuro de la inteligencia artificial y su integración en aplicaciones web. A medida que las tecnologías continúan evolucionando, la posibilidad de ejecutar modelos de IA de manera local y eficiente en el navegador podría transformar la forma en que interactuamos con la tecnología. La combinación de accesibilidad, rendimiento y privacidad ofrece un camino prometedor para el desarrollo de aplicaciones más inteligentes y personalizadas, que se adapten a las necesidades individuales de los usuarios. Este enfoque no solo democratiza el acceso a la inteligencia artificial, sino que también fomenta la innovación en el desarrollo de software.

## **RESULTADOS DE LA IMPLEMENTACION DEL CODIGO.**

En la siguiente imagen se muestra la interfaz del GPT creado.

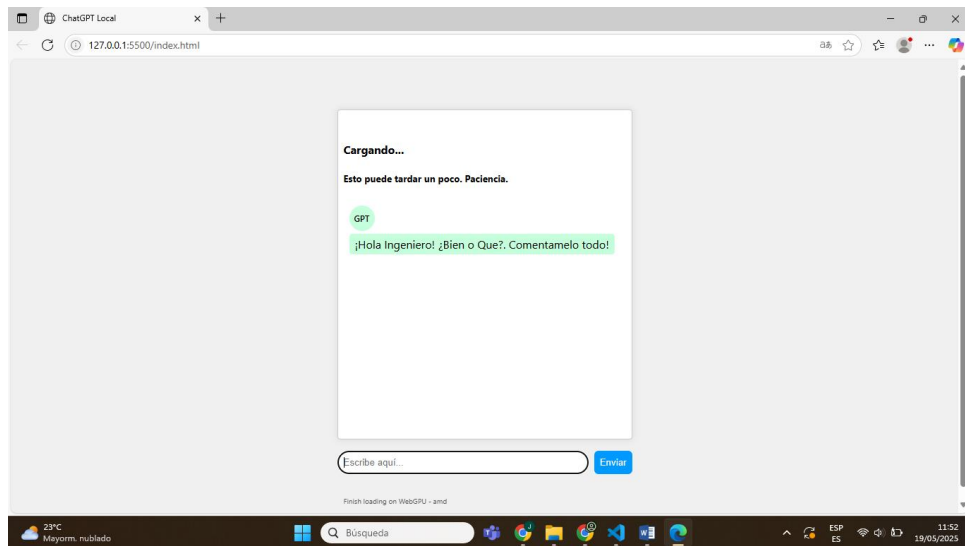


Fig 1. Interfaz GPT 2.0.

Diferentes preguntas hechas.

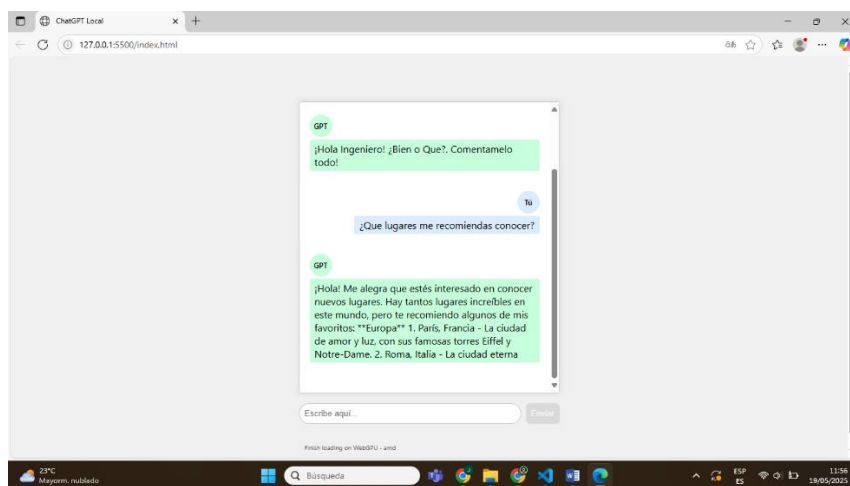


Fig 2. Prueba 1.

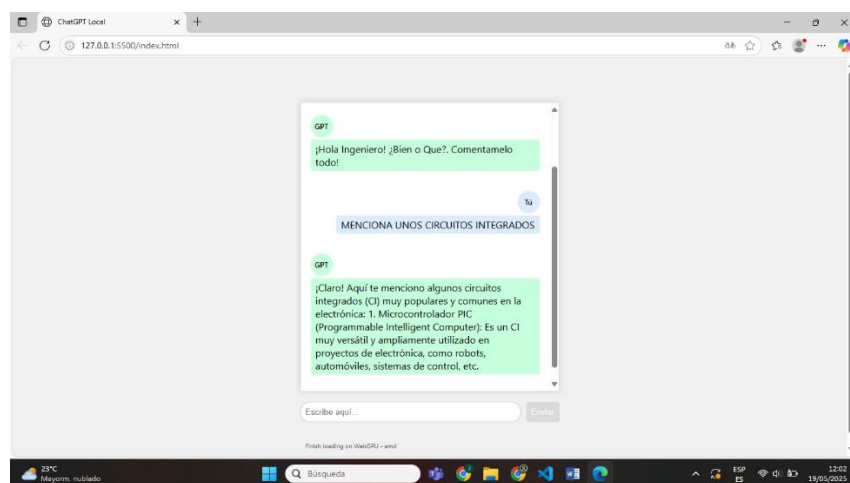


Fig 3. Prueba 2.

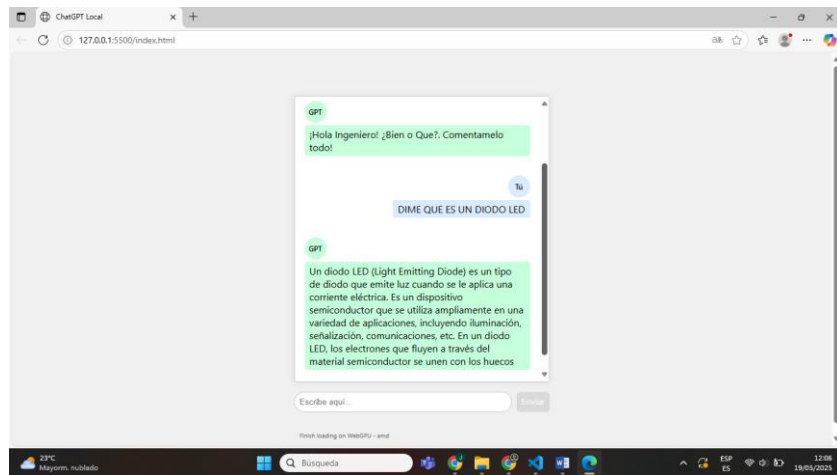


Fig 4. Prueba 3.

## REFERENCIAS.

<https://www.youtube.com/watch?v=HvoiF1MCPGs&t=84s>