



Tecnológico de Monterrey
Escuela de Ingeniería y Ciencias

Tecnológico de Monterrey

ESCUELA DE INGENIERÍA Y CIENCIAS

Inteligencia Artificial Avanzada para la Ciencia de Datos

Desarrollo e implementación de un asistente virtual para generación de promocionales.

03/12/2023

Profesores:

Dr. Daniel Otero Fadul

Equipo:

Pablo Gabriel Galeana Benítez - A01735281

Leonardo Ramírez Ramírez - A01351715

Miguel Chávez Silva - A01661109

Alessandro Arguelles Elias - A00829536

Luis Eduardo Aguilar Moreno - A01367545

Agustín Tapia - A01367639



Índice

1. Introducción	3
2. Problemática	3
3. Propuesta de solución	4
4. Desarrollo de la solución	4
4.1. Generación de texto	4
4.1.1. Obtención y definición del dataset	4
4.1.2. Elección del modelo base	5
4.1.3. Fine-tuning	6
4.2. Generación de imágenes	7
4.2.1. Obtención y definición del dataset	7
4.2.2. Elección del modelo base	8
4.2.3. Fine-tuning	9
4.3. Chatbot	11
4.3.1. Diseño del chatbot en Amazon Lex	11
4.3.2. Prompt engineering	12
5. Implementación de la solución	12
6. Resultados	14
7. Futuras recomendaciones	14
8. Conclusión	15
9. Anexos	16



1. Introducción

La incorporación de tecnologías innovadoras se ha vuelto importante para destacar en un mercado saturado y captar la atención de los consumidores. Una de las herramientas emergentes que ha revolucionado la manera en que las empresas abordan sus estrategias promocionales es la utilización de Inteligencia Artificial (IA) como asistentes virtuales. Los chatbots o asistentes de inteligencia artificial, como Bard, ChatGPT o Claude, emplean técnicas de procesamiento de lenguaje natural para proporcionar respuestas relevantes de manera casi instantánea, imitando una interacción humana.

Sin embargo, estas capacidades también nos plantean nuevos desafíos y responsabilidades. Es crucial abordar las implicaciones éticas, sociales y legales, reflexionando sobre nuestra función como usuarios. La inteligencia artificial generativa presenta dilemas éticos adicionales asociados con posibles sesgos en los resultados. Los modelos lingüísticos pueden replicar prejuicios y estereotipos perjudiciales presentes en los datos utilizados para su entrenamiento. Para abordar este problema, es común recurrir a la alineación de los sistemas, que implica ajustar y volver a entrenar los modelos para evitar la perpetuación de sesgos perjudiciales. Finalmente, el presente estudio detalla el proceso de desarrollo e implementación de un modelo destinado a la generación de texto e imágenes. Este análisis aborda los desafíos y limitaciones inherentes al proyecto, al tiempo que destaca los resultados obtenidos, los cuales se alinean con los principios éticos fundamentales de la inteligencia artificial y se basan en el uso de soluciones de código abierto[6].

Como se mencionó, la IA ha estado evolucionando y las empresas han tratado de evolucionar con ella. Hoy en día las empresas han estado invirtiendo mucho dinero para poder tener las últimas tecnologías dentro de sus oficinas y poder maximizar el impacto del negocio. La Inteligencia Artificial abarca una variedad de cosas, una de ellas es el marketing. El marketing es de suma importancia para las empresas y poder darse a conocer así como sus servicios, la IA puede jugar un rol muy importante dentro del marketing, desde creando pequeños slogans hasta pósteres completos de publicidad. Se han desarrollado chatbots con una interfaz amigable para que cualquier persona pueda hacer uso de ello. En general, el propósito de este reto es crear una interfaz de un chatbot que sea capaz de generar promociones así como imágenes de propaganda y puedan ser utilizados por Arcacontinental.

2. Problemática

Se busca desarrollar una solución integral destinada a la creación de promociones para los productos de Arca Continental. Esta solución se basa en la implementación de tecnologías de inteligencia artificial, incluyendo modelos de generación de imágenes, modelos de generación de texto y modelos de procesamiento de lenguaje natural. El objetivo es permitir a los usuarios finales interactuar de manera sencilla con la herramienta, asegurando que esta pueda proporcionar soluciones eficientes, prácticas y beneficiosas.



3. Propuesta de solución

Para abordar la problemática presentada, se propone una solución que se fundamenta en la implementación de un modelo de generación de imágenes y un modelo de generación de texto. Estos modelos interactuarán con el usuario a través de un chatbot, utilizando una interfaz de fácil acceso y comprensión.

En relación al modelo de generación de imágenes, se busca un enfoque basado en aprendizaje profundo que transforme el texto en representaciones visuales detalladas. Este modelo debe tener la capacidad de generar imágenes coherentes y específicas, ajustándose a las descripciones relacionadas con los productos del socio formador. Es esencial que el modelo sea adaptable a conjuntos de datos personalizados de los productos de Arca Continental, logrando un equilibrio óptimo entre la calidad de las imágenes generadas, el tiempo requerido para la generación y la coherencia visual.

En el ámbito de la generación de texto, se pretende implementar un modelo generador de texto capaz producir textos promocionales que guarden coherencia con los productos de nuestro socio formador, Arca Continental. Los modelos, basados en la transformación de texto a texto, deben asegurar eficacia tanto en la calidad del contenido generado como en el tiempo de generación. Además, se busca que el modelo sea ajustable, permitiendo su adaptación a las necesidades específicas de Arca Continental, garantizando así una respuesta precisa a las particularidades del contexto comercial.

4. Desarrollo de la solución

4.1. Generación de texto

Para la propuesta de generación de texto se implementó el modelo T5 (*Text-to-Text Transfer Transformer*) con arquitectura basada en la conversión de texto a texto para problemas de procesamiento de lenguaje natural. A continuación se describen los pasos seguidos para la implementación del modelo.

4.1.1. Obtención y definición del dataset

Como parte integral del desarrollo del modelo para su ajuste a la necesidad planteada, fue indispensable contar con una base de datos para el entrenamiento fine-tuning. El acercamiento seguido fue crear una base de datos propia con ejemplos de entradas y salidas de texto en formato de **prompt** y **target** siendo el *prompt* la posible petición de estrategia promocional del producto, y el *target* siendo la respuesta sugerida que el modelo tendría que generar. Dichos ejemplos de estrategias promocionales se obtuvieron conjuntamente de internet y de una base de datos compartida por el socio formador.

Para lograr el mejor rendimiento del modelo fue indispensable contar con ejemplos variados y generales para la mayor cantidad de productos de Arca Continental dentro de la base de datos. Para



dicha tarea, se utilizó ChatGPT como herramienta de ayuda para la optimización en la generación de una gran cantidad de ejemplos.

En la tabla 1 se muestran algunos ejemplos de la base de datos creada. Es crucial mencionar que la base de datos se creó sin tomar en consideración las redes de negocio por lo que el contenido de las propuestas promocionales pueden tener un bias en la asignación del precio.

Prompt	Target
Generate a marketing offer:	Buy 4 cokes and get the last one for free!
Create a special deal for Coca Cola:	Get a 6-pack of Coca Cola for the price of 5!
Come up with a promotion for Coca Cola products:	Buy any two Coca Cola drinks and get 20 % off the total.
Design an offer for Coke lovers:	Purchase 3 cokes and receive a complimentary Coca Cola glass.
Generate a discount for Coca Cola beverages:	Enjoy a special price of 50 pesos for a pack of 8 Coca Cola cans.

Tabla 1: Algunos ejemplos de la base de datos construida

Una vez se termino de construir la base de datos, se subió a la plataforma de HuggingFace para que después ésta fuera utilizada en el entrenamiento fine-tuning del modelo base seleccionado. Al final se obtuvieron 1004 ejemplos. Se puede visitar la base de datos en el siguiente link

Tras tener una base de datos con ejemplos de peticiones y propuestas lo más detalladas, generalizables, y con numerosos ejemplos se asegura tener un mejor rendimiento para que el modelo T5 pueda aprender las relaciones entre el vocabulario y sintaxis utilizado en las peticiones y lo que se espera que genere en las propuestas.

4.1.2. Elección del modelo base

El modelo T5 particular seleccionado fue *FLAN T5 Small*, el cual a diferencia de otros modelos T5, es mejor en varias tareas de procesamiento de lenguaje natural (NLP) manteniendo el mismo número de parámetros (77 millones). Estos modelos presentan una gran versatilidad ya que fueron creados para aprovechar la tranferencia de aprendizaje (learning transfer) lo que hace más eficiente al modelo pues permite un buen desempeño utilizando lo aprendido en otras tareas de NLP [5]. Además, la ventaja competitiva que presentó el modelo FLAN T5 ante otros modelos de generación de texto como GPT-2 y BERT fue su enfoque unificado para las tareas de NLP. Esto hace que los modelos T5 puedan abordar una amplia variedad de tareas de NLP bajo el formato de entrada-salida de texto sin necesidad de ajustar finamente los parámetros previamente aprendidos [5].



De manera general, los modelos T5, tras contar con una arquitectura de red neuronal transformer, alcanzan excelentes métricas de rendimiento logrando resultados significativos para una variedad de tareas como: resúmenes de texto, traducción automática, pregunta y respuesta, generación de texto contextualizado, entre otras [5]. El funcionamiento de la red transformer se basa en la implementación de mecanismos de atención para el procesamiento de secuencias de entrada y generar secuencias de salida. Su estructura permite una atención global a las secuencias de entrada, permitiendo así una comprensión contextual completa mejorando la captura de relaciones a larga distancia [9].

4.1.3. Fine-tuning

Finalmente, se procedió a realizar el fine-tuning. Esta técnica consiste en tomar un modelo pre-entrenado, generalmente en un conjunto de datos grande y genérico, y adaptarlo o afinarlo para una tarea o dominio específico con un conjunto de datos más pequeño y particular [4].

Para el entrenamiento fine-tuning fue necesario preprocesar la base de datos y tokenizar su contenido. El procesamiento de la base de datos se realizó para que ésta mantuviera una estructura en particular y el modelo fuera capaz de utilizarla para el fine-tuning. Para el procesamiento se utilizó la librería de *datasets* de HuggingFace la cuál creó un objeto de tipo *DatasetDict* dándole la estructura en forma de diccionarios anidados, además de dividir en subconjuntos de entrenamiento, validación y prueba. Después, se utilizó el tokenizador *T5Tokenizer* de HuggingFace para tokenizar y preparar las secuencias de texto para que fueran utilizadas por el modelo T5. La tabla 2 muestra los parámetros utilizados para dicho proceso (*max_input_length* varía según el tamaño de la secuencia más larga en el conjunto de datos). La tokenización genera ID's de las secuencias de datos tokenizadas así como sus respectivas máscaras de atención y labels; todos siendo tensores de varias dimensiones.

Parámetro	Valor
<i>max_input_length</i>	80
<i>max_target_length</i>	130

Tabla 2: Parámetros para el proceso de tokenización de las secuencias de texto de toda la base de datos

Una vez habiendo tokenizado la base de datos se realizó el fine-tuning bajo el enfoque sequence-to-sequence utilizando la librería de HuggingFace *Seq2SeqTrainer*. La tabla 3 muestra los hiperparámetros utilizados en la función *Seq2SeqTrainer* (cabe mencionar que los valores de los hiperparámetros se seleccionaron según el tamaño del conjunto de datos).

La tabla 4 muestra los resultados obtenidos tras el proceso de fine-tuning. En este caso, sólo se entrenó al modelo por una época por lo que únicamente se muestran los resultados para una sola. Según se puede observar las métricas de rendimiento del modelo como las dos medidas Rouge, RougeL y RougeM son relativamente bajas considerando el tamaño y complejidad de la base de datos.



Hiperparámetro Nombre del Argumento	Valor Valor
model_name	Flan T5 Small
evaluation_strategy	epoch
learning_rate	1e-5
per_device_train_batch_size	6
per_device_eval_batch_size	6
weight_decay	0.01
save_total_limit	3
num_train_epochs	2
predict_with_generate	True
fp16	True
push_to_hub	True

Tabla 3: Hiperparámetros para el proceso de fine-tuning utilizando la función Se2SeqTrainer

Train Loss	Epoch	Step	Val Loss	Rouge1	Rouge2	RougeL	RougeLsum	Gen Len
No log	1.0	101	nan	35.8419	14.1387	29.6846	29.7698	17.0846

Tabla 4: Resultados del Entrenamiento

4.2. Generación de imágenes

Como se menciono anteriormente, una parte muy importante de la propuesta es la generación de imágenes. Para esta, se tuvieron que desarrollar algunos pasos previos debido a las necesidades que impone un modelo ajustado a necesidades específicas. Todos y cada uno de ellos son como se describe adelante.

4.2.1. Obtención y definición del dataset

En primer lugar, debido a que no se contaba con un dataset para el fine-tuning, este tuvo que ser creado utilizando imágenes que se encuentran en la red. Para ello, se utilizó el sitio web Coca-Cola en tu hogar. Básicamente se obtuvo el sitemap del sitio, el cual es un archivo de tipo xml (posteriormente convertido a json) que contiene toda la arquitectura del sitio y por lo tanto los links a cada uno de las diferentes imágenes junto con sus títulos, posteriormente se utilizó el archivo para descargar cada una de las imágenes correspondientes a productos y asignarles como título la descripción que se encuentra en el sitio. Simultáneamente, mientras se guardaban las imágenes se creaba un archivo tipo csv que contuviera el link y la descripción de los productos.

Una vez fueron descargadas todas las imágenes, utilizando el csv mencionado se procedió a describir de manera más eficiente cada uno de los productos en inglés (debido a que la mayoría de los modelos fueron entrenados originalmente en ese idioma y su encoder funciona mejor así), utilizando

una descripción base como:

Sin embargo, no todas las imágenes contenían solo un producto o incluso, tenían un fondo negro, el cual no es beneficioso para el entrenamiento de este tipo de modelos. Debido a lo anterior, tuvieron que se depuradas las imágenes que contenían esas características, como se muestra en la figura 4.



(a) Imágenes de varios productos. (b) Imagen de producto con fondo oscuro.

Figura 1: Ejemplos de imágenes depuradas del dataset original.

Para el final de la depuración, de las originales 618 imágenes, solo quedaron 451. Una vez tenido el dataset descrito y depurado, utilizando la librería *datasets* de HuggingFace se transformo en un objeto que pudiera servir como entrada a un modelo de generación de imágenes y posteriormente se empujo hacia un centro de datasets (en el sitio web de HuggingFace), debido a que así sería mucho más fácil su almacenamiento y su acceso para el entrenamiento. Se puede visitar el dataset en el siguiente link. Una vista previa se puede observar en la figura 2.

4.2.2. Elección del modelo base

Una vez que se contó con el dataset de entrenamiento, se selecciono el modelo base para su entrenamiento. En realidad no hubo una búsqueda tan intensiva alrededor de los modelos, debido a que la mayoría de los que se encuentran disponibles y de fácil acceso para un entrenamiento fine-tuning están en HuggingFace y se basan en el mismo algoritmo que es *stable diffusion*.

En términos generales, este algoritmo se basa en la propagación gradual de información o cambios a lo largo de una red o estructura. En lugar de realizar cambios abruptos, el algoritmo de

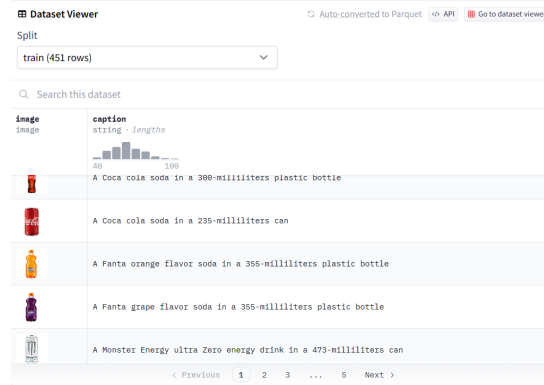


Figura 2: Vista previa del dataset en HuggingFace.

stable-diffusion aplica transformaciones suaves o gradualmente ponderadas para evitar fluctuaciones bruscas o efectos no deseados en el sistema. Originalmente desarrollado por Runway y LMU Múnich para generar imágenes digitales de alta calidad a partir de descripciones en lenguaje natural o estímulos. [7]

Los factores más relevantes para seleccionar el modelo bajo las anteriores premisas entonces se cerraron a que; tuviera una licencia de uso que permitiera trabajar con el y generar imágenes para comercializar, que su arquitectura no fuese tan pesada en términos de hardware y que tuviera un buen desempeño.

Fue entonces que se selecciono el modelo Stable-Diffusion v2 de StabilityAi, el cual es una versión más avanzada del modelo original de Runway y con una arquitectura mucho más ligera, lo que permite un fine-tuning con menos recursos, sumado que cuenta con un mejor desempeño en imágenes de calidad 768x768 píxeles en comparación con su antecesor. [8]

4.2.3. Fine-tuning

Finalmente, con el modelo seleccionado y el dataset listo, se procedió a realizar el fine-tuning donde básicamente se buscó entrenar el modelo base para que fuera capaz de reconocer los productos necesarios no solo por nombre, si no por tamaño y tipo de envase.

El fine-tuning se realizó con la ayuda de la biblioteca *Diffusers* de Hugging Face que provee una documentación bastante comprensible acerca del entrenamiento del modelo al extraerlo de su centro de modelos. Los hiperparametros durante el este fueron los que se muestran en la tabla 5.

El fine-tuning duro 9:30hrs y se requirieron 200Gb de almacenamiento para guardar los checkpoints, así como una computadora virtual con 40Gb de memoria gráfica. Su rendimiento se muestra en la figura 3.

Argumento	Valor
-mixed_precision	fp16
-pretrained_model_name_or_path	stabilityai/stable-diffusion-2-1
-dataset_name	GabrielGaleana34/Products
-use_ema	True
-resolution	1024
-center_crop	True
-random_flip	True
-train_batch_size	1
-gradient_accumulation_steps	4
-gradient_checkpointing	True
-max_train_steps	10000
-learning_rate	1e-06
-max_grad_norm	1
-resume_from_checkpoint	latest
-caption_column	caption
-validation_prompts	A Coca-cola soda in a plastic bottle
-lr_scheduler	constant
-lr_warmup_steps	0
-push_to_hub	True

Tabla 5: Hiperparametros para fine-tuning.

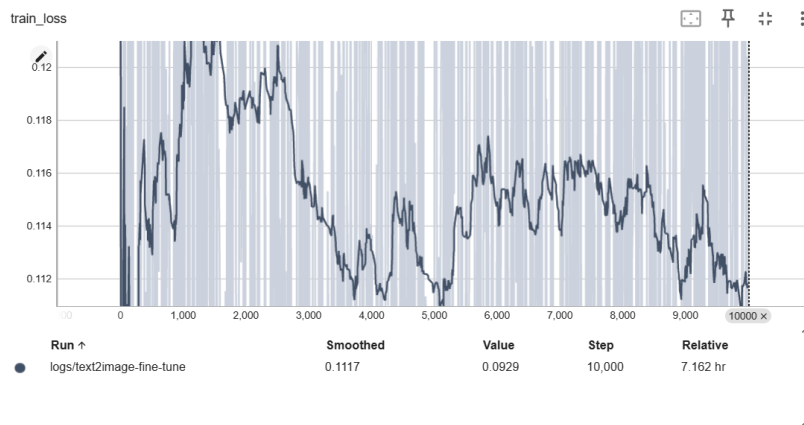


Figura 3: Desempeño del modelo durante el fine-tuning.

Una vez que termino el proceso fue empujado hacia HuggingFace y su centro de modelos y puede visitarse en el siguiente link



4.3. Chatbot

La ultima parte de la propuesta de solución es un chatbot y representa el eje central de la misma, ya que es la interfaz con el usuario y la conexión entre distintas partes de la solución. La idea principal detrás del chatbot es crear una interfaz cómoda, segura y confiable para que cualquier usuario pueda crear sus promociones sin importar su conocimiento con las tecnologías más avanzadas en inteligencia artificial generativa. Para ello, se definieron algunas preguntas clave que buscan recabar datos vitales basados en el dataset provisto por la sociaformadora, como el producto objetivo, el entorno en donde se vende o la clase de negocio en que se hace. Una vez se recaban los datos se conecta con otros componentes en la implementación para realizar la inferencia de los modelos antes mencionados y así, devolver al cliente su petición.

4.3.1. Diseño del chatbot en Amazon Lex

Para el diseño del chatbot se utilizaron los servicios de AWS debido a su fácil implementación y compatibilidad con lenguajes de programación como lo es Python a través de su biblioteca Boto3. Se utilizó en especifico, Amazon Lex con la consola V1, ya que es bastante conocida y fácil de manejar. Solo se diseño un único Intent, es decir, solo una clase de consulta al chatbot, ya que no se pretendía generalizar para otra clase de usos. Este Intent, constaba de varios Slots (par de respuestas entre usuario y chatbot), con el fin de recabar la información antes mencionada. Para ser más específico, la información recabada por el chatbot era:

- Tipo de promoción (Imagen y texto o alguno de los dos).
- Producto.
- Tamaño.
- Tipo de envase.
- Precio.
- Tipo de tienda
- Tipo de entorno.

Una vez termina de recabar la información, se procede a verificar si el usuario esta seguro de la información provista, en caso de decir que sí, se generará la promoción, en caso de que no, se tendrá que reiniciar la conexión con el chatbot. Una vista de ejemplo del chat se muestra en la figura ??.



Figura 4: Ejemplo de conversación con chatbot.

4.3.2. Prompt engineering

En lo que respecta al prompt engineering del chatbot, éste recaba los atributos previamente mencionados como tipo de promoción, producto, tamaño, etc. Una vez teniendo todos los atributos, el modelo será capaz de generar el texto promocional y la imagen correspondiente según los atributos especificados por el usuario. Sin embargo, es importante mencionar que para la generación de texto, la estructura del prompt deberá contener **Generate a promotion:** como parte inicial del prompt seguido de las especificaciones concretas del producto y tipo de promoción que se quiere generar. Esta llave especial especifica qué tipo de tarea de NLP el usuario quiere generar con el objetivo de que T5 sea capaz de entregar secuencias de texto adecuadas a los requerimientos del usuario.

Una particularidad de la estructura general del prompt para ambos modelos es que debe ser en idioma inglés pues tanto T5 y Stable Diffusion se adecúan de mayor manera a este idioma.

5. Implementación de la solución

Para la implementación de la solución se requirió a los servicios de AWS. AWS son múltiples servicios que se ofrecen en la nube ofreciendo soluciones escalables para la computación, el almacenamiento, las bases de datos, el análisis, entre otras cosas. El uso de AWS tiene costo dependiendo de los servicios a usar y la cantidad de los mismos.

El chatbot es la parte esencial dentro de la implementación, ya que es la conexión entre lo que desea el usuario y todos los modelos. Al entender esto, primero se diseñó una interfaz amigable al usuario para que pueda sin problema hacer uso de la herramienta.

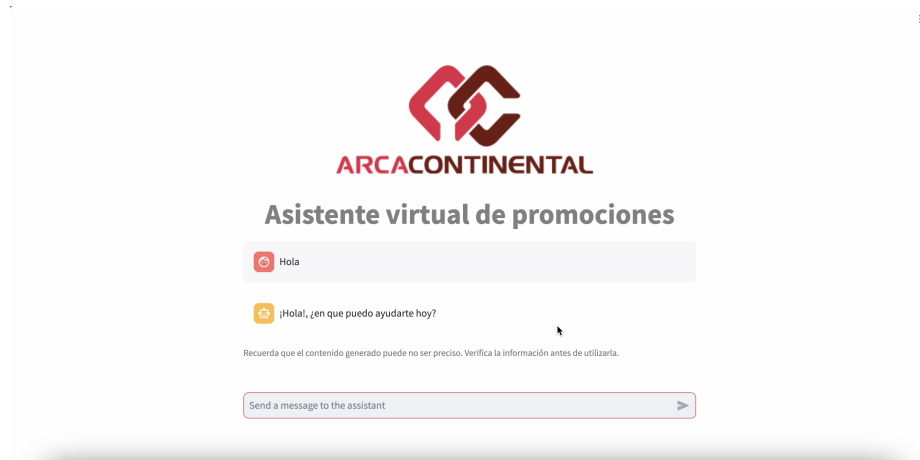


Figura 5: Interfaz de Chatbot

En la figura 5 se puede observar el de Arcacontinental, con un subtítulo que indica que ese chatbot sirve como asistente virtual de promociones. Hasta abajo hay una barra para poder escribir y poder hablar con el chatbot y finalmente, arriba de la barra se muestra la historia del texto que se ha ido desarrollando. Como se puede observar, la interfaz es amigable para cualquier persona que tenga deseos de usarla.

La interfaz del usuario se creó con Streamlit. Streamlit es un framework de acceso libre, desarrolladas para Python que ayudan a poder desarrollar interfaces web sin tener que hacer uso de HTML o CSS. Está diseñado para facilitar la creación de aplicaciones de machine learning, visualización de datos y paneles de control de manera rápida y sencilla [3]. Además se usó Boto 3, que es un kit de desarrollo de software para Python creada por AWS para escribir código que permita estar conectado a la nube de AWS, permitiendo una integración eficiente y efectiva con la amplia gama de servicios que ofrece AWS, facilitando la automatización y la gestión de infraestructuras en la nube. Se usó Boto3 para poder conectar la interfaz a Amazon Inferentia.

Para el almacenamiento de los modelos, se usó Amazon Inferentia, un producto de AWS. Amazon Inferentia es un chip de procesamiento para poder optimizar los modelos de Machine Learning, es decir, es el hardware donde se almacenan los modelos. Se almacenaron los modelos, tanto de imagen como de texto, en Amazon Inferentia, con esto se pudo lograr un alto y constante rendimiento. Además se complementa con Amazon Neuron que sirve para optimizar y acelerar las cargas de trabajo de inferencia de machine learning. Amazon Neuron usa transformers y diffusers para poder ejecutar estas tareas. Un transformer es un tipo de arquitectura de modelo, se usa debido a su

eficacia en tareas de procesamiento de lenguaje natural[2]. Un diffuser es una herramienta que ayuda a generar resultados muy parecidos a los ejemplos con los que fue entrenado el modelo, de esta forma se evita obtener un resultado muy diferente al esperado[1].

6. Resultados

Finalmente, después de la implementación de la solución, se pudo lograr obtener un chatbot. Se hicieron pruebas en donde se le preguntó al chatbot que generara una imagen en un ambiente escolar con una Coca-Cola de 2 litros de vidrio.



Figura 6: Póster de propaganda

En la imagen 6 se puede observar una Coca-Cola de 2L centrada en la imagen donde se puede apreciar que es el corazón de la foto en un comedor escolar y al fondo a la derecha se puede apreciar un póster igual de Coca-Cola.

Si se desea visualizar un vídeo en donde se tiene la interacción completa con el chatbot y además se obtiene al final la imagen que generó el modelo, siga el siguiente link

7. Futuras recomendaciones

Los diferentes modelos que se usaron fueron de acceso libre, por lo que tienen algún tipo de debilidad dependiendo de lo que se quiera, para el modelo de imágenes, es muy sensible al poder escribir cosas, sin embargo, en las imágenes se veía legible el logo de Coca-Cola. Se intentó utilizar el modelo para otros tipo de productos y obteníamos resultados pero en lo particular se grabó más el producto de Coca-Cola, es por eso que para el modelo generativo de imágenes se recomienda obtener



una base de datos más amplia, mejor definida y completa para el tipo de promociones requeridas, de esta forma se podrían obtener mejores resultados ante otro tipo de producto.

Por parte del modelo de texto, en particular sí se tenían resultados, solo que en inglés, si es que se quiere en español, el modelo utilizado podría no ser el más adecuado, por otra parte, tiene noción de los precios en dólares, así que de igual forma se podría enfocar en enseñar la moneda del lugar de origen. El dataset utilizado no era lo suficientemente extenso como para poder desarrollar una oferta muy atractiva es por eso que se recomienda un dataste mucho más extenso con diferentes promociones para que pueda aprender mejor qué tipo de promociones son las que se buscan. Por otra parte, la rentabilidad de la oferta podría no ser la mejor si es que el modelo no hace algún tipo de predicción acerca de los ingresos contra los egresos de dicha oferta. Utilizar servicios existentes para potencializar y generalizar la efectividad del chatbot, como Amazon Transcribe, Amazon Translate y Amazon Comprehend, es importante comentar que dichos servicios cuestan, pero que sin embargo ayudan de manera muy eficaz con el objetivo.

Considerar utilizar otros modelos de generación de imágenes en cuanto estos estén disponibles (DALLE3, GPT4) como base para la creación de modelos personalizados, ya que son modelos mucho más actualizados y pueden ayudar con la mejora de la calidad de imagen y de texto, mejor comprensión del contexto para brindar un mejor producto, reducción de respuestas erróneas, reducción de sesgos, entre otros. Al ser un proyecto con recurso monetarios limitados, no se pudieron aprovechar herramientas que facilitarían y potenciarían los resultados finales, es por eso que se recomienda lograr destinar ciertos recursos hacia el proyecto.

8. Conclusión

Se puede concluir que se logró resolver el problema planteado, se aprendieron a usar diferentes herramientas que pudieron potenciar las soluciones propuestas. El chatbot desarrollado permitió una mayor automatización en la creación de contenido publicitario, reduciendo el tiempo y los recursos necesarios para desarrollar campañas efectivas, resultando así en una eficiencia operativa significativamente mejorada. Los diferentes modelos que se usaron sí pudieron ser entrenados para lograr que se adaptaran a la problemática en cuestión. Los desafíos que fueron apareciendo a lo largo del proyecto se pudieron resolver lo que potenció las habilidades del equipo. La IA es una herramienta que debe ser aprovechada al máximo en los próximos años, aprender algunas ramas de la IA es de mucha ayuda en la vida diaria ya que las compañías están tratando de implementarla. La experiencia adquirida y los resultados obtenidos nos impulsan a continuar explorando nuevas tecnologías y enfoques innovadores.



Referencias

- [1] (2022). Diffusion models clearly explained.
- [2] (2022). ¿qué es un modelo transformer?
- [3] (2023). Crear una interfaz en streamlit — keepcoding bootcamps.
- [4] Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.
- [5] Chung, H. W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., Li, Y., Wang, X., Dehghani, M., Brahma, S., et al. (2022). Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.
- [6] Franganillo, J. (2023). Los grandes modelos de lenguaje: una oportunidad para la profesión bibliotecaria. *Anuario ThinkEPI*, 17.
- [7] Patil, S., Cuenca, P., Lambert, N., and von Platen, P. (2022). Stable diffusion with diffusers. *Hugging Face Blog*. [<https://huggingface.co/blog/rlhf>](https://huggingface.co/blog/stable_diffusion).
- [8] Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. (2021). High-resolution image synthesis with latent diffusion models.
- [9] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

9. Anexos

Se puede encontrar una liga al repositorio con los códigos usados en el siguiente link.

Los créditos de las fotografías pertenecen a sus respectivos autores. ©