

Generación de Prompts

Idea Alquímica

Desarrollar y presentar una propuesta para el Proyecto Final, en el cual seleccionarás una problemática de tu interés y desarrollarás una

propuesta de solución vinculada a la generación de prompts.

Tabla de contenido

Presentación del Problema a Abordar2

Desarrollo de la Propuesta de Solución 3

Viabilidad Económica del Proyecto4

Metodología6

Herramientas7

Descripción de Prompts.....8

Problemas con Pagos 9

Devoluciones y Reembolsos..... 10

Justificación de la Viabilidad del Proyecto 11

IA y Mano Humana: Colaboración para un Trabajo Más Placentero y Eficaz..... 15

Casos Prompt 16

Presentación del Problema a Abordar

Problema Elegido: La ineficiencia en la atención al cliente en empresas de comercio electrónico.

Una atención al cliente deficiente puede llevar a la pérdida de clientes y una mala reputación de la empresa. Los clientes esperan respuestas rápidas y precisas a sus consultas, lo cual puede ser difícil de lograr sin la ayuda de tecnologías avanzadas. Mejorar la atención al cliente puede aumentar la satisfacción del cliente y, en última instancia, las ventas.

Logo de la empresa



Nombre de la empresa: AI Innovations

Desarrollo de la Propuesta de Solución

Solución Propuesta: Utilizar prompts específicos para generar respuestas automáticas y eficaces a las consultas más comunes de los clientes utilizando IA.

Se utilizarán modelos de lenguaje IA, como los ofrecidos por OpenAI, para generar respuestas personalizadas y precisas a las consultas de los clientes. Esto incluye tanto modelos texto-texto (para responder preguntas frecuentes) como texto-imagen (para proporcionar visuales de productos o soluciones gráficas).



Viabilidad Económica del Proyecto

Para garantizar la viabilidad económica de la implementación de la IA en el servicio de atención al cliente, realicé una estimación de costos basada en la documentación oficial de OpenAI. A continuación, se detallan los recursos necesarios y el costo estimado para cada componente del sistema.

Recursos Requeridos y Costos Estimados:

- **Prompt System (Entrada del Sistema):**
 - **Tokens:** 249 tokens
 - **Costo:** USD 0.0001245 por cada solicitud al sistema
- **Prompt User (Entrada del Usuario):**
 - **Tokens:** 106 tokens
 - **Costo:** USD 0.000053 por cada consulta realizada por un usuario
- **Salida del Asistente (Respuesta de la IA):**
 - **Tokens:** 127 tokens
 - **Costo:** USD 0.0001905 por cada respuesta generada por el asistente
- **Generación de Imágenes:**
 - **Resolución:** 256x256 píxeles
 - **Costo:** USD 0.015 por cada imagen generada

Precios de Generación de Texto:

- **Costo por Entrada:**
 - **Tarifa: USD 0.0005 por cada 1K tokens**
- **Costo por Salida:**
 - **Tarifa: USD 0.0015 por cada 1K tokens generados**

Estrategias para Controlar los Costos:

1. Optimización del Prompting:

- **Emplear tácticas de prompting que optimicen la relevancia de las respuestas, evitando el uso excesivo de tokens y limitando el desperdicio de recursos.**

2. Definición de max_tokens:

- **Configurar un límite máximo de tokens (max_tokens) según la longitud requerida de las respuestas, para asegurar un control sobre los costos asociados a la generación de texto.**

3. Uso de Resolución Baja en Imágenes:

- **Utilizar imágenes de baja resolución (256x256 píxeles) para minimizar los costos de generación de imágenes, asegurando que sean suficientemente claras para su propósito sin incurrir en gastos innecesarios.**

Resumen de Costos:

El modelo propuesto permite un control preciso sobre los costos asociados a la implementación de IA en el servicio de atención al cliente.

Mediante la aplicación de estrategias de optimización y la elección de resoluciones adecuadas para las imágenes, se garantiza que la solución sea no solo efectiva, sino también económicamente viable

Metodología

1. Recopilación de Datos:

- **Descripción:** Recopilar una pieza de código que se desea analizar.
- **Objetivo:** Obtener una muestra representativa del código que será evaluado por la IA para generar un análisis preciso.

2. Procesamiento de Texto:

- **Herramienta:** Utilizar GPT-3.5 de OpenAI.
- **Descripción:**
 - **Análisis del Código:** Ingresar el código recopilado en el sistema para que el modelo de lenguaje lo procese.
 - **Salida Esperada:** Generar una breve explicación del propósito del código, evaluar su sustentabilidad y proporcionar consejos para reducir su huella de carbono.

3. Generación de Imagen:

- **Herramienta:** Utilizar CHAT GPT y COPILOT para la generación de imágenes.
- **Descripción:**
 - **Convertir la evaluación generada por GPT-3.5 en una imagen visual que represente los resultados obtenidos.**
 - **Objetivo:** Proveer un recurso visual que ayude a entender y comunicar la evaluación del código de manera más efectiva.

Herramientas

- **Lenguaje de Programación:**
 - **Python:** Utilizado para integrar las API de OpenAI y Chat GPT/Copilot y para orquestar el flujo de trabajo del proyecto.
- **APIs Utilizadas:**
 - **OpenAI GPT-3.5 API:** Empleada para el procesamiento de texto, generando explicaciones y evaluaciones del código ingresado.
 - **GPT/Copilot:** Utilizada para la creación de imágenes basadas en la evaluación de la sustentabilidad del código.
- **Plataforma de Desarrollo:**
 - **ChatGPT 3.5:** Utilizado como base para interactuar con los modelos de lenguaje y facilitar la generación de prompts y respuestas.

Descripción de Prompts:

Consulta sobre estado de pedido:

Prompt: ¿Cuál es el estado de mi pedido con número [número de pedido]?

Respuesta esperada: Tu pedido con número [número de pedido] está actualmente en [estado del pedido] y se espera que llegue el [fecha estimada de entrega].

Ejemplo de Prueba

Prompt 1: Consulta sobre el estado de un pedido


Entrada del Cliente:

 Copiar código

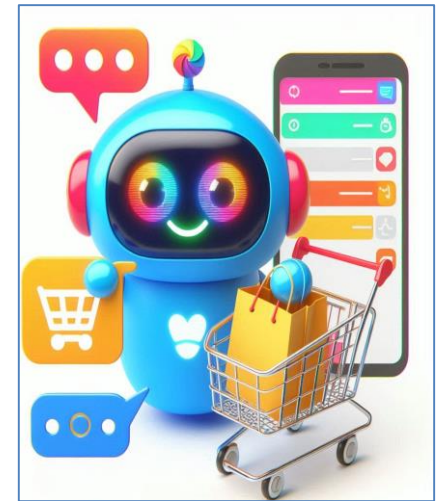
¿Cuál es el estado de mi pedido con número 12345?

Respuesta Generada por la IA:

less

 Copiar código

Tu pedido con número 12345 está actualmente en tránsito y se espera que llegue el 25 de julio de 2024. Puedes seguir el estado de tu pedido en tiempo real utilizando el siguiente enlace: [\[enlace de seguimiento\]](#).



Problemas con Pagos:

Prompt: Estoy teniendo problemas para procesar mi pago. ¿Qué puedo hacer?

Respuesta esperada: Por favor, verifica que los detalles de tu tarjeta sean correctos y que tengas suficiente saldo. Si el problema persiste, intenta usar otro método de pago o contacta a nuestro soporte al cliente en [correo electrónico o número de teléfono].

Prompt 2: Problemas con el pago

Entrada del Cliente:

Copiar código

Estoy teniendo problemas para procesar mi pago. ¿Qué puedo hacer?

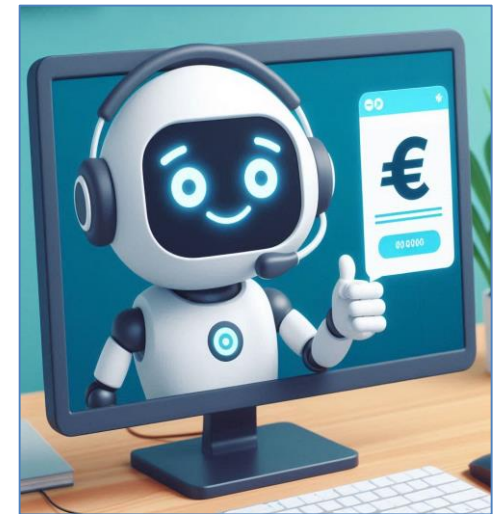
Respuesta Generada por la IA:

CSS

Lamentamos que estés teniendo problemas para procesar tu pago. Por favor, verifica que los detalles de tu tarjeta sean correctos y que tengas suficiente saldo. Si el problema persiste,

Copiar código

intenta usar otro método de pago o contacta a nuestro soporte al cliente en soporte@ejemplo.com o llama al 0800-123-456.



Devoluciones y Reembolsos:

Prompt: ¿Cómo puedo devolver un producto y solicitar un reembolso?

Respuesta esperada: Para devolver un producto, sigue estos pasos: [instrucciones de devolución]. Una vez recibido el producto devuelto, procesaremos tu reembolso en [número de días] días laborales.

Ejemplo de Texto a Imagen

Entrada del Cliente:

CSS

Copiar código

¿Pueden mostrarme una imagen del producto XYZ?

Respuesta Generada por la IA:

less

Copiar código

Aquí tienes una imagen del producto XYZ:
![Imagen del Producto XYZ](enlace_a_la_imagen_del_producto_XYZ.jpg)



Justificación de la Viabilidad del Proyecto

Viabilidad Técnica: Disponemos de acceso a la API de OpenAI y un equipo de desarrolladores con experiencia en integración de IA. Además, las herramientas necesarias para probar y ajustar los prompts están fácilmente disponibles.

Justificación de Elecciones: Hemos elegido utilizar la IA para mejorar la atención al cliente debido a su capacidad para manejar grandes volúmenes de consultas de manera eficiente y efectiva. Esta tecnología permite proporcionar respuestas rápidas y precisas, mejorando la experiencia del cliente y liberando tiempo para que los agentes humanos puedan manejar consultas más complejas.

Respuestas Inmediatas: Las respuestas automáticas brindan soluciones rápidas a las consultas de los clientes, sin demoras.

Reducción de Tareas Repetitivas: La IA automatiza respuestas estándar, liberando a los agentes humanos de tareas monótonas.



Optimización del Tiempo: Los Chatbots pueden atender mensajes las 24 horas del día, mejorando la experiencia del usuario.

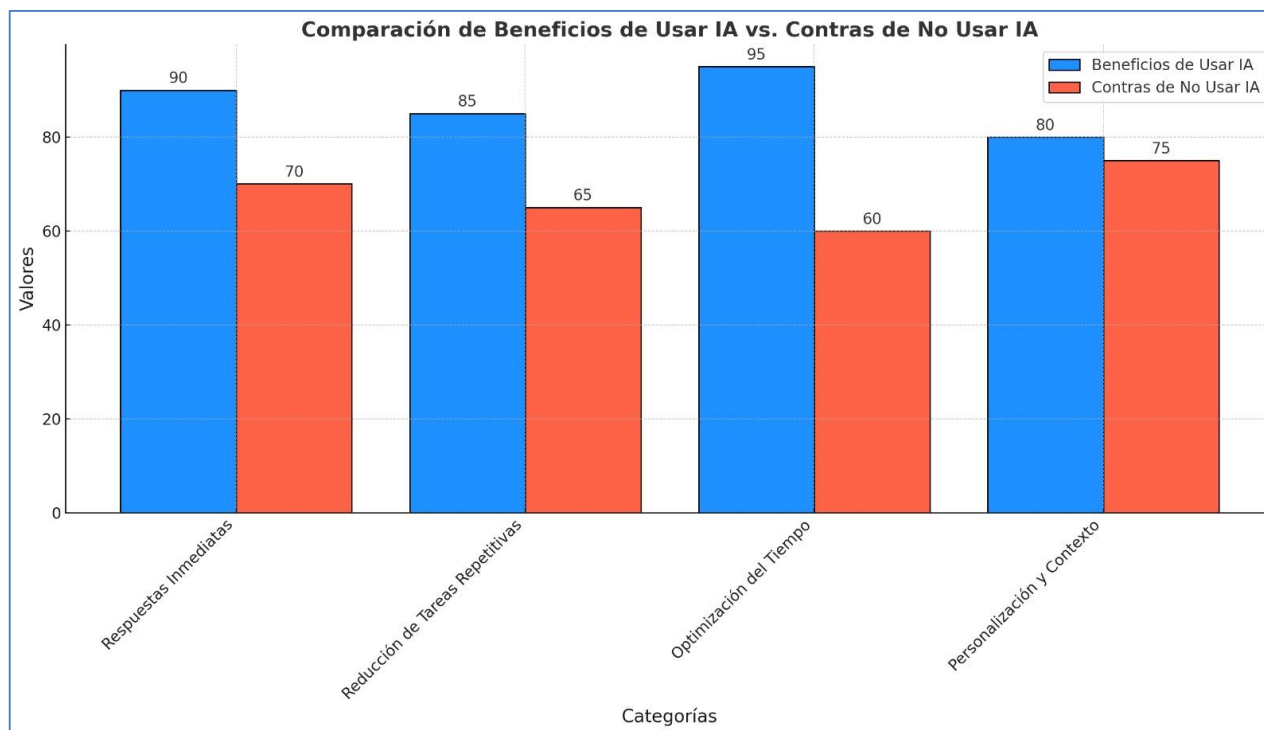


Personalización y Contexto: Las respuestas automáticas pueden ser personalizadas y adaptadas al contexto de cada consulta.

La IA al Alcance de Todos: Uso mundial sin demoras. A puede ser utilizada en todo el mundo por múltiples personas al mismo tiempo sin demoras ni problemas.



GENERACIÓN DE PROMPTS



IA y Mano Humana: Colaboración para un Trabajo Más Placentero y Eficaz

La inteligencia artificial, al trabajar en conjunto con la mano humana, puede hacer que el trabajo sea más placentero y llevadero, pero es crucial entender que ambas dependen mutuamente para lograr estos beneficios.



Casos Prompt

Caso 1: TEXTO - TEXTO

```
import openai  
  
openai.api_key = ""
```

```
# Contexto y prompt
```

```
context = """
```

```
Situación: Un cliente pregunta sobre el estado de su pedido usando su número de identificación.
```

```
Propósito: Informar al cliente sobre el estado actual de su pedido y la fecha estimada de entrega para mejorar su experiencia y gestionar expectativas.
```

```
Información del Pedido:
```

- Número de Pedido: 378
- Estado: Enviado
- Fecha Estimada de Entrega: 5 de agosto de 2024

```
"""
```

```
prompt = "¿Cuál es el estado de mi pedido con número 378?"
```

```
conversation = [  
    {"role": "system", "content": context},  
    {"role": "user", "content": prompt}  
]  
  
response = openai.ChatCompletion.create( model='gpt-3.5-turbo',  
messages=conversation, max_tokens=500,  
temperature=0.7  
)  
  
# Corrected way to access the response content
```

```
message = response['choices'][0]['message']['content']  
  
print("Assistant: {}".format(message))
```

Caso 2: TEXTO - TEXTO y TEXTO - IMAGEN

```
import openai

openai.api_key = ""

context = """ Situación: Un usuario está teniendo problemas para procesar su pago en el sitio web. Propósito: Asistir al usuario en resolver el problema con su pago, ofreciendo alternativas y soporte adicional si es necesario. """

prompt = "Estoy teniendo problemas para procesar mi pago. ¿Qué puedo hacer?"

conversation = [

    {"role": "system", "content": context},

    {"role": "user", "content": prompt}

]

response = openai.ChatCompletion.create( model='gpt-3.5-turbo',

messages=conversation, max_tokens=500,

temperature=0.7

)

# Extraer y mostrar la respuesta

message = response['choices'][0]['message']['content']

print("Assistant: {}".format(message))

# Pedir al usuario que ingrese el nombre del artículo

articulo = input("Por favor, ingresa el nombre del artículo con el que estás teniendo problemas: ")
```

```
# Mostrar un mensaje con el nombre del artículo

print(f"Has indicado que tienes problemas con el artículo: {articulo}")

# Crear el prompt para la imagen basada únicamente en el artículo ingresado

prompt_imagen = f"Una imagen de un {articulo}."

# Generar la imagen con DALL·E

response_imagen = openai.Image.create(

    prompt=prompt_imagen,

    n=1,

    size="1024x1024"

)

# Obtener la URL de la imagen generada

image_url = response_imagen['data'][0]['url']

# Mostrar la imagen generada

display(Image(url=image_url))
```


Se ingresa un artículo: Impresora



Caso 3: TEXTO - TEXTO y TEXTO - IMAGEN

```
import openai

openai.api_key = ""

context = """ Situación: Un usuario desea devolver un producto y solicitar un reembolso. Propósito: Guiar al usuario a través del proceso de devolución y reembolso para asegurar una experiencia sin problemas."""

prompt = "¿Cómo puedo devolver un producto y solicitar un reembolso?"

conversation = [

    {"role": "system", "content": context},

    {"role": "user", "content": prompt}

]

response = openai.ChatCompletion.create(
    model='gpt-3.5-turbo',

    messages=conversation,
    max_tokens=500,

    temperature=0.7

)
```

```
# Extraer y mostrar la respuesta

message = response['choices'][0]['message']['content']

print("Assistant: {}".format(message))

# Pedir al usuario que ingrese el número de comprobante

numero_comprobante = input("Por favor, ingresa el número de comprobante: ")

# Pedir al usuario que ingrese el nombre del artículo

articulo = input("Por favor, ingresa el nombre del artículo que deseas devolver: ")

# Mostrar el número de comprobante y el artículo ingresado

print(f"El número de comprobante es: {numero_comprobante}")

print(f"El nombre del artículo es: {articulo}")

# Preguntar si desea devolver el dinero

desea_reembolso = input(f"¿Deseas que se te devuelva el dinero por el artículo '{articulo}'? (si/no): ").strip().lower()

if desea_reembolso == 'si':

    # Simulación del monto a devolver (puedes ajustar según tu lógica)

    monto_a_devolver = 100.00 # Ejemplo: $100.00

    print(f"El monto a devolver es: ${monto_a_devolver:.2f}")
```

else:

Crear el prompt para generar imágenes de artículos alternativos

prompt_imagenes = f"Genera imágenes de artículos alternativos para cambiar el artículo '{articulo}'."

Generar las imágenes con DALL·E

response_imagenes = openai.Image.create(
 prompt=prompt_imagenes,
 n=3, # Generar 3 imágenes de artículos alternativos
 size="1024x1024"
)

Mostrar las imágenes generadas

for i in range(len(response_imagenes['data'])):

image_url = response_imagenes['data'][i]['url']