

«Talento Tech»

Inteligencia Artificial

Clase 05





Clase N° 5 | Prompt Engineering y Herramientas de IA

Temario:

- Prompt Engineering
- Prompts - estructura
- Herramientas de IA para la vida cotidiana.



Prompt engineering

Detrás del funcionamiento de cualquier inteligencia artificial, ya sea en sus respuestas textuales o en la generación de imágenes, hay un extenso proceso de desarrollo y prueba de sistemas. También intervienen personas que formulan solicitudes para obtener **respuestas adecuadas**. Este rol clave es desempeñado por los ingenieros de prompts, quienes optimizan la interacción entre los usuarios y la inteligencia artificial para **garantizar resultados efectivos y precisos**. Esa es la labor de los **prompt engineers**.

El prompt engineering es una disciplina enfocada en potenciar el uso de modelos de lenguaje basados en inteligencia artificial. Esto mediante el diseño y prueba de frases o entradas que hagan que un sistema arroje los resultados que necesita un usuario.

¿Qué es prompt engineering?



Es una disciplina enfocada en potenciar el uso de modelos de lenguaje basados en IA.



Mediante el diseño y prueba de frases o entradas.

HubSpot



Su objetivo es encontrar la ruta correcta para obtener resultados adecuados de estas tecnologías.

El trabajo de un prompt engineer no exige conocimientos de programación, desarrollo informático ni escritura de código. Estos profesionales sólo deben poseer las habilidades de comunicación pertinentes para asegurar que el sistema de IA pueda darnos lo que estamos buscando con precisión.

Tareas que realiza un prompt engineer:

- **Hace pruebas de interacción:** A través de la prueba de diferentes oraciones y comandos, el prompt engineer evalúa el éxito que tiene el sistema para responder a preguntas concretas y específicas encontrando defectos para luego solucionarlos.
- **Entrena al sistema:** Los modelos de lenguaje necesitan someterse a un entrenamiento. Este proceso ocurre cuando, de manera iterativa, se corre el programa con algunos valores de entrada para guiar al modelo y que sus respuestas sean pertinentes y ajustadas a un tema concreto. De este modo, una herramienta puede destinarse a públicos específicos y generar contenidos que sean valiosos para ellos.
- **Determina el tono de las respuestas:** Establece un tono de interacción lúdico y divertido, serio y profesional o neutro y servicial dirigiéndolo a un tipo de usuario concreto
- **Delimita los alcances de los modelos:** Deben asegurarse de que la herramienta, basada en modelos de lenguaje, cumpla con su función (y nada más). Por ejemplo, si dispones de un chatbot para tu tienda online de ropa, el sistema debe responder preguntas sobre el servicio, ayudar al usuario a comprar o administrar la información de su cuenta. No te servirá de nada contar con un chatbot que establezca conversaciones casuales con el visitante o que dé respuestas sobre salud u otros temas.

Si te interesa saber más te dejamos un recurso extra gratuito:

<https://saturdays.ai/curso-ia-generativa/>

Estructura de Prompts

Cuando nos comunicamos con chatbots o modelos de lenguaje avanzados como ChatGPT, la forma en que formulamos preguntas o comandos juega un papel crucial en la calidad de las respuestas que recibimos. Este arte de hacer preguntas específicas se conoce como "Prompt Engineering". Imagina que estás afinando una herramienta poderosa, y cada palabra que eliges es como ajustar un instrumento fino para obtener la melodía perfecta. En esta breve introducción, exploraremos la importancia de conocer la estructura y cómo preguntar adecuadamente para obtener resultados precisos y útiles de estas potentes inteligencias artificiales. ¡Comencemos a descubrir el mundo del Prompt Engineering!

Ahora conozcamos un poco sobre lo que es un Prompt. En el contexto de la informática y la inteligencia artificial, un "Prompt" se refiere a una solicitud o instrucción que se envía a un sistema para obtener una respuesta específica. En el ámbito de los modelos de lenguaje, como GPT, generalmente toma la forma de una oración o frase que se utiliza para iniciar la generación de texto.

Por ejemplo, si estás utilizando un chatbot y le escribes "¿Cuál es la capital de Francia?", esa pregunta se considera el Prompt, y el chatbot generará una respuesta basada en la información que tiene disponible.

Estructura de un Prompt.

La estructura de un Prompt puede variar según el contexto y el tipo de modelo de lenguaje o sistema con el que estés interactuando. Sin embargo, en términos generales, puede incluir los siguientes elementos:

- **Inicio claro:** El Prompt comienza con una instrucción o pregunta clara. Es esencial que la solicitud esté formulada de manera precisa para obtener la respuesta deseada.
- **Detalles específicos:** Incluir detalles específicos en el Prompt puede ayudar a guiar la respuesta del modelo. Cuanta más información relevante proporcionas, más probable es que obtengas una respuesta precisa.
- **Contexto adecuado:** Si es relevante, puedes incluir información contextual para asegurarte de que el modelo comprenda el contexto de tu pregunta. Esto puede ayudar a mejorar la calidad de la respuesta.
- **Formato consistente:** Mantener un formato consistente en tus interacciones puede ser útil. Por ejemplo, si estás realizando una serie de preguntas, mantener un formato similar puede facilitar la interpretación para el modelo.
- **Lenguaje claro y natural:** Aunque los modelos de lenguaje pueden comprender instrucciones más técnicas, es a menudo beneficioso utilizar un lenguaje claro y natural en tus prompts para mejorar la comprensión y generar respuestas más coherentes.



Ejemplos

Prompt sobre tipos de Datos y Variables:

"Hola, ¿puedes explicarme qué son los tipos de datos en Python y cómo se declaran las variables?"

Estoy interesado en entender la diferencia entre tipos de datos simples y compuestos, ¿podrías proporcionar ejemplos de ambos?"

"En el contexto de la programación, ¿cómo afecta la elección del tipo de dato al rendimiento del programa?. Por favor, explícame de manera sencilla cómo Python gestiona la asignación de memoria para variables. ¡Gracias por tu ayuda! Esto me será útil al programar en Python"

Saludo y Apertura:

- "Hola, ¿puedes explicarme qué son los tipos de datos en Python y cómo se declaran las variables?"

Detalles Relevantes:

- "Estoy interesado en entender la diferencia entre tipos de datos simples y compuestos, ¿podrías proporcionar ejemplos de ambos?"

Contexto (si es necesario):

- "En el contexto de la programación, ¿cómo afecta la elección del tipo de dato al rendimiento del programa?"

Instrucción Adicional (si es necesario):

- "Por favor, explícame de manera sencilla cómo Python gestiona la asignación de memoria para variables."

Cierre:

- "¡Gracias por tu ayuda! Esto me será útil al programar en Python."

Prompt para resolver problemas de error que nos de la consola de Visual Studio Code.

"Hola chat, estoy trabajando en un programa Python para calcular el área de un triángulo. Tengo un bloque de código que consta de 10 líneas, pero al ejecutarlo, me da un error. ¿Podrías ayudarme a identificar y corregir el problema? Aquí está el código:

```
#Codigo con Error
def calcular_area_triangulo(base, altura):
    area = base * altura / 2
    return area

base = 5
resultado = calcular_area_triangulo(base)
print(f"El área del triángulo es: {resultado}")
```

El error que obtengo por consola es el siguiente:

TypeError: calcular_area_triangulo() missing 1 required positional argument: 'altura'

Apreciaría tu ayuda para solucionar este problema."

Saludo y Apertura:

- "Hola chat, estoy trabajando en un programa Python para calcular el área de un triángulo."

Detalles Relevantes:



- ""Tengo un bloque de código que consta de 10 líneas, pero al ejecutarlo, me da un error."

Contexto (si es necesario):

- "Aquí está el código con el error que estoy enfrentando:"

```
#Codigo con Error
def calcular_area_triangulo(base, altura):
    area = base * altura / 2
    return area

base = 5
resultado = calcular_area_triangulo(base)
print(f"El área del triangulo es: {resultado}")
```

Instrucción Adicional (si es necesario):

- "El error que obtengo es el siguiente: TypeError: calcular_area_triangulo() missing 1 required positional argument: 'altura'. Apreciaría tu ayuda para solucionar este problema."

Cierre:

- "¡Espero tu orientación para corregir este error! Gracias de antemano por tu ayuda."

Herramientas de IA para la vida cotidiana

Existen numerosas herramientas de inteligencia artificial (IA) que pueden facilitar y mejorar la vida cotidiana. Desde la integración aún más profunda de la IA en la vida diaria, como la atención médica hasta la creación de ciudades inteligentes. Las posibilidades son infinitas. Te presentamos algunas categorías y ejemplos de herramientas que quizás no conozcas:

Diferentes Herramientas de IA.

- [Microsoft Designer](#): Creador de imágenes a partir de un Prompt con Dall-E.
- [Microsoft Copilot](#): Chatbot
- [Chat Pdf](#) : Resumir archivos en PDF e interacción con un chatbot para el aprendizaje.
- [Grok](#): Puede ser utilizada para crear asistentes virtuales altamente eficientes, generar contenido en tiempo real, y mejorar la interacción con los usuarios en plataformas de medios digitales.
- [Journeai](#): Planificador de Viajes.
- [AI Recipe](#): Recetas de cocinas con tan solo colocar ingredientes.
- [Rix](#): ChatBot "exclusivo" para los desarrolladores.
- [Phin](#): ChatBot que te facilita la creación de código y añade explicaciones.
- [Invideo](#): Crea videos a partir de Prompts.
- [ResumelA](#): Crea un CV en cuestión de segundos con IA.
- [Gemini](#): El chatbot de Google.
- [Gamma](#): Creación de presentaciones, documentos y páginas web.
- [PDFIA](#): Página para resumir pdf
- [Notion](#): Usa IA para organizar tareas y colaborar en proyectos.
- [MyFitnessPal](#): Usa IA para rastrear hábitos alimenticios y ofrecer consejos para una mejor salud.
- [Headspace](#): Ofrece meditaciones y técnicas de bienestar personalizadas según tus necesidades.
- [Duolingo](#): Utiliza IA para enseñarte idiomas de manera interactiva y adaptativa.

Categorías

Dentro de todas las herramientas de inteligencia artificial que podemos encontrar, ninguna sirve para todo lo que queramos. Dependiendo la función que queremos cumplir, tenemos que buscar la herramienta de inteligencia artificial que tenga el motor que nosotros necesitamos. Si queremos resumir un pdf, podemos hacerlo con chat GPT, pero sabemos que hay herramientas que tienen un algoritmo que trabaja especialmente para resumir un PDF.

Por eso las herramientas de IA se suelen dividir principalmente entre las siguientes categorías:



1. Asistentes virtuales.
2. Aplicaciones de Productividad.
3. Herramientas de comunicación.
4. Bienestar y salud.
5. Educación y aprendizaje.
6. Robótica y automatización

¿Cómo ordenamos las herramientas que vimos anteriormente en esta lista?

¿Conoces más herramientas que quieras compartir?

Objetivos a Futuro

- **Ética y Seguridad:** La automatización y la IA plantean preguntas sobre la ética del trabajo, la privacidad y la responsabilidad en caso de errores.
- **Integración:** La implementación efectiva de robots e IA en sistemas existentes puede ser compleja y costosa.
- **Desarrollo Continuo:** A medida que la tecnología avanza, se espera que la robótica y la automatización con IA continúen evolucionando, impactando significativamente en la economía y la sociedad.



Desafío N° 5:

1. Pensá 4 herramientas de IA que crees que necesitas para tu día a día.
2. Desarrolla un prompt sobre las herramientas que pensaste anteriormente y usalo con culturizar chatbot para obtener información.
3. Desarrolla una presentación en [Gamma](#) con lo que buscaste.
4. Compartí el link de la presentación en el campus así como con tu instructor y tus compañeros!.



Buenos Aires
aprende
Agencia de Actividades para el Futuro

BA Buenos
Aires
Ciudad