

# Curso de Desarrollo de Chatbots con OpenAI

# ¿Cómo usar la API de OpenAI?

Con su documentación



**Reto:** usa la  
**documentación** para  
crear una nueva  
**API Key**

# Chat Completions

```
openai.chat.completion.create(  
  model="gpt-3.5-turbo",  
  messages=[
```

```
)  
]
```

```
openai.chat.completion.create(  
  model="gpt-3.5-turbo",  
  messages=[  
    {"role": "system", "content":  
    "Eres un asistente."},  
    {"role": "user", "content":  
    "¿Quién ganó el mundial de fútbol?"},  
    {"role": "assistant",  
    "content": "El mundial de 2022 lo ganó  
    Argentina"},  
    {"role": "user", "content":  
    "¿Dónde se jugó?"}  
  ]  
)
```

```
        {"role": "system", "content":  
"Eres un asistente."},  
        {"role": "user", "content":  
"¿Quién ganó el mundial de fútbol?"},  
        {"role": "assistant",  
"content": "El mundial de 2022 lo ganó  
Argentina"},  
        {"role": "user", "content":  
"¿Dónde se jugó?"}  
    ]  
)
```

# ¿Qué has aprendido de la API?

1. Navegar  
documentación
2. Cargar modelo
3. Chat Completions



# Parámetros de Completions

```
response= openai.chat.completions.create(  
    model='gpt-3.5-turbo',  
    messages= [  
        {"role": "system", "content" : "Eres un  
asistente que da información sobre  
historia"},  
        {"role": "user", "content" : "¿Quién  
descubrió América?"}  
    ],  
    max_tokens= 50,  
  
)
```

```
response= openai.chat.completions.create(  
    model='gpt-3.5-turbo',  
    messages= [  
        {"role": "system", "content" : "Eres un  
asistente que da información sobre  
historia"},  
        {"role": "user", "content" : "¿Quién  
descubrió América?" }  
    ],  
    max_tokens= 50,  
    temperature= 0,  
    top_p= 1,  
    n= 1  
)
```

*Parámetros  
opcionales  
más usados*

```
response =  
openai.Completion.create(  
    engine="text-davinci-003",  
    prompt="¿Qué es IA?",  
    max_tokens=150,  
    temperature= 0,  
    top_p = 1,  
    n = 1  
)
```

Parámetros  
opcionales  
más usados

# Temperature

Mi curso favorito de Platzi es

---

Curso de API  
de OpenAI

0.6

Curso de  
Fundamentos de ETL  
con Python y Pentaho

0.3

Curso de Python:  
PIP y Entornos  
Virtuales

0.1

**Temperature  
= 0**

Mi curso favorito es Curso de API  
de OpenAI

Mi curso favorito es Curso de API  
de OpenAI

Mi curso favorito es Curso de API  
de OpenAI

# Temperature

Mi curso favorito de Platzi es

---

Curso de API  
de OpenAI

0.6

Curso de  
Fundamentos de ETL  
con Python y Pentaho

0.3

Curso de Python:  
PIP y Entornos  
Virtuales

0.1

**Temperature  
= 0.3**

Mi curso favorito es Curso de API  
de OpenAI

Mi curso favorito es Curso de API  
de OpenAI

Mi curso favorito es Curso de  
Fundamentos de ETL con Python  
y Pentaho

# Temperature

Mi curso favorito de Platzi es

---

Curso de API  
de OpenAI

0.6

Curso de  
Fundamentos de ETL  
con Python y Pentaho

0.3

Curso de Python:  
PIP y Entornos  
Virtuales

0.1

**Temperature  
= 0.7**

Mi curso favorito es Curso de  
Python: PIP y Entornos Virtuales

Mi curso favorito es Curso de API  
de OpenAI

Mi curso favorito es Curso de  
Fundamentos de ETL con Python  
y Pentaho



**Reto: modificar  
parámetros y  
observar cambios  
en resultados.**



**Manos a  
la obra con  
la API de  
OpenAI**



**Reto:** encuentra  
**errores** que tiene el  
modelo y mejora  
el **prompt**

# Buenas prácticas al usar modelos de OpenAI

# Uso

- Especifica claramente tu solicitud.
- Utiliza la instrucción inicial.
- Controla la longitud de la respuesta.
- Experimenta con la temperatura.

# Billing

- Consumo de usuarios y facturación.
- Soft limit y hard limit.
- Quota increase request.

# Seguridad

- Gestión y solución de problemas.
- Ética y consideraciones legales.
- Privacidad de los datos.
- Control de users.

# Fine-tuning de modelos de OpenAI

# ¿Cuándo NO hacer fine-tuning?

- Prompt engineering.
- Encadenamiento de prompts.
- Function calling.



**Iterar con  
prompt  
engineering**

¿Cuándo  
SÍ hacer  
fine-tuning?

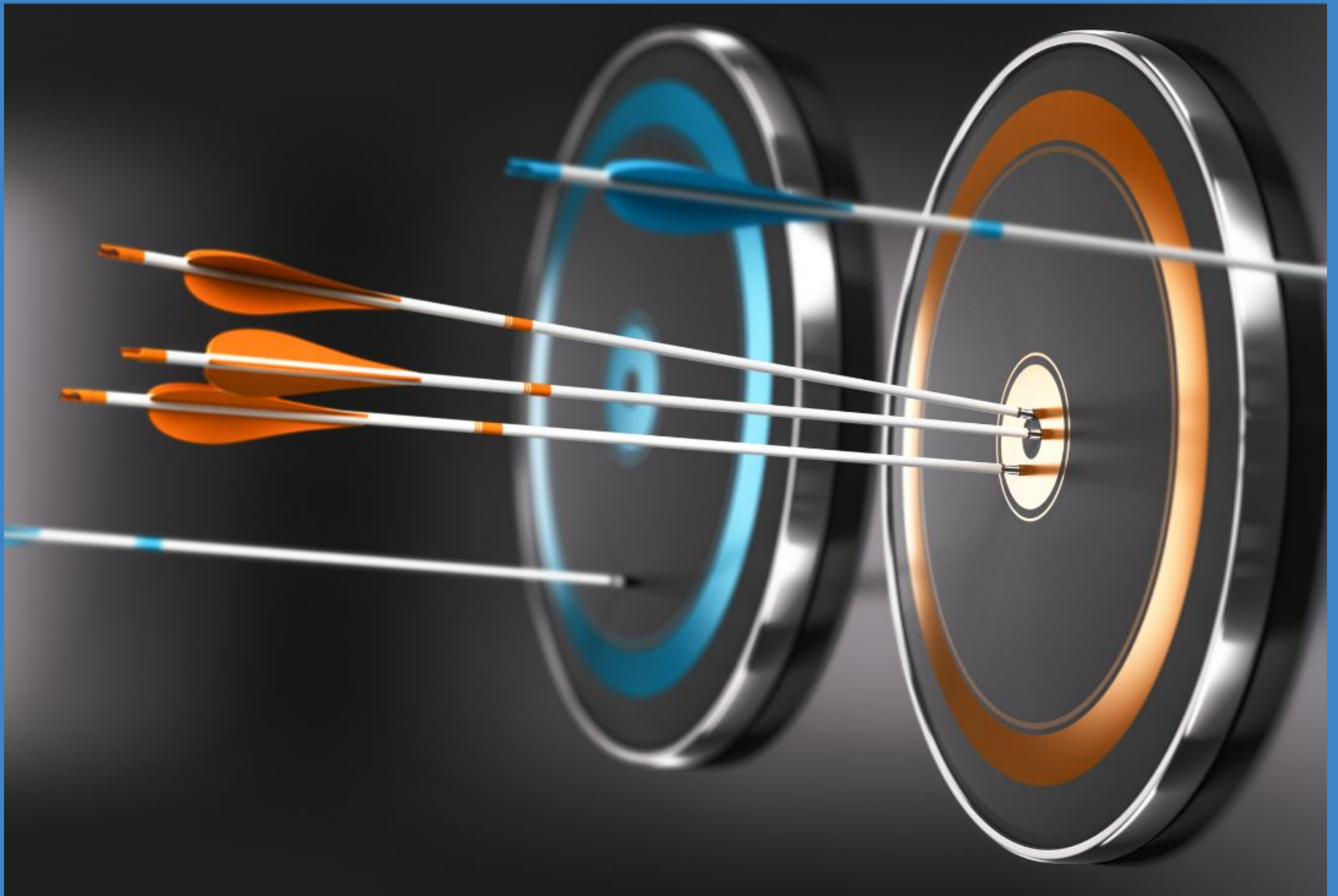


# Estilos, tono, formato

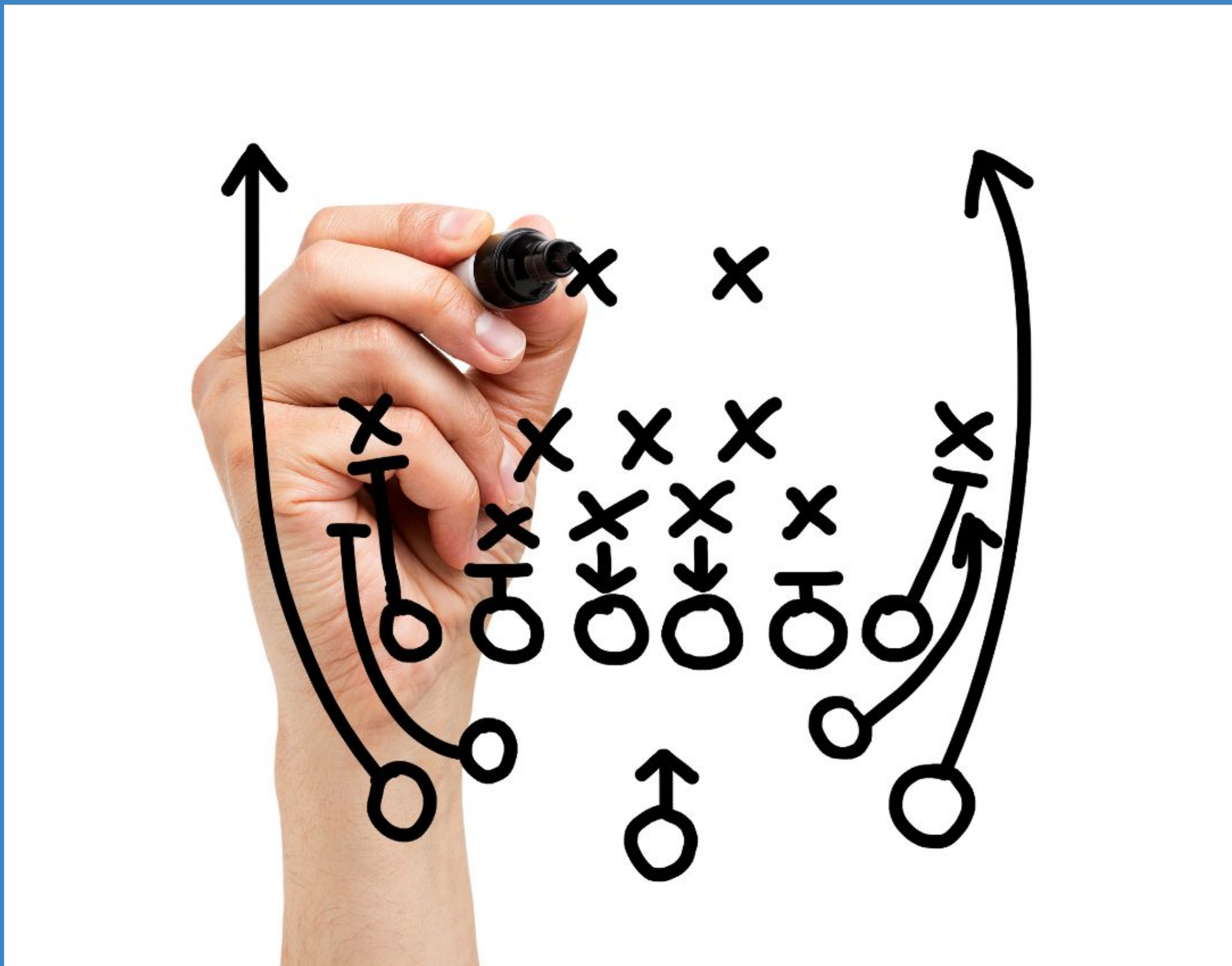




# Confiabilidad para salida deseada



# Corregir fallas al seguir instrucciones complejas



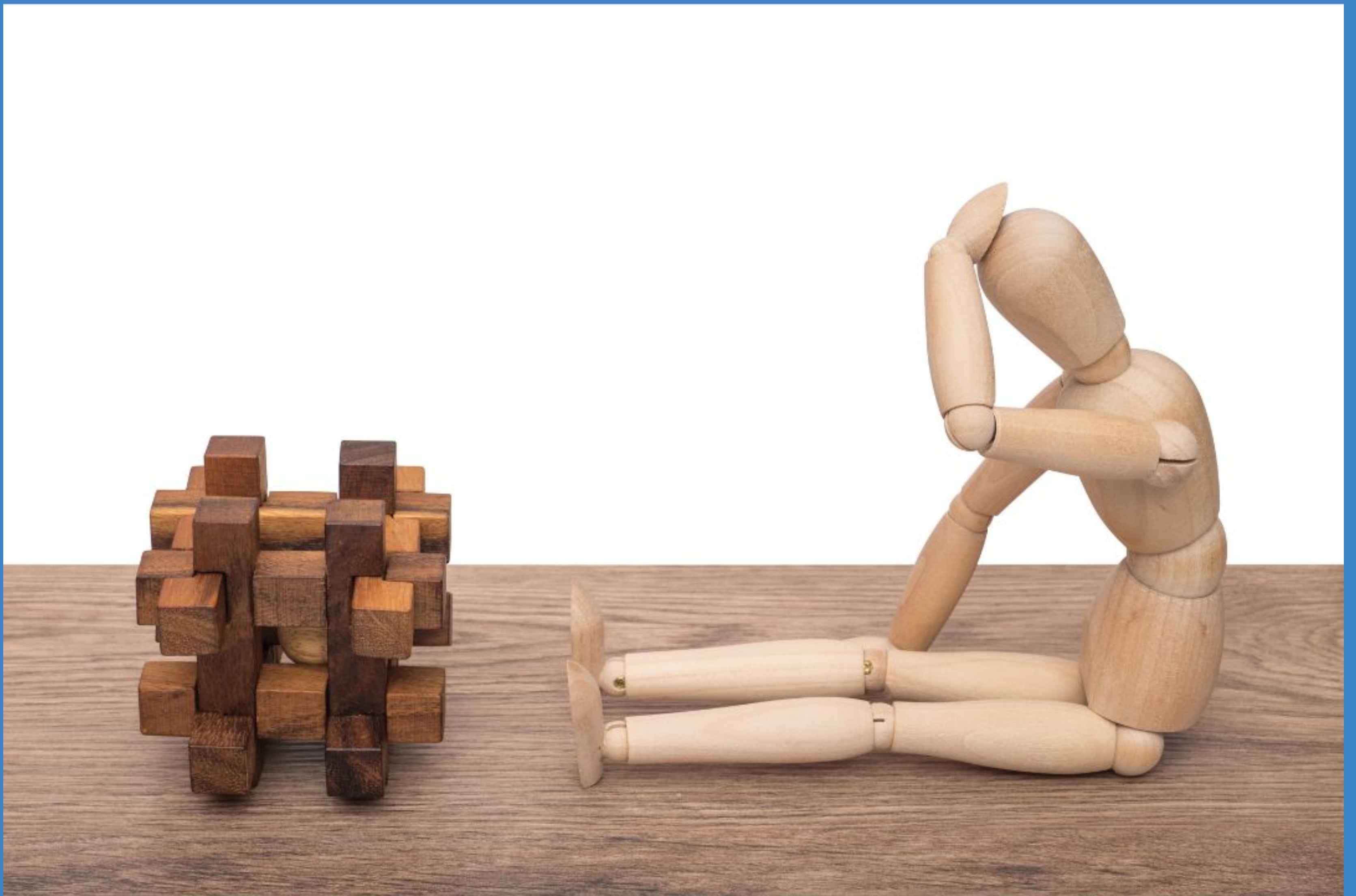


**Numerosos casos  
especiales de  
manera específica**





# Tarea difícil de articular en una solicitud



# Chatbots y asistentes



How can I help you today?

**Come up with concepts**  
for a retro-style arcade game

**Show me a code snippet**  
of a website's sticky header

**Explain why popcorn pops**  
to a kid who loves watching it in the microwave

**Write a message**  
that goes with a kitten gif for a friend on a rou...

Message ChatGPT...



ChatGPT can make mistakes. Consider checking important information.



# Platzibot



# Costos de uso de OpenAI: tokenización de texto



**Reto: Tokeniza 5 conversaciones de tu elección. Calcula su costo.**

# Formato de datos para fine-tuning de modelo

```
{"role": "system",  
"content": "Eres un  
asistente."},  
{"role": "user", "content":  
"¿Quién ganó el mundial de  
fútbol?"},  
{"role": "assistant",  
"content": "El mundial de  
2022 lo ganó Argentina"}
```

# Preparación de datos para fine-tuning



 **El modelo es tan bueno  
como la calidad de sus datos**





**Reto: aplica validación de formato y conteo de tokens con `data_val.jsonl`.**



# Fine-tuning de modelo de OpenAI



**Reto: aplica fine-tuning desde la notebook con el mismo dataset pero con una cantidad menor de registros.**

# Playground de OpenAI

Probar modelos sin código

# Pruebas al modelo con fine-tuning



**Reto:**

**Compara resultados y  
evalúa costos/beneficios  
de los modelos con  
fine-tuning.**

**Optimizar el  
modelo:  
ajuste de  
parámetros**



**Reto: itera los parámetros.  
Busca el mejor  
desempeño posible.**

# ¿Cómo has aplicado fine-tuning?

- 1. Formateo de datos**
- 2. Fine-tuning**
- 3. Evaluación**
- 4. Optimizar parámetros**



# Creación de chatbot en Telegram

**Procesando la  
entrada del  
usuario para  
el chatbot**

# Función main del chatbot

# Integración del modelo de OpenAI a Telegram



**Reto:** prueba integrando los **otros modelos** a los que aplicaste fine-tuning.

# Errores y excepciones de la API de OpenAI



**Reto:** montar el chatbot  
en un **servidor** para que  
esté **público**.

# **Recomendaciones finales y alternativas de proyectos**



# Beneficios

- Resultados de mayor calidad que el diseño de prompts.
- Capacidad para entrenar en más ejemplos de los que caben en un prompt.
- Ahorro de tokens debido a prompts más cortos.

# Pasos a seguir

- Incrementar datos de entrenamiento.
- Entrena un nuevo modelo.
- Integra tu modelo con fine tuning a otras apps.
- Explorar otras tareas.



**Comparte la versión final  
de tu chatbot con  
el link del repositorio  
de GitHub.**