

# Curso de Chatbots con OpenAI

Carla Marcela Florida Román

# ¿Cómo usar la API de OpenAI?

Con su documentación



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

**Cargar  
modelo de  
la API de  
OpenAI**

```
import openai
```

```
openai.api_key = "OPENAI_API_KEY"
```

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

```
print(response.choices[0].text)
```



**Reto: captura y  
compara el tiempo de  
ejecución de  
diferentes modelos.**

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



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



# Parámetros de Completion

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

```
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.**

**ChatCompletion**



```
openai.ChatCompletion.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??"}  
    ]  
)
```

# Diferencias

Feature	completion	ChatCompletion
Input	Un solo prompt	Una serie de mensajes
Output	Una sola completion	Un mensaje generado por el modelo.
Casos de uso	Tareas de un solo turno, como generar texto o responder preguntas	Conversaciones de múltiples turnos.
Modelos	Todos los modelos	gpt-3.5-turbo y gpt-4

# ¿Qué has aprendido de la API?

1. Navegar  
documentación
2. Cargar modelo
3. Completion
4. ChatCompletion

# **Fine-tuning a modelos de OpenAI**

# Clasificación

- ¿El modelo está haciendo declaraciones falsas?
- Análisis de sentimientos.
- Categorización para clasificación de correo electrónico.

# Generación condicional

- Anuncio atractivo basado en un artículo de Wikipedia.
- Extracción de entidades.

# Generación condicional

- Chatbot de atención al cliente.
- Descripción de producto basada en una lista técnica de propiedades.



# Chatbots y asistentes

[+ New Thread](#)

## ChatGPT

Examples

"Explain quantum computing in simple terms" →  
"Got any creative ideas for a 10 year old's birthday?" →  
"How do I make an HTTP request in Javascript?" →

Capabilities

Remembers what user said earlier in the conversation  
Allows user to provide follow-up corrections  
Trained to decline inappropriate requests

Limitations

May occasionally generate incorrect information  
May occasionally produce harmful instructions or biased content  
Limited knowledge of world and events after 2021

Light Mode

OpenAI Discord

Updates & FAQ

Log out

➤

Free Research Preview: ChatGPT is optimized for dialogue. Our goal is to make AI systems more natural to interact with, and your feedback will help us improve our systems and make them safer.



# Costos de uso de OpenAI: tokenización de texto



**Reto: Tokeniza 5 frases de tu elección en la herramienta. Calcula su costo de uso en modelos InstructGPT.**

**Formato de  
datos para  
entrenamiento  
de modelo**

# Formateo de datos

- Cada **prompt** debe terminar con un separador fijo para informar al modelo cuando finaliza la solicitud y comienza la finalización.

# Formateo de datos

- Cada **completion** debe comenzar con un espacio en blanco debido a nuestra tokenización.

# Formateo de datos

- Cada **completion** debe terminar con una secuencia de parada fija para informar al modelo cuando finaliza la finalización.

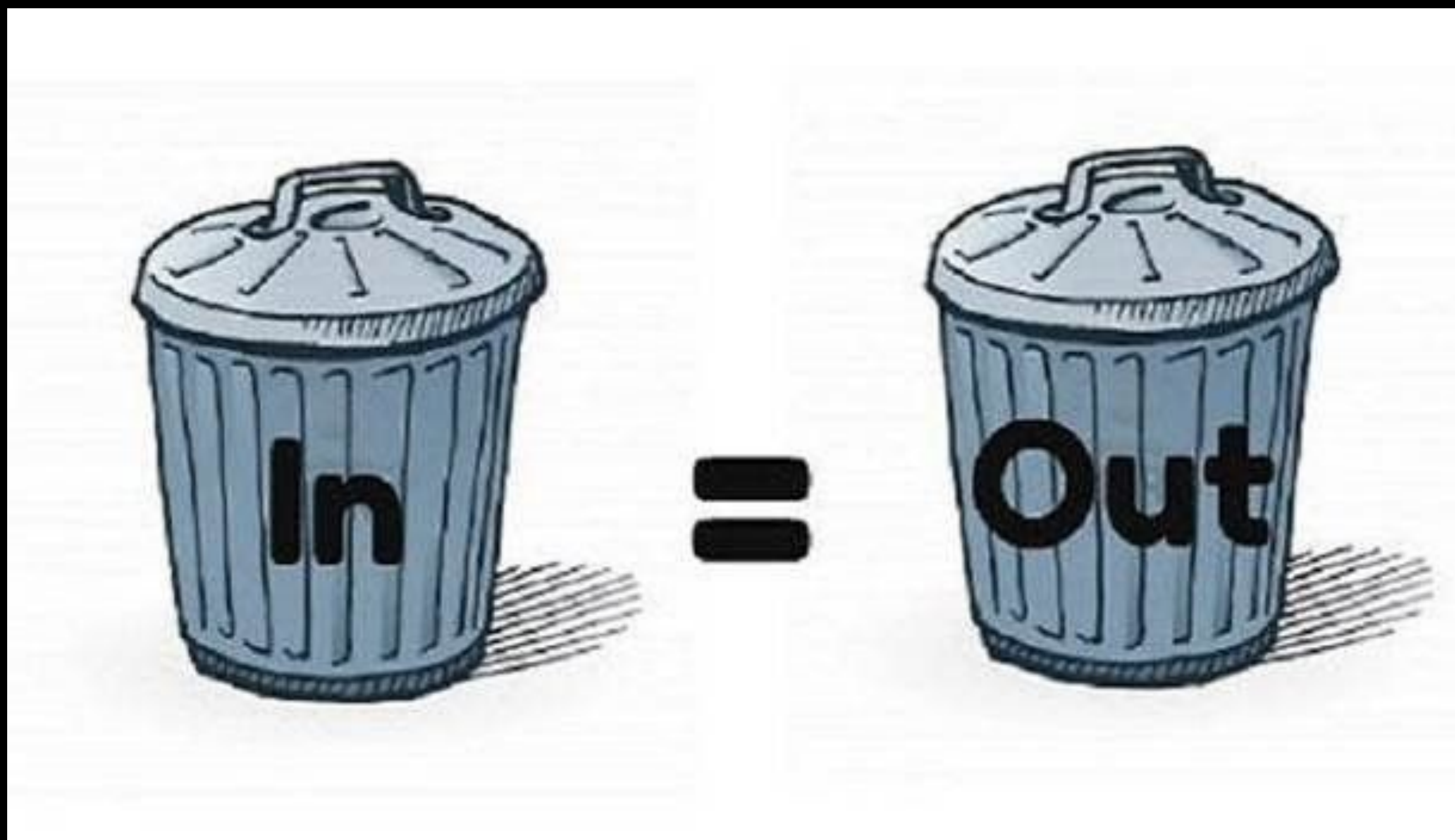
# Formateo de datos

- En el modelo entrenado se debe utilizar la **misma estructura de prompt** con la que fue entrenado.

# Preparación de datos para fine-tuning



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



# Fine-tuning del modelo

**Evaluar  
modelo con  
fine-tuning**



**Métricas automáticas**



**Diversidad y novedad**



# **Evaluación de dominio específico**



**Evaluación humana**



**Reto:** aplica fine-tuning a otro de los modelos con el mismo dataset, **compara resultados y evalúa resultados/costos.**



**Mejores  
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.

**Optimizar el  
modelo:  
ajuste de  
hiper  
parámetros**



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

# ¿Cómo has aplicado fine-tuning?

1. Formateo de datos
2. Fine-tuning
3. Evaluación
4. Optimizar hiper parámetros
5. Iterar

# 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.**