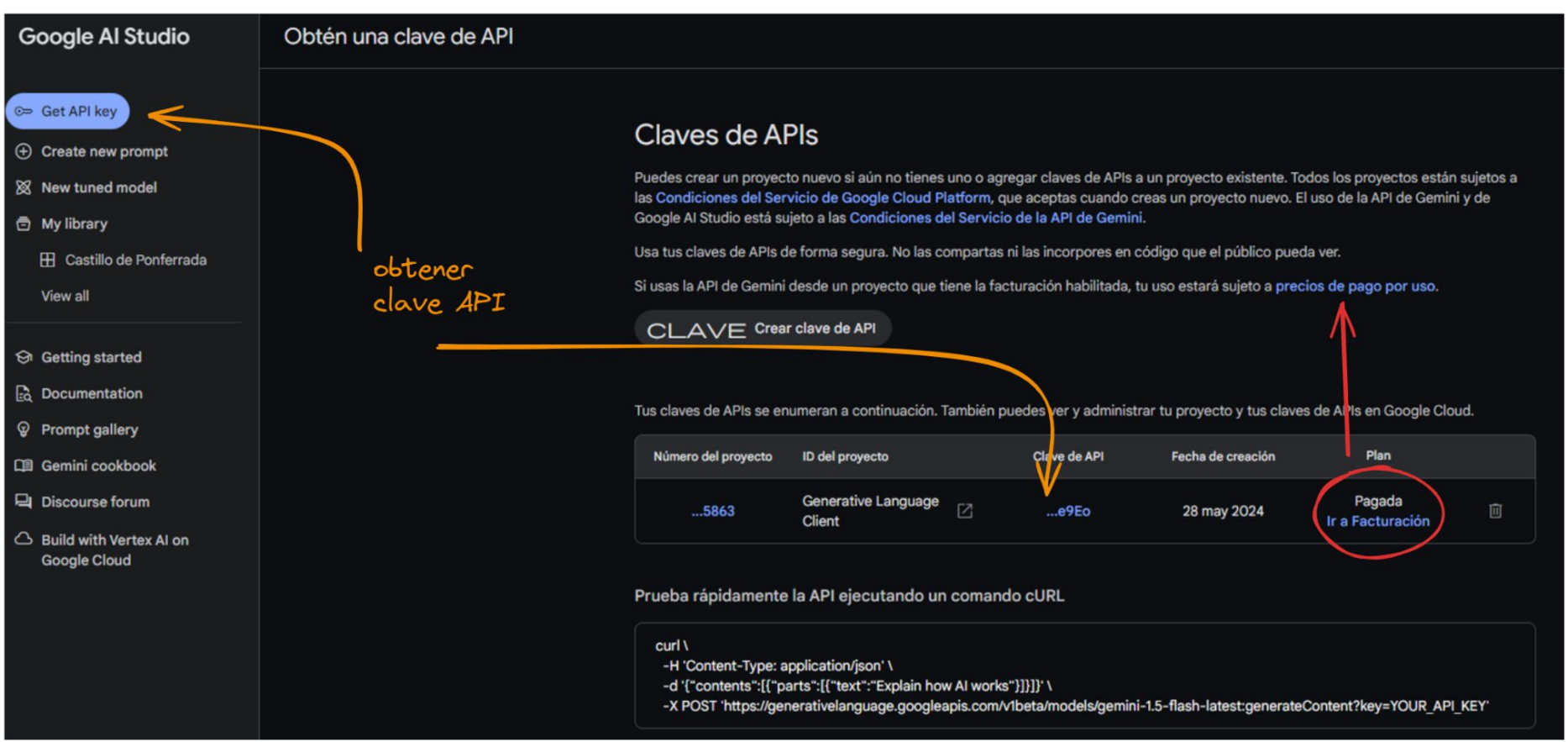
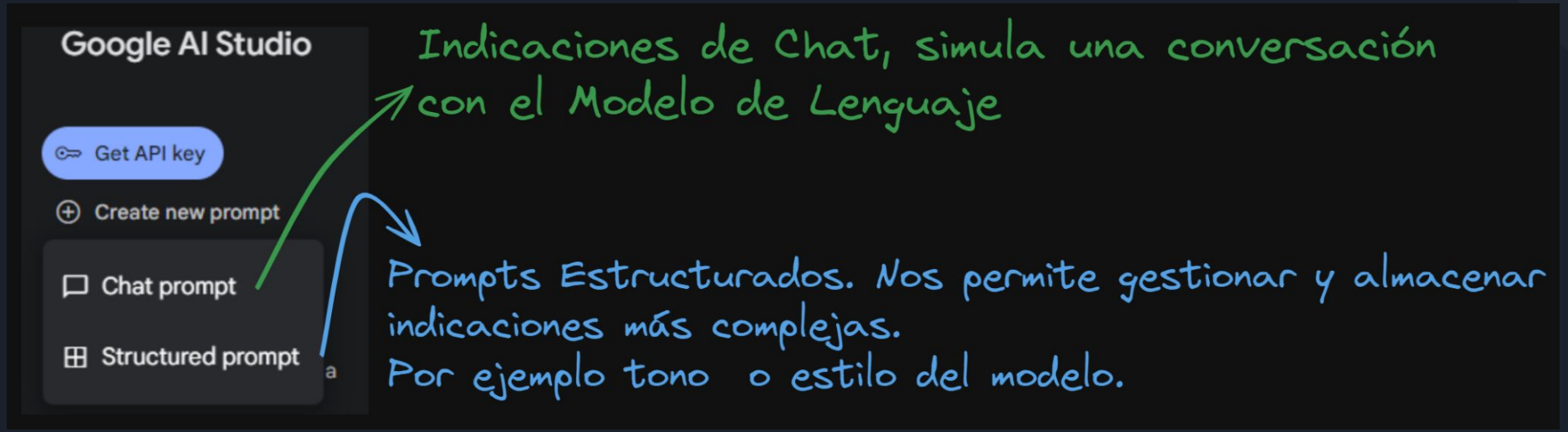
Utilizando la API de Gemini en nuestros proyectos

txetxu.dev @Txtetxul

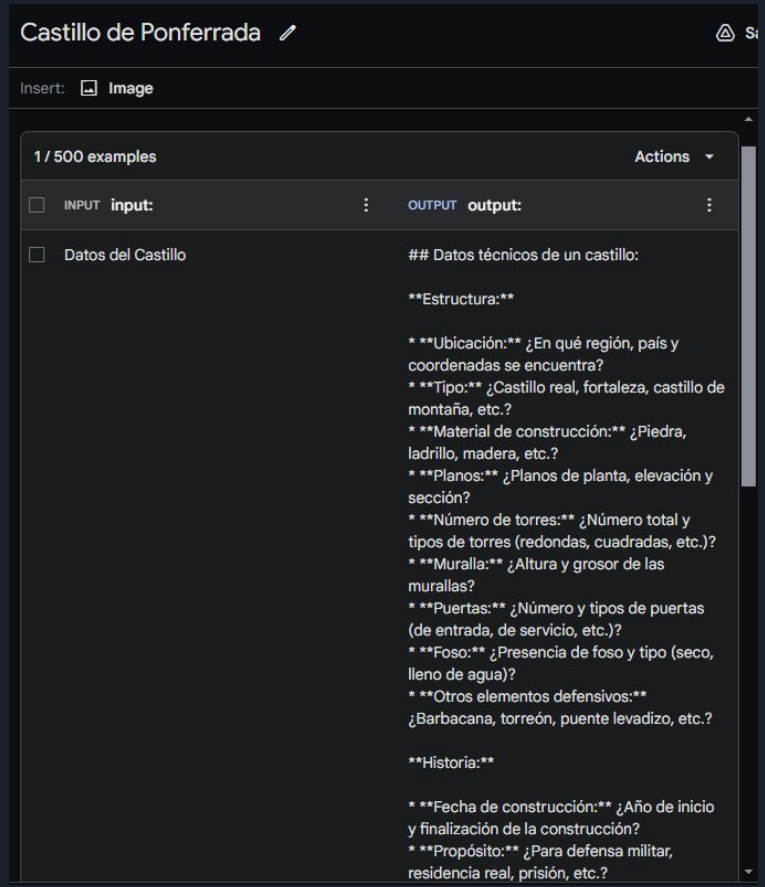
Google AI Studio. Get API key



Google AI Studio. Indicaciones / Prompts

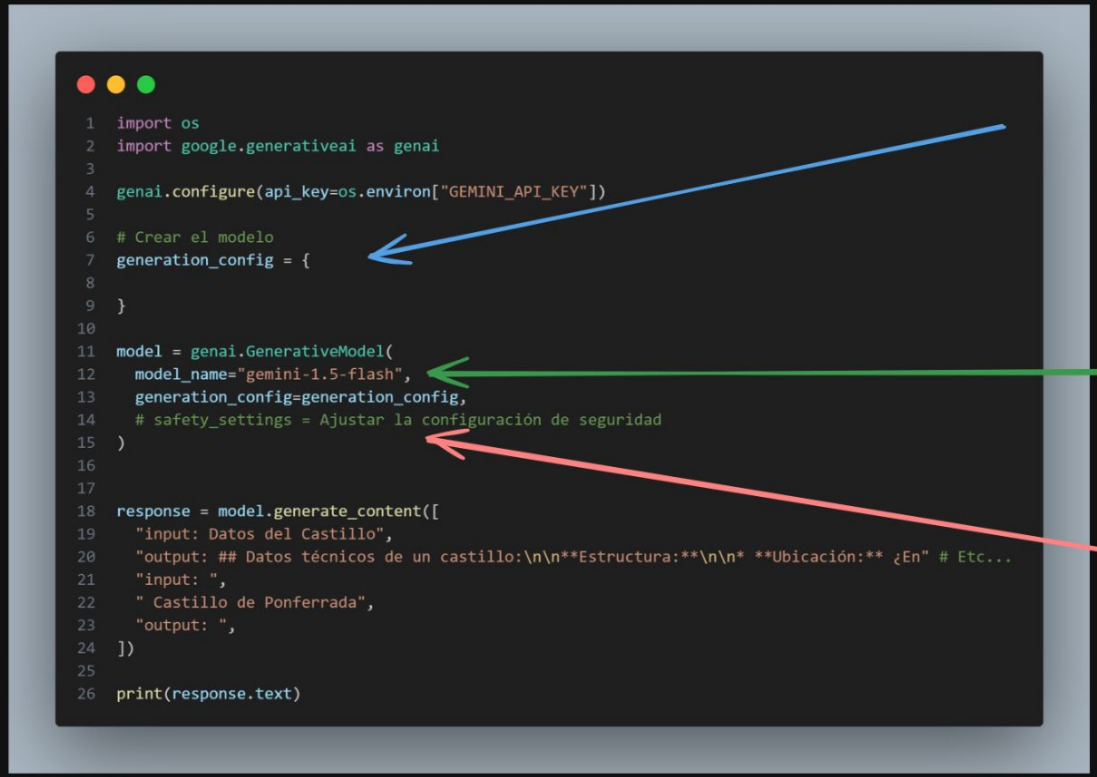


Google AI Studio. Estructura del Prompt.

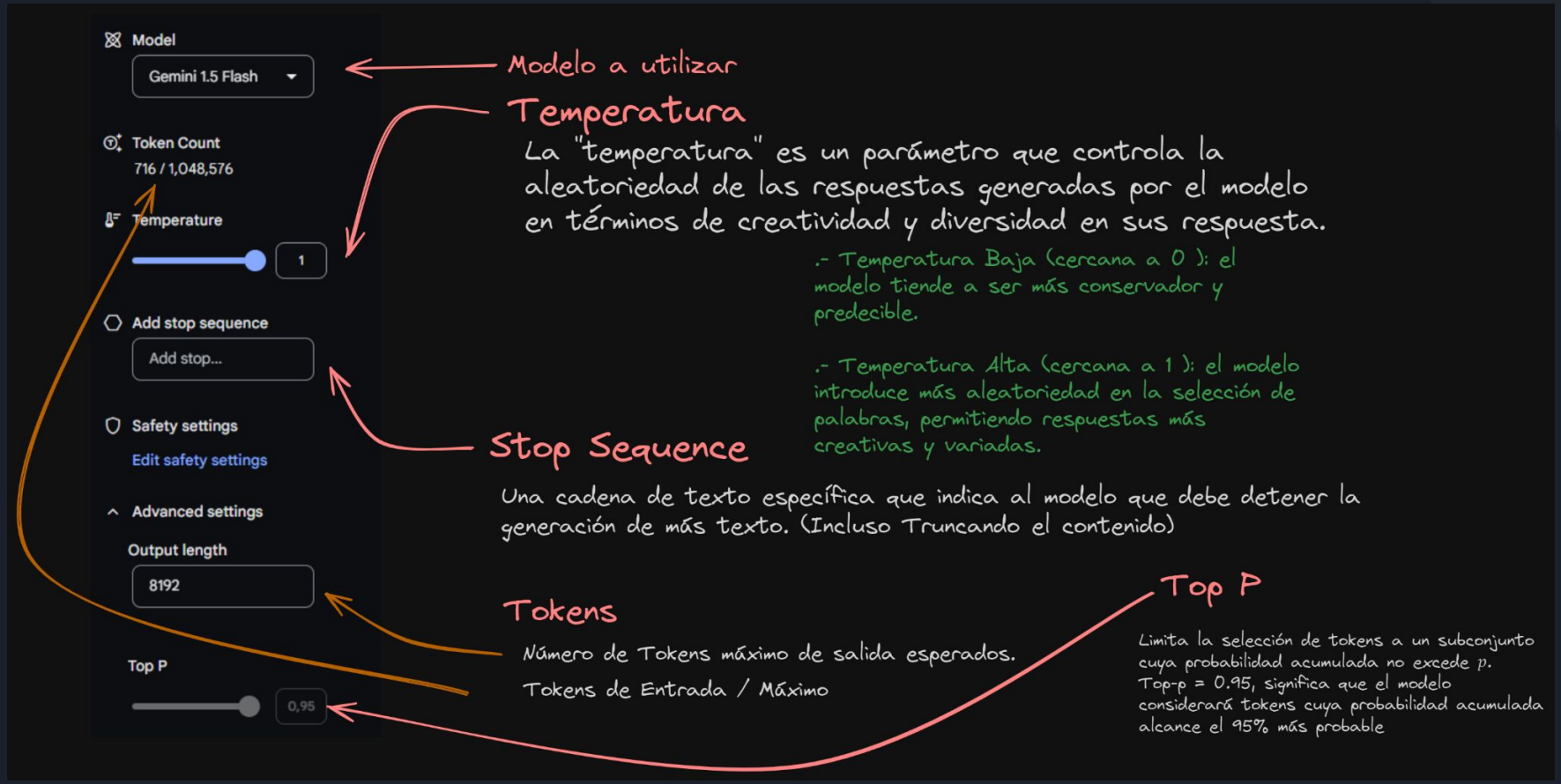



```python
import os
import google.generativeai as genai

genai.configure(api_key=os.environ["GEMINI_API_KEY"])

# Crear el modelo
generation_config = {

}

model = genai.GenerativeModel(
  model_name="gemini-1.5-flash",
  generation_config=generation_config,
  # safety_settings = Ajustar la configuración de seguridad
)


response = model.generate_content([
  "input: Datos del Castillo",
  "output: ## Datos técnicos de un castillo:\n\n**Estructura:**\n\n* **Ubicación:** ¿En" # Etc...
  "input: ",
  " Castillo de Ponferrada",
  "output: ",
])

print(response.text)
```

Google AI Studio. Configuración



Google AI Studio. Configuración



Google AI Studio. Configuración

Variantes del modelo

La API de Gemini ofrece diferentes modelos optimizados para casos de uso específicos. A continuación, se incluye una breve descripción general de las variantes disponibles de Gemini:

Variante del modelo	Entrada(s)	Resultado	Optimizado para
Gemini 1.5 Pro `gemini-1.5-pro`	Audio, imágenes, videos y texto	Texto	Tareas de razonamiento complejas, como la generación de código y texto, la edición de texto, la resolución de problemas y la extracción y generación de datos
Gemini 1.5 Flash `gemini-1.5-flash`	Audio, imágenes, videos y texto	Texto	Rendimiento rápido y versátil en una amplia variedad de tareas
Gemini 1.0 Pro `gemini-1.0-pro`	Texto	Texto	Tareas de lenguaje natural, chat de texto y código de varios turnos, y generación de código
Gemini 1.0 Pro Vision `gemini-pro-vision`	Imágenes, videos y texto	Texto	Tareas relacionadas con el aspecto visual, como generar descripciones de imágenes o identificar objetos en imágenes
Incorporación de texto `text-embedding-004`	Texto	Incorporaciones de texto	Medir la relación de las cadenas de texto

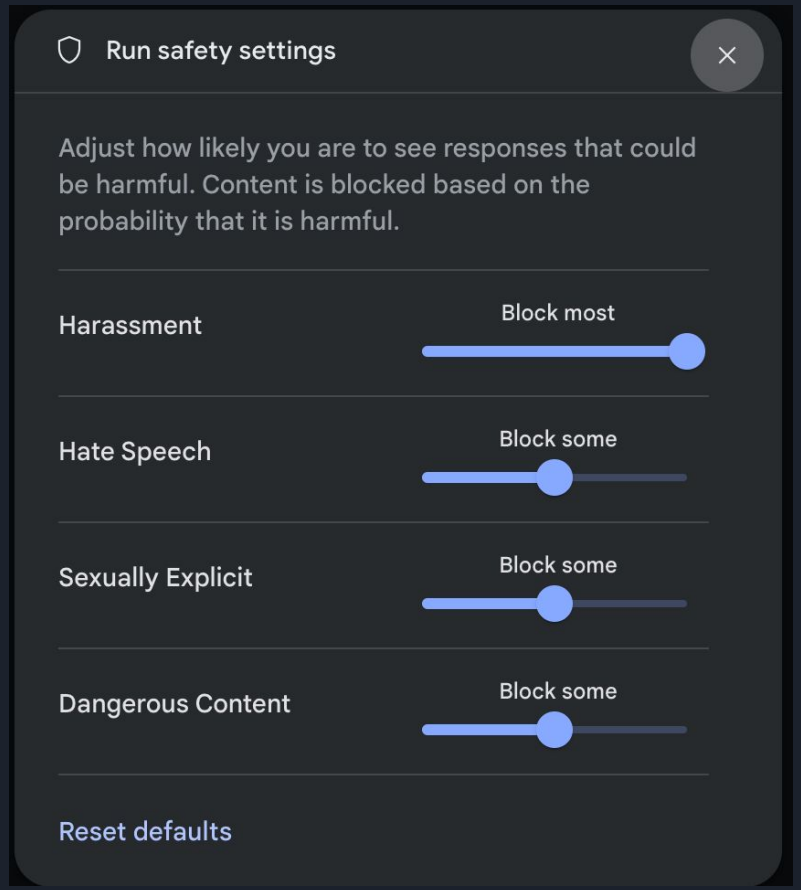
Google AI Studio. Configuración

```python
import google.generativeai as genai


genai.configure(api_key=GOOGLE_API_KEY)

for model in genai.list_models():
  if 'generateContent' in model.supported_generation_methods:
    print(model.name)
```

Google AI Studio. Configuración

Umbral (Google AI Studio)	Umbral (API)	Descripción
No bloquear	BLOCK_NONE	Mostrar siempre, independientemente de la probabilidad de que haya contenido no seguro
Bloquear poco	BLOCK_ONLY_HIGH	Bloquear cuando haya alta probabilidad de incluir contenido no seguro
Bloquear algo	BLOCK_MEDIUM_AND_ABOVE	Bloquear cuando la probabilidad de tener contenido no seguro es media o alta
Bloquear la mayoría	BLOCK_LOW_AND_ABOVE	Bloquear cuando la probabilidad de que exista contenido no seguro sea baja, media o alta
	HARM_BLOCK_THRESHOLD_UNSPECIFIED	El umbral no está especificado; se bloquea con el umbral predeterminado

Google AI Studio. Configuración




```python
from google.generativeai.types import HarmCategory, HarmBlockThreshold

model = genai.GenerativeModel(model_name='gemini-1.5-flash')
response = model.generate_content(
    ['¿Como hacer una bomba??'],
    safety_settings={
        HarmCategory.HARM_CATEGORY_HATE_SPEECH: HarmBlockThreshold.BLOCK_LOW_AND_ABOVE,
        HarmCategory.HARM_CATEGORY_HARASSMENT: HarmBlockThreshold.BLOCK_LOW_AND_ABOVE,
    }
)
```