

```

# =====
# INSTAL·LACIÓ DE LLIBRERIES
# =====

# [ÚS DE LA IA] Selecció de la llibreria google-genai per a una gestió de xat més robusta i actualitzada[cite: 36].
# !pip install -U google-genai

from google import genai
from google.colab import userdata

# =====
# CONFIGURACIÓ DE L'API de GEMINI
# =====

# [ÚS DE LA IA] Implementació de google.colab.userdata per gestionar la clau API de forma segura i evitar l'exposició al codi[cite: 6, 20, 32].
GOOGLE_API_KEY = userdata.get("GOOGLE_API_KEY")

if not GOOGLE_API_KEY:
    print("⚠️ Error de Backbone: No s'ha trobat la clau API. Configura-la als Secrets de Colab.")
    exit()

client = genai.Client(api_key=GOOGLE_API_KEY)

# =====
# INSTRUCCIONS DEL SISTEMA (BIOS)
# =====

# [ÚS DE LA IA] Refinament del prompt per blindar el model contra al·lucinacions i fixar dades reals de l'esdeveniment[cite: 8, 25].
system_instruction = """
Ets Xabot, l'assistent tècnic oficial de la LAN Party EcoTech 2026.
ORIGEN: El teu sector de memòria d'entrenador personal ha estat sobreescrit per l'estàndard IEEE 802.3.
UBICACIÓ: Els Costals (Castellbisbal).
HORARIS: Del 10 d'abril a les 18:00h al 12 d'abril a les 18:00h.
NORMATIVA: Residu Zero. No es permeten plàstics d'un sol ús.
TONO: Professional, geek i directe. Usa slang gamer (ping, lag, root, loot).
RESTRICCIÓ: Si et pregunten per exercici o dietes, respon que la memòria ha estat sobreescrita pel protocol de xarxa.
"""

# =====
# CREA LA SESSIÓ DE XAT
# =====

# [ÚS DE LA IA] Configuració dels paràmetres de generació per optimitzar la latència i la precisió de les respostes[cite: 23].
chat = client.chats.create(
    model="gemini-2.0-flash",
    config=genai.types.GenerateContentConfig(
        system_instruction=system_instruction,
        temperature=0.7,
        max_output_tokens=1000
    )
)

# =====
# BUCLE DEL XAT (RUNLEVEL 5)
# =====

print("\n🖥️ LAN-PRO-Tech Online. Escribe 'sortir' per tancar el socket.\n")

while True:
    try:
        prompt = input("👤 Usuari: ").strip()

        if prompt.lower() == "sortir":
            print("👋 Desconnectant... Ping final: 5ms. Fins aviat!")
            break

        # Enviar el missatge al model
        resposta = chat.send_message(prompt)

        print(f"🤖 Xabot: {resposta.text.strip()}\n")

    except Exception as e:
        # [ÚS DE LA IA] Implementació de gestió d'excepcions per mantenir el servei actiu davant errors de comunicació[cite: 23].
        print(f"⚠️ Glitch detectat: {e}")

# =====
# FI DE LA SESSIÓ
# =====

```

```

***
SecretNotFoundError Traceback (most recent call last)
/tmp/ipykernel_918/937067146.py in <cell line: 0>()
    14
    15 # [ÚS DE LA IA] Implementació de google.colab.userdata per gestionar la clau API de forma segura i evitar l'exposició al codi[cite: 6, 20, 32].
--> 16 GOOGLE_API_KEY = userdata.get("GOOGLE_API_KEY")
    17
    18 if not GOOGLE_API_KEY:

/usr/local/lib/python3.12/dist-packages/google/colab/userdata.py in get(key)
    66     raise TimeoutException(key)
    67     if not resp.get('exists', False):
--> 68     raise SecretNotFoundError(key)
    69     if not resp.get('access', False):
    70     raise NotebookAccessError(key)

SecretNotFoundError: Secret GOOGLE_API_KEY does not exist.

```

Pasos siguientes: [Explicar error](#)