

PC Doctor IA

Federico Gimenez Sabbattini

IA: Generacion de Prompts -Carreras Intensivas

Comisión #76165

## **Resumen:**

PC Doctor AI es una aplicación web basada en inteligencia artificial diseñada para ofrecer un diagnóstico inicial de problemas técnicos en PCs. Esta solución busca reducir la carga en los técnicos especializados y mejorar la eficiencia, permitiendo a los usuarios resolver problemas simples por sí mismos.

## **Introducción:**

### **Nombre del proyecto:**

PC Doctor AI

### **Presentación del problema a abordar:**

En el ámbito del servicio técnico de PCs, los usuarios frecuentemente enfrentan problemas que pueden ser resueltos con simples instrucciones, sin la necesidad de recurrir a un técnico especializado. Sin embargo, debido a la falta de conocimientos técnicos, muchos usuarios buscan asistencia profesional para problemas menores, lo que incrementa los costos y tiempos de espera.

### **Desarrollo de la propuesta de solución:**

La solución propuesta es la creación de una aplicación web interactiva basada en IA que guíe a los usuarios a través de una serie de preguntas y diagnósticos automáticos. Utilizando el modelo de Generative AI (Gemini) de Google, la aplicación interpretará las respuestas de los usuarios y ofrecerá soluciones básicas. En casos donde el problema no pueda ser resuelto por el usuario, la aplicación proporcionará información de contacto de un técnico especializado.

### **Justificación de la viabilidad del proyecto:**

El proyecto es viable técnicamente gracias a la disponibilidad de herramientas avanzadas como Google Generative AI y NLTK para el procesamiento del lenguaje natural. Además, la implementación en una plataforma web permite que sea accesible para cualquier usuario con acceso a internet.

## **Objetivos:**

1. Reducir la carga en técnicos especializados filtrando problemas menores.
2. Mejorar la eficiencia del servicio técnico.
3. Ahorrar costos y tiempo para los usuarios.
4. Proporcionar una herramienta interactiva y fácil de usar.

## **Metodología:**

El proyecto se llevará a cabo siguiendo estos pasos:

1. Definir y entrenar el modelo de IA con prompts específicos.
2. Implementar la lógica de diagnóstico utilizando Google Generative AI.
3. Desarrollar la interfaz de usuario en una plataforma web (por ejemplo, usando Streamlit).
4. Realizar pruebas y ajustes según sea necesario para mejorar la precisión de los diagnósticos.

## Herramientas y tecnologías:

1. **Google Generative AI (Gemini):** Utilizado para el procesamiento de lenguaje natural y generación de diagnósticos.
2. **NLTK:** Biblioteca para el procesamiento de lenguaje natural.
3. **Streamlit:** Plataforma para el desarrollo de la aplicación web interactiva.
4. **Python:** Lenguaje de programación principal del proyecto.

## Implementación:

Aquí tienes el código base del proyecto:

```
import google.generativeai as genai
import nltk
import requests

# Configurando la API KEY de Generative AI de Google
genai.configure(api_key="API-KEY")
model = genai.GenerativeModel("gemini-1.5-flash")

generation_config = genai.GenerationConfig(
    max_output_tokens=1000,
    temperature=0.1,
)

# Instrucción del sistema para el diagnóstico de PC
system_instruction = "Eres un experto en servicio técnico de PC. Diagnostica problemas técnicos descritos por los usuarios y proporciona soluciones básicas y concisas, bastante simples. Si el problema no puede ser resuelto por el usuario, indica que se necesita asistencia técnica."

# Descargar los datos necesarios de NLTK
nltk.download('punkt')
nltk.download('averaged_perceptron_tagger')

# Definir la función de diagnóstico utilizando Google Generative AI con el prompt específico
```

```

def gemini_diagnosis(user_input):
    prompt = f"{system_instruction}\n\nUsuario: {user_input}\nDiagnóstico:"
    response = model.generate_content(
        prompt,
        generation_config=generation_config
    )
    diagnosis = response.text.strip()
    return diagnosis

def main():
    print("PC Doctor AI - Diagnóstico Técnico")
    user_input = input("Describa su problema aquí: ")
    diagnosis = gemini_diagnosis(user_input)
    print("Diagnóstico: ", diagnosis)

    # Proporcionar información de contacto de un técnico especializado
    if "asistencia técnica" in diagnosis:
        print("Por favor, contacte a un técnico especializado en el siguiente enlace:")
        print("Información de contacto del técnico: 123456789")

if __name__ == "__main__":
    main()

```

### Instalación y Configuración:

- Primero, se instalan las bibliotecas necesarias (**google-generativeai**, **nltk** y **requests**) para que el script funcione correctamente. Luego, se configura la API de Google Generative AI utilizando tu clave de API y se inicializa el modelo generativo.

### Instrucciones del Sistema:

- Se define una instrucción clara para el modelo, indicándole que actúe como un experto en servicio técnico de PC. La instrucción le dice al modelo que diagnostique problemas técnicos y proporcione soluciones o indique si se necesita asistencia técnica.

### Descarga de Datos de NLTK:

- Se descargan los recursos necesarios del Natural Language Toolkit (NLTK) que facilitan el procesamiento del lenguaje natural.

### Función de Diagnóstico:

- Se define una función (`gemini_diagnosis`) que toma la descripción del problema por parte del usuario, la combina con las instrucciones del sistema y utiliza el modelo generativo de Google para crear un diagnóstico.

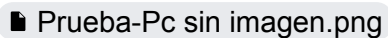
### **Función Principal:**

- La función principal (`main`) se encarga de recibir la descripción del problema del usuario, llamar a la función de diagnóstico, imprimir el resultado y proporcionar información de contacto si se necesita asistencia técnica.

### **Ejecución del Programa:**

- Un bloque de control asegura que la función principal solo se ejecute si el script se ejecuta directamente, no cuando se importa como un módulo.

### **Resultados:**

 Prueba-Pc sin imagen.png

La solución que brinda son simples y bastante acordes a lo que un técnico indicaría como primer chequeo

### **Conclusiones:**

La versión inicial da resultados bastante alineados con la idea original. Las instrucciones y la guía paso a paso para sortear los problemas más básicos son simples y fáciles de seguir, pueden ayudar a sortear algunos de los problemas más simples sin que el usuario genere un fallo o daño al equipo lo que ahorra tanto tiempo como dinero para el usuario.