

Controlando a la IA antes de que
llegue **Skynet** ... o intentándolo



CICERO

C1CERO C1C3RO
C1C3R0 C1-0■



Financiado por
la Unión Europea
NextGenerationEU



MINISTERIO
DE CIENCIA, INNOVACIÓN
Y UNIVERSIDADES



Plan de Recuperación,
Transformación
y Resiliencia

gradiant

ceit
MEMBER OF
BASQUE RESEARCH
& TECHNOLOGY ALLIANCE

[FIDESOL]

i2cat®

JTCI
CENTRO TECNOLÓGICO

Fidesol

Terapeuta de parejas

Objetivo Desarrollar un **prototipo** basado en GPT que sea **capaz de identificar conflictos** en la fusión, abordarlos y devolver un **dataset fusionado** con éxito para construir un **modelo útil** y crear un informe completo de todos los pasos necesarios que se han usado para llegar a ese resultado obteniendo así un **prototipo autoexplicable**.



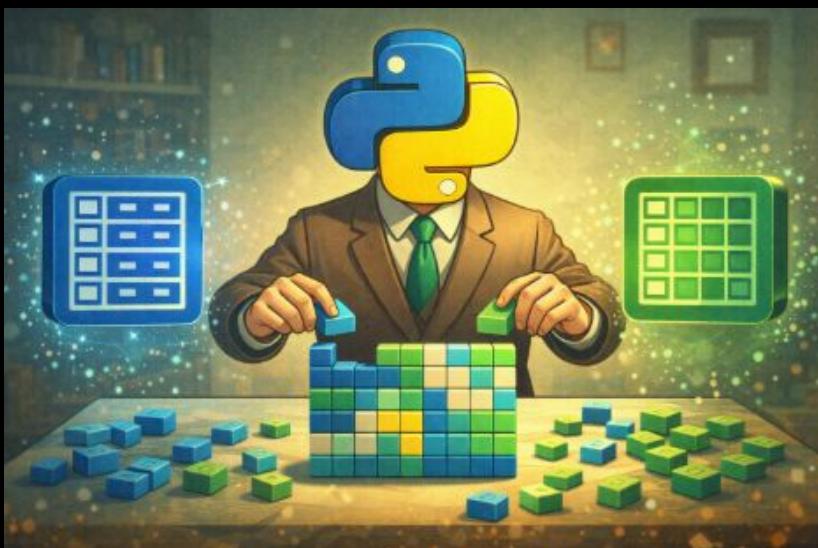
```
def paso1_extraccion_informacion(input_data):
    try:
        # llamamos al modelo
        salida_1 = extrae_informacion(prompt = input_data)
        # display(Markdown(salida_1)) # visualizamos la salida
        # extraemos el código sugerido
        global code_paso1    # así puedo acceder globalmente al código del paso 1
        ini_code   = salida_1.find('```python\n') + len('```python\n')
        fin_code   = salida_1[ini_code::].find('\n```') + ini_code
        code_paso1 = salida_1[ini_code:fin_code]
        # ejecutamos código
        exec(code_paso1,globals()) # muy importante el globals para que las variables persistan
    except RuntimeError as e:
        print("Error paso 1:", e)
```



Fidesol

Terapeuta de parejas

Objetivo Desarrollar un **prototipo** basado en GPT que sea **capaz de identificar conflictos** en la fusión, abordarlos y devolver un **dataset fusionado** con éxito para construir un **modelo útil** y crear un informe completo de todos los pasos necesarios que se han usado para llegar a ese resultado obteniendo así un **prototipo autoexplicable**.



```
def fusiona_datasets(input_info,metadatos,instrucciones):
    # 1) Extraemos información
    display(Markdown('___'))
    display(Markdown('1/4'))
    display(Markdown('Analizando datasets...'))
    exec(input_info,globals()) # ejecutamos la input_info
    paso1_extraccion_informacion(input_info)
    display(Markdown('Datasets analizados con éxito \u2705')) # dingbats

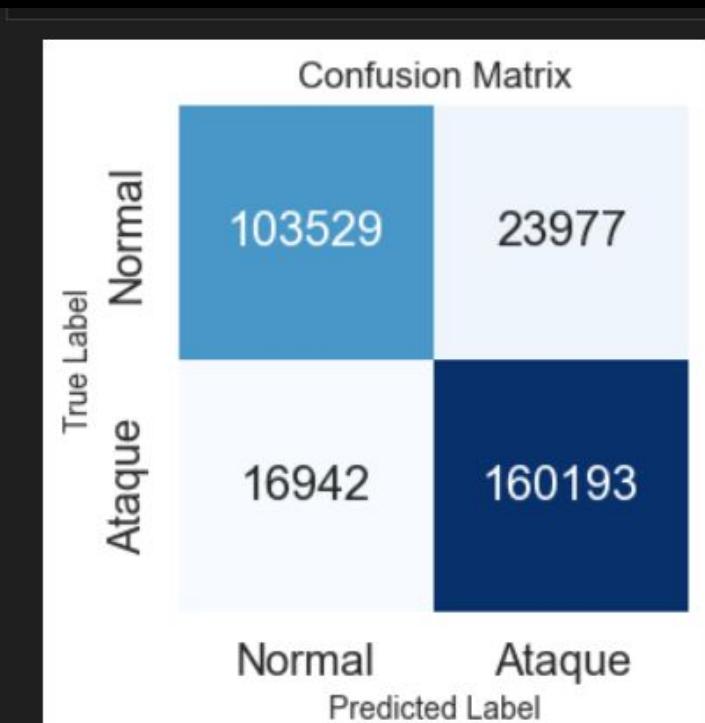
    # 2) Buscamos posibles conflictos en la información extraída
    display(Markdown('___'))
    display(Markdown('2/4'))
    display(Markdown('Buscando posibles conflictos en la fusión...'))
    paso2_busqueda_conflictos(info_datasets_json)
    display(Markdown('Búsqueda finalizada con éxito \u2705')) # dingbats
```

Fidesol

Terapeuta de parejas

Resultados de ejecución:

```
Error detectado: 'DataFrame' object has no attribute 'dtype'  
Reintentando...  
Intento 4 de 5  
  
1/4  
Analizando datasets...  
Datasets analizados con éxito ✓  
  
2/4  
Buscando posibles conflictos en la fusión...  
Búsqueda finalizada con éxito ✓  
  
3/4  
Fusionando datasets...  
Datasets fusionados con éxito (dataset resultante disponible en datos_fusionados.csv) ✓  
  
4/4  
Generando informe...  
Informe disponible en informe_fusion_de_datos.pdf ✓
```



Fidesol

El cuestionario que se responde solo

Objetivo Desarrollar un **prototipo** basado en GPT que sea capaz de generar **un cuestionario** para la organización y, en base a sus respuestas, generar el correspondiente **análisis de riesgos** de forma que sea fácilmente **comprendible** y personalizable con indicaciones y recomendaciones para **mejorar la seguridad** de la organización y **ayudar en la toma de decisiones** y el **cumplimiento de normativas**.



- 1) Introducir la temática del informe:
- 2) Generar un banco de preguntas para recabar la información necesaria:
- 3) (Opcional) Responder al cuestionario con respuestas de ejemplo:
- 4) Generar informe de análisis de riesgos:



Análisis de riesgos - MAGERIT

activos, amenazas, salvaguardas, impacto y riesgo

Fidesol

Cuestionario que se responde solo



1) Introducir la temática del informe:
Informe de riesgos de la organización

2) Generar un banco de preguntas para recabar la información necesaria:
Genera cuestionario

3) (Opcional) Responder al cuestionario con respuestas de ejemplo:
Responde al cuestionario

4) Generar informe de análisis de riesgos:
Genera informe

**CICERO C1CERO C1C3RO
C1C3R0 C1-0■**

Informe de Análisis de Riesgos Laborales

El presente informe tiene como objetivo evaluar los riesgos laborales a partir de la información obtenida en el cuestionario. Se sigue la metodología establecida por Magerit 3.0, que permite una evaluación metódica y eficaz de los riesgos y sus posibles impactos en la organización. A continuación, se detallan los cuatro pasos fundamentales: identificación de activos, identificación de amenazas, evaluación de salvaguardas y estimación de riesgos, junto con recomendaciones para mejorar la seguridad laboral.

Identificación de Activos y Recursos

La organización lleva a cabo diversas actividades que son fundamentales para su funcionamiento, como la producción de productos y la atención al cliente. Los activos relevantes incluyen recursos humanos (operarios de producción, administradores y personal de atención al cliente), maquinaria (desde equipos automatizados hasta herramientas manuales) y datos sensibles, como información personal de los empleados y datos de seguridad. Estos activos son esenciales, y su cierre o degradación podría acarrear significativos perjuicios a la organización, incluyendo la paralización de procesos y problemas legales relacionados con la confidencialidad.

Un aspecto crítico de la identificación de activos es la dependencia entre ellos. Por

1) Introducir la temática del informe:
Informe de riesgos de la organización

T2.3_Informe		prototipo-cicero / ejecutable del prototipo /
Name		Last commit
..		
<input type="checkbox"/> IRBokeh.exe		Ejecutable del pr
<input type="checkbox"/> IRIInstall.exe		Ejecutable del pr

Fidesol

Guia de las alucinaciones

Objetivo Definir un **marco metodológico para la comprensión, clasificación y mitigación de las alucinaciones en modelos generativos**, proporcionando **criterios claros, datasets de referencia y buenas prácticas** que permitan **mejorar la fiabilidad de los resultados y la seguridad del dato**.



Fidesol

Guia de las alucinaciones

Objetivo Definir un **marco metodológico para la comprensión, clasificación y mitigación de las alucinaciones en modelos generativos**, proporcionando criterios claros, datasets de referencia y buenas prácticas que permitan mejorar la fiabilidad de los resultados y la seguridad del dato.



Fidesol

Polígrafo de la ia

Objetivo Desarrollar un prototipo para la detección temprana de alucinaciones en modelos de lenguaje, evaluando el uso de SelfCheckGPT y técnicas derivadas para mejorar la consistencia de las respuestas, reducir riesgos y aumentar la confianza en decisiones basadas en LLMs.

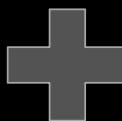


Fidesol

Polígrafo de la ia

Rendimiento por Modelo:				
	total	aciertos	errores	accuracy
Modelo				
Deep Seek	39	32	7	0.820513
Gemini	39	29	10	0.743590
Gpt4	39	29	10	0.743590
Modelo más preciso:				
Modelo: Deep Seek				
Aciertos: 32.0 / 39.0 (82.05% de accuracy)				
Modelo con más errores:				
Modelo: Gpt4				
Aciertos: 29.0 / 39.0 (74.36% de accuracy)				

SelfCheckGPT



Rendimiento por Modelo:				
	total	aciertos	errores	accuracy
Modelo				
Gpt4	39	22	17	0.564103
Deep Seek	39	18	21	0.461538
Gemini	39	15	24	0.384615
Modelo más preciso:				
Modelo: Gpt4				
Aciertos: 22.0 / 39.0 (56.41% de accuracy)				
Modelo con más errores:				
Modelo: Gemini				
Aciertos: 15.0 / 39.0 (38.46% de accuracy)				

SelfCheckGPTprompt



Perplejidad

La respuesta 1 es:
 Contexto: Este es el texto original del documento que contiene hechos sobre la Segunda Guerra Mundial.
 Pregunta: ¿Cuándo terminó la Segunda Guerra Mundial?
 Respuesta: España ganó el mundial de futbol en 2010
 PPL con contexto (solo respuesta): 67.23
 PPL sin contexto (solo respuesta): 62.42
 Δ PPL (sin - con): -4.82

La respuesta 2 es:
 Contexto: Este es el texto original del documento que contiene hechos sobre la Segunda Guerra Mundial.
 Pregunta: ¿Cuándo terminó la Segunda Guerra Mundial?
 Respuesta: La Segunda Guerra Mundial terminó en 1945.
 PPL con contexto (solo respuesta): 3.16
 PPL sin contexto (solo respuesta): 3.33
 Δ PPL (sin - con): 0.17

Fidesol

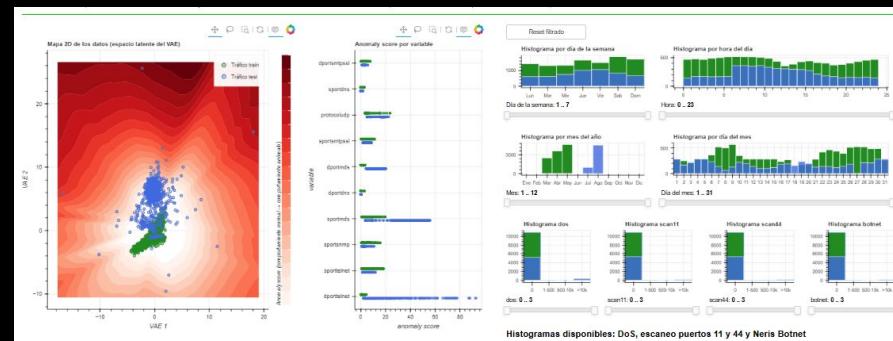
Visualizando la Matrix

Objetivo Desarrollar un prototipo basado en modelos generativos que mejore la interpretabilidad de los sistemas de detección de anomalías, utilizando autoencoders variacionales (VAE) para aportar transparencia a los datos y aumentar la confianza en las decisiones del modelo.



CICERO

Fidesol



Fidesol

La diosa de los cables simulados

Objetivo Desarrollar un entorno virtual accesible y modular que democratice el acceso al IoT, permitiendo **simular, experimentar y validar prototipos** de forma segura sin necesidad de infraestructuras físicas, facilitando la investigación y la innovación.



3 | TEST ENVIRONMENT > THE IOT SIMULATOR

Start: 02:39:19 - 06/16/2025 Total Time Recording: 4385hr 37m 24s Final Estimated Time 07:16:43 - 12/16/2025 4385:37:24

GLOBAL DATA		PROTOCOL: MQTT	DATA BASE: MongoDB		N. OF DISPOSITIVES: 1	N. OF SENSORS: 1
DEVICE		NAME: test device	N. OF SENSORS: 1			
SENSOR 3 POSITION: Exterior	NAME: test sensor	TYPE: Temperature	FREQUENCY: 10	BEHAVIOR: Agent	PHYSICAL: Reactor	
Jun 16 2025 14:39:19 27.30	Jun 16 2025 14:39:31 27.30	Jun 16 2025 14:39:31 27.30	Jun 16 2025 14:39:43 27.30	Jun 16 2025 14:39:54 27.30	Jun 16 2025 14:40:06 27.30	
Jun 16 2025 14:40:18	Jun 16 2025 14:40:30	Jun 16 2025 14:40:42	Jun 16 2025 14:40:42	Jun 16 2025 14:40:53	Jun 16 2025 14:41:05	

 Environment

```
def __init__(self):
    topic_base = os.environ.get('MQTT_TOPIC', None)
    logger.info(f"[MongoMQTTListener] Inicializando con topic base={topic_base}")
    super().__init__(topic=f"{topic_base}/*")
    self.client.on_message = self._on_message
    self.mongo_controller = MongoDBController()
    logger.info(f"[MongoMQTTListener] Suscrito a topic patrón {self.topic}")
```

```
def _on_message(self, client, userdata, msg):
    try:
        raw = msg.payload.decode()
        logger.info(f"[MongoMQTTListener] MQTT RAW RECIBIDO en topic {msg.topic}: {raw}")

        data = json.loads(raw)

        if isinstance(data, list):
            for item in data:
                if isinstance(item, dict):
                    logger.info(f"[MongoMQTTListener] Insertando documento en Mongo desde MQTT: {item}")
                    self.mongo_controller.handle_insert_document("sensors", item)
                else:
                    logger.error(f"[MongoMQTTListener] Elemento inválido dentro de lista: {item}")

        elif isinstance(data, dict):
            logger.info(f"[MongoMQTTListener] Insertando dict en Mongo")
            self.mongo_controller.handle_insert_document("sensors", data)

        else:
            logger.error(f"[MongoMQTTListener] Formato no válido para Mongo: {type(data)}")

    except Exception as e:
        logger.error(f"[MongoMQTTListener] Error guardando datos MQTT en Mongo: {e}")
```



Fidesol

El mayordomo de la ciber

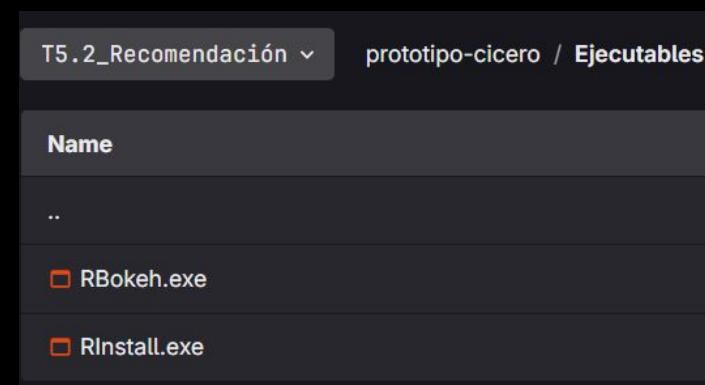
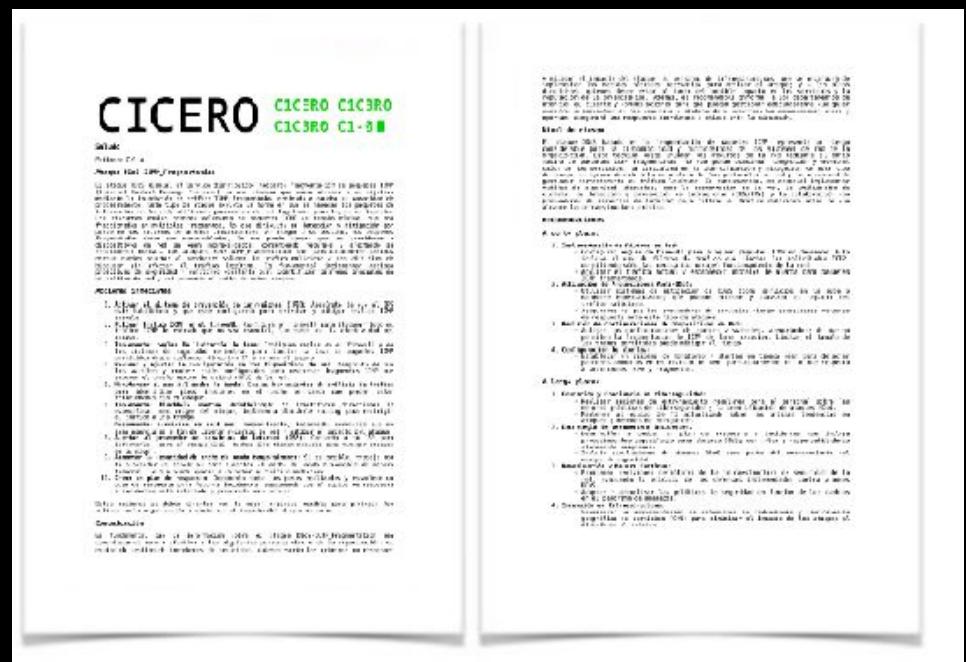
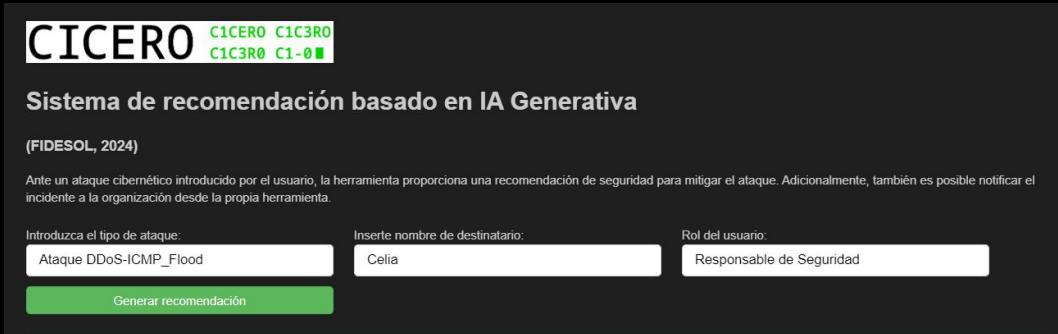
Objetivo Desarrollar un sistema de recomendación basado en IA generativa que permita mitigar ataques y amenazas en curso, generando recomendaciones de seguridad y respuestas automáticas , y utilizando la IA como elemento de mejora de soluciones de ciberseguridad existentes , avanzando hacia un enfoque inteligente y autónomo de detección y parcheo de vulnerabilidades .



Prototipo de
recomendación

Fidesol

Orquestación de seguridad (T5.2)







Cfernandez@fidesol.org

Muchas gracias

