



UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN  
Facultad de Ciencias Físico Matemáticas



# Proyecto Integrador de Aprendizaje

## Etapas 1: Desarrollo del Proyecto Api. OPEN THREAT EXCHANGE.

**Equipo 03:**

**Eduardo Vanoye Chi.**

**Yolanda del Carmen García Carranza.**

**Yair González Reyes.**

**Oliver Mauricio Alvarado García.**

**Alfredo Mendoza Hernandez**

# PLANTEAMIENTO\_DEL PROBLEMA\_

Como estudiantes universitarios, experimentamos una constante interacción con páginas y programas que apoyan las investigaciones que realizamos para comprender algunos de los temas que se ven en clase, de ahí surge la necesidad de proteger nuestros datos y la información manejada en estas páginas, para que podamos tener verificada la seguridad con la que contamos al trabajar en línea.

## DESCRIPCIÓN\_DE\_LA\_API ELEGIDA\_

La API seleccionada, ofrece una solución a la problemática presentada, esta lleva por nombre Open Threat Exchange.

OTX ofrece acceso abierto a una comunidad global de investigadores de amenazas y profesionales de la seguridad. Ofrece datos de amenazas generados por la comunidad, facilita la investigación colaborativa y automatiza el proceso de actualización de la infraestructura

# ALGORITMO

D.V:

LIMITE\_DE\_RESULTADOS: Entero.

INDICADOR[:]:Lista de tipo caracter.

LIMITE: Entero.

INDICADORES: Entero.

URL: Cadena de caracteres.

RESPONSE: Cadenas de caracteres.

DATOS: Cadena de caracteres.

1.- Inicio.

2.- Importar (API\_KEY)

3.- Leer (LÍMITE\_DE\_RESULTADOS)

4.- Intentar,

5.- Obtener (PULSO)

6.- INDICADORES [:LIMITE]

7.- Leer (INDICADOR)

8.- Para INDICADORES desde 0 hasta INDICADOR

9.- Mostrar "Tipo", "Dominio", "Indicador"

10.- sino,

11.- Mostrar "No hay indicadores disponibles para este pulso."

12.- Fin del condicional.

13.- INDICADORES +=1

14.- Fin del repetir.

15.- Intentar,

16.- Mostrar "Hubo un error al obtener indicadores"

17.- Leer (URL)

18.- Intentar,

19.- Leer (RESPONSE)

20.-. DATOS = RESPONSE.JSON()

21.- Excepto,

22.- Mostrar ("Error HTTP: " ERRH)

23.-Mostrar ("Error de conexión: " ERRC)

24.- Mostrar ("Tiempo de espera agotado: " ERRT)

25.- Mostrar ("Error desconocido en la petición: " ERR)

26.- Mostrar ("Error al decodificar JSON: " ERRJ")

27.- Fin del algoritmo.

# DESCRIPCIÓN\_DE\_LA ESTRUCTURA\_DE\_DATOS\_

La gestión de datos en este proyecto se basa principalmente en estructuras como listas y diccionarios. Al realizar una solicitud a la API de OTX, la respuesta se recibe en formato JSON, el cual es automáticamente interpretado en Python como una lista de diccionarios. Esto permite acceder de forma sencilla y ordenada a los diferentes elementos de los indicadores, facilitando su análisis y visualización dentro del programa.

# JUSTIFICACIÓN\_DEL\_ TRATAMIENTO\_DE\_DATOS APLICADO\_

La API de OTX facilita significativamente la gestión de datos gracias a que cuenta con una librería oficial diseñada específicamente para interactuar con su plataforma. Esto permite acceder, consultar y manipular información de amenazas de forma más directa y estructurada, sin tener que construir manualmente las solicitudes HTTP. A diferencia de otras APIs más generales, OTX proporciona herramientas optimizadas que simplifican el trabajo con los datos, haciéndolo más eficiente y accesible para desarrolladores y analistas de ciberseguridad.

# MINUTAS\_DE\_TRABAJO\_

15 de abril del 2025.

Equipo: 03.

Miembros del equipo:

- Eduardo Vanoye Chi.
- Yair González Reyes.
- Yolanda del Carmen García Carranza.
- Oliver Mauricio Alvarado García.
- Alfredo Mendoza Hernández.

Objetivos:

- Comenzar la organización del trabajo.
- Decidir que API se utilizará en el desarrollo de este proyecto.
- Indicar los roles que tomará cada integrante.

Dificultades:

- Diferencia notoria en los horarios para las juntas de trabajo del proyecto.
- Falta de comunicación.

Próximos pasos:

- Iniciar el código.
- Definir los puntos importantes a seguir.

23 de abril del 2025.

Equipo: 03.

Miembros del equipo:

- Eduardo Vanoye Chi.
- Yair González Reyes.
- Yolanda del Carmen García Carranza.
- Oliver Mauricio Alvarado García.
- Alfredo Mendoza Hernández.

Objetivos:

- Iniciar con el código.

- Iniciar con el código.
- Acordar bien las funciones necesarias para hacer uso de la API.
- Separar por pasos la investigación a realizar.

Dificultades:

- Falta de organización.
- Desconocimiento del uso de la API.

Próximos pasos:

- Acabar la primera parte del código.
- Designar los roles para documentación e investigación.

07 de mayo del 2025.

Equipo: 03.

Miembros del equipo:

- Eduardo Vanoye Chi.
- Yair González Reyes.
- Yolanda del Carmen García Carranza.
- Oliver Mauricio Alvarado García.
- Alfredo Mendoza Hernández.

Objetivos:

- Investigar los diferentes programas maliciosos de los que podemos implementar información que el programa pueda detectar.
- Tratar la lógica del código, para poder realizar las gráficas pertinentes.

Dificultades:

- Desconocimiento de las páginas a utilizar.
- Se detectó cierta falta de comunicación.

Próximos pasos:

- Concluir la investigación.
- Finalizar la primera parte de la documentación

12 de mayo del 2025.

Equipo: 03.

Miembros del equipo:

- Eduardo Vanoye Chi.
- Yair González Reyes.
- Yolanda del Carmen García Carranza.
- Oliver Mauricio Alvarado García.
- Alfredo Mendoza Hernández.

Objetivos:

- Investigar paquetes de información en la web que contengan dominios maliciosos relacionados con google.
- Definir los horarios para las grabaciones de la primera y la segunda fase.

Dificultades:

- Desconocimiento de las páginas a utilizar.
- Se detectó cierta falta de comunicación.

Próximos pasos:

- Concluir la investigación.
- Grabar los videos.
- Finalizar la primera parte de la documentación.