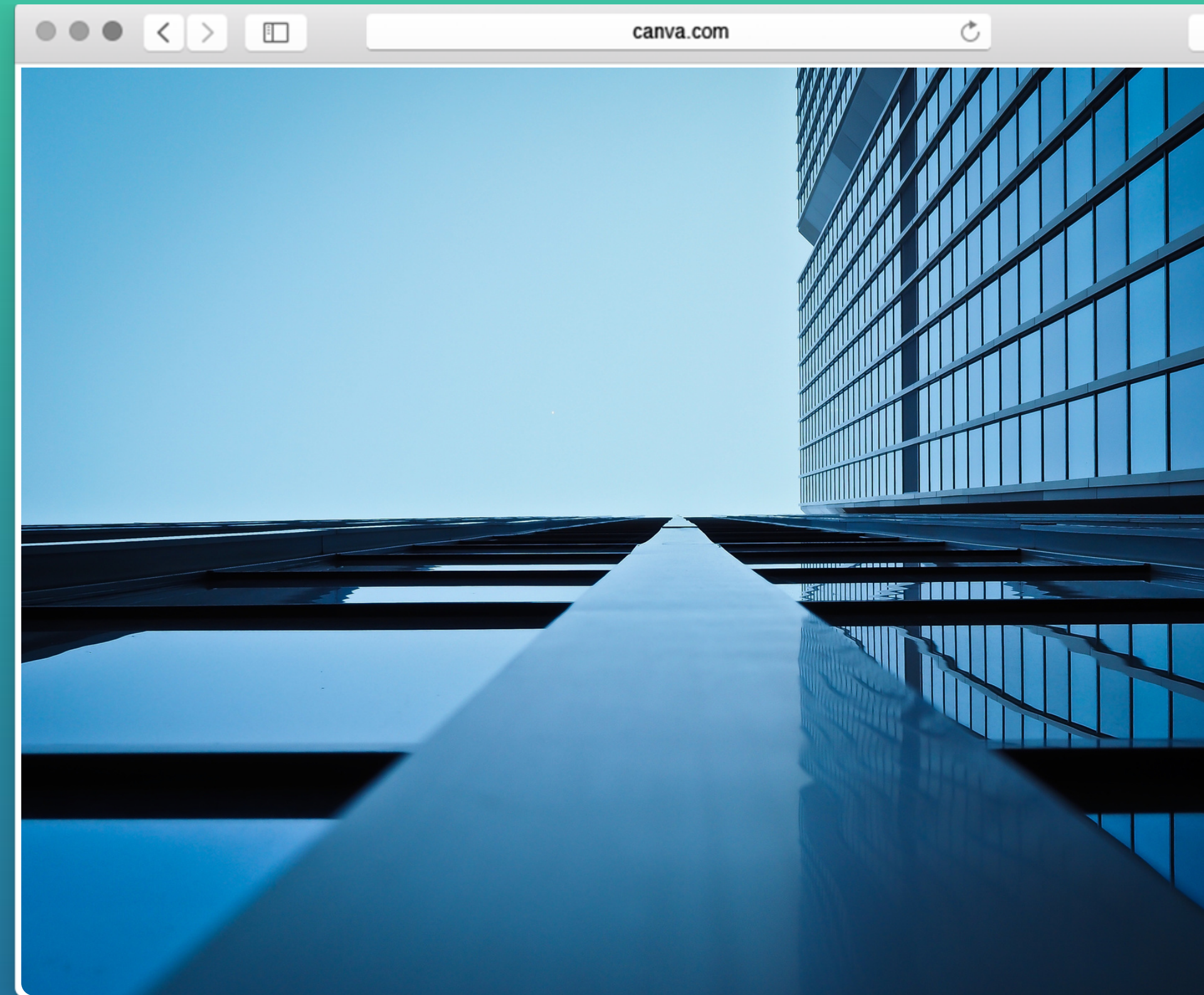
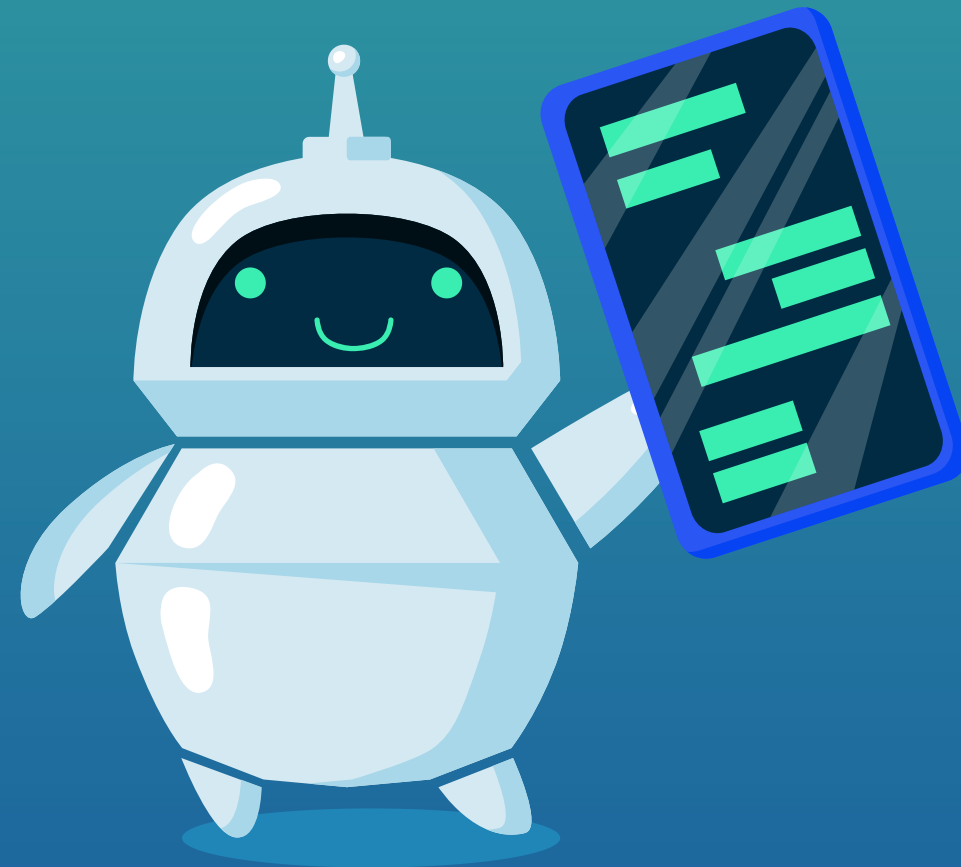


AI PLAGIARISM

Carlos
Raul
Angela
Andreu
Georgina



Como se hizo

01

Detección de plagio con el uso de IA.

02

Investigar la posibilidad que un sistema de IA detecte que un texto no sea original.

03

Casos contemplados:

- Contenido copiado de internet.
- Contenido creado por una IA.
- Contenido copiado de otra fuente con copyright.

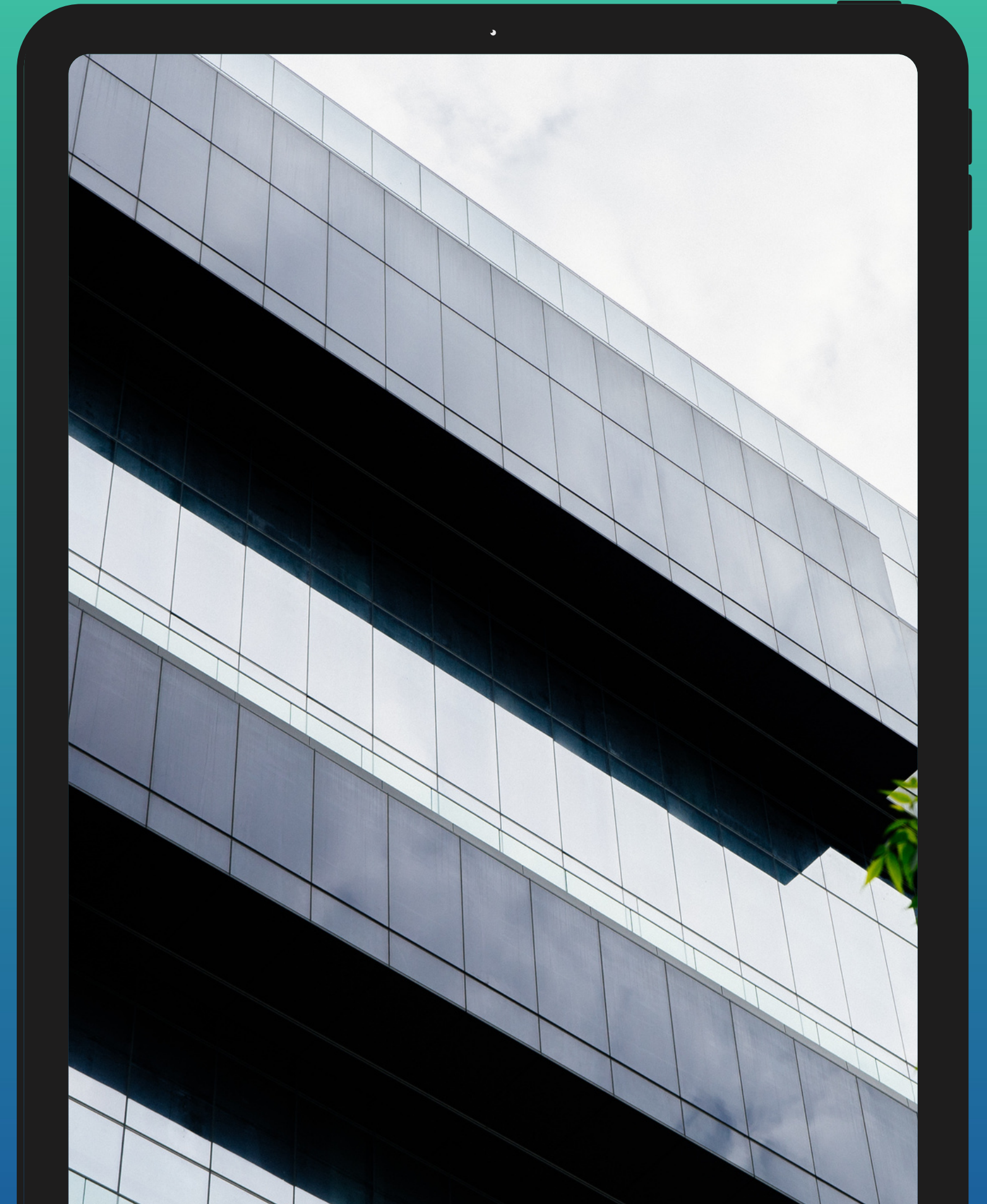
Contenido

- 01 Conexión con la API de OpenIA.
- 02 Uso de la API de OpenIA.
- 03 Conexión con la API de Copyleaks.
- 04 Uso de la API de Copyleaks.
- 05 Conclusiones

Para esta investigación
usamos las API gratuitas
que encontramos:

- [OpenAI \(ChatGPT\)](#)
 - [Copyleaks](#)
-

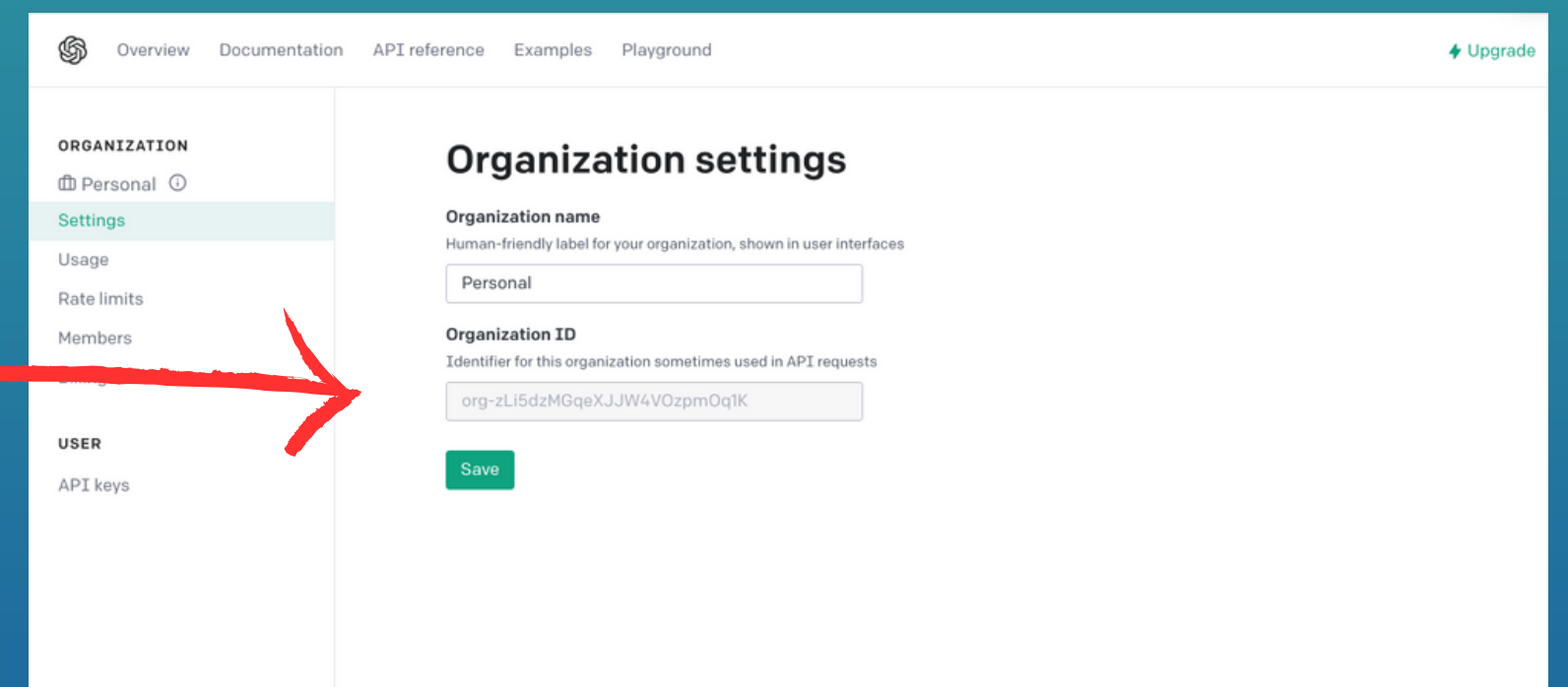
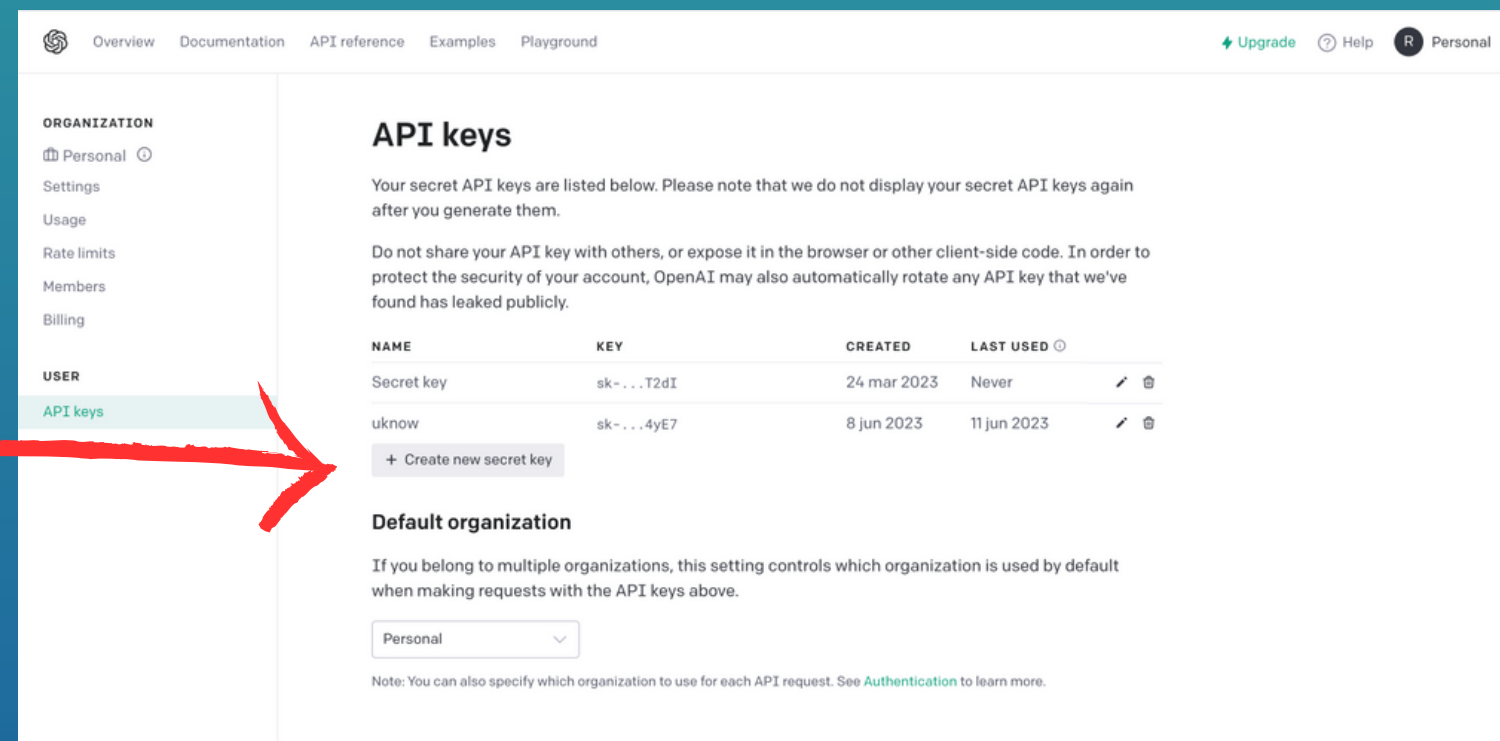
Encontramos otras
herramientas interesantes
como [ZeroGPT](#) para la
detección de creaciones de AI,
pero no tiene un servicio
gratuito, así que para nuestro
uso lo descartamos.



Conexión a Open AI API

<https://openai.com/>

Requiere el registro y la creación de una KEY personal como password, y el uso del ID de la organización como username, y así identificarse a la hora de hacer peticiones a la API.



Uso de Open AI API

Usamos un archivo .env para guardar las variables con los datos y así hacer la conexión y lo usamos para hacer la petición usando las siguientes propiedades:

- **model:** text-davinci-003
- **prompt:** El texto que le queremos enviar al modelo
- **max-tokens:** La cantidad de texto que quieres recibir como respuesta.
- **temperature:** La coherencia que queremos mantener en la respuesta. Con el 0 especificamos que ante la misma pregunta nos de la misma respuesta.

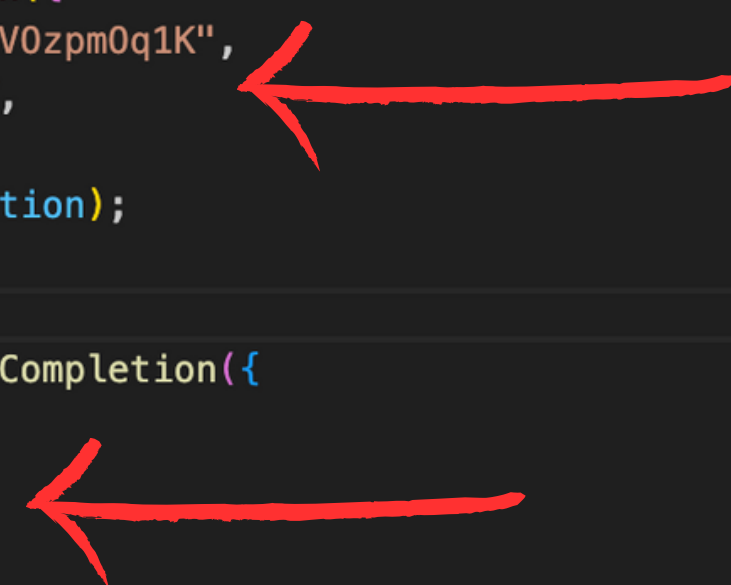
```
import dotenv from "dotenv";
import { Configuration, OpenAIApi } from "openai";
import { prompt } from "../data";

dotenv.config();

const configuration = new Configuration({
  organization: "org-zLi5dzMGqeXJJW4V0zpm0q1K",
  apiKey: process.env.OPENAI_API_KEY,
});
const openai = new OpenAIApi(configuration);

const checkOriginality = async () => {
  const result = await openai.createCompletion({
    model: "text-davinci-003",
    prompt: prompt,
    max_tokens: 1000,
    temperature: 0,
  });
  const data = result.data;
  console.log(data.choices[0].text);
}

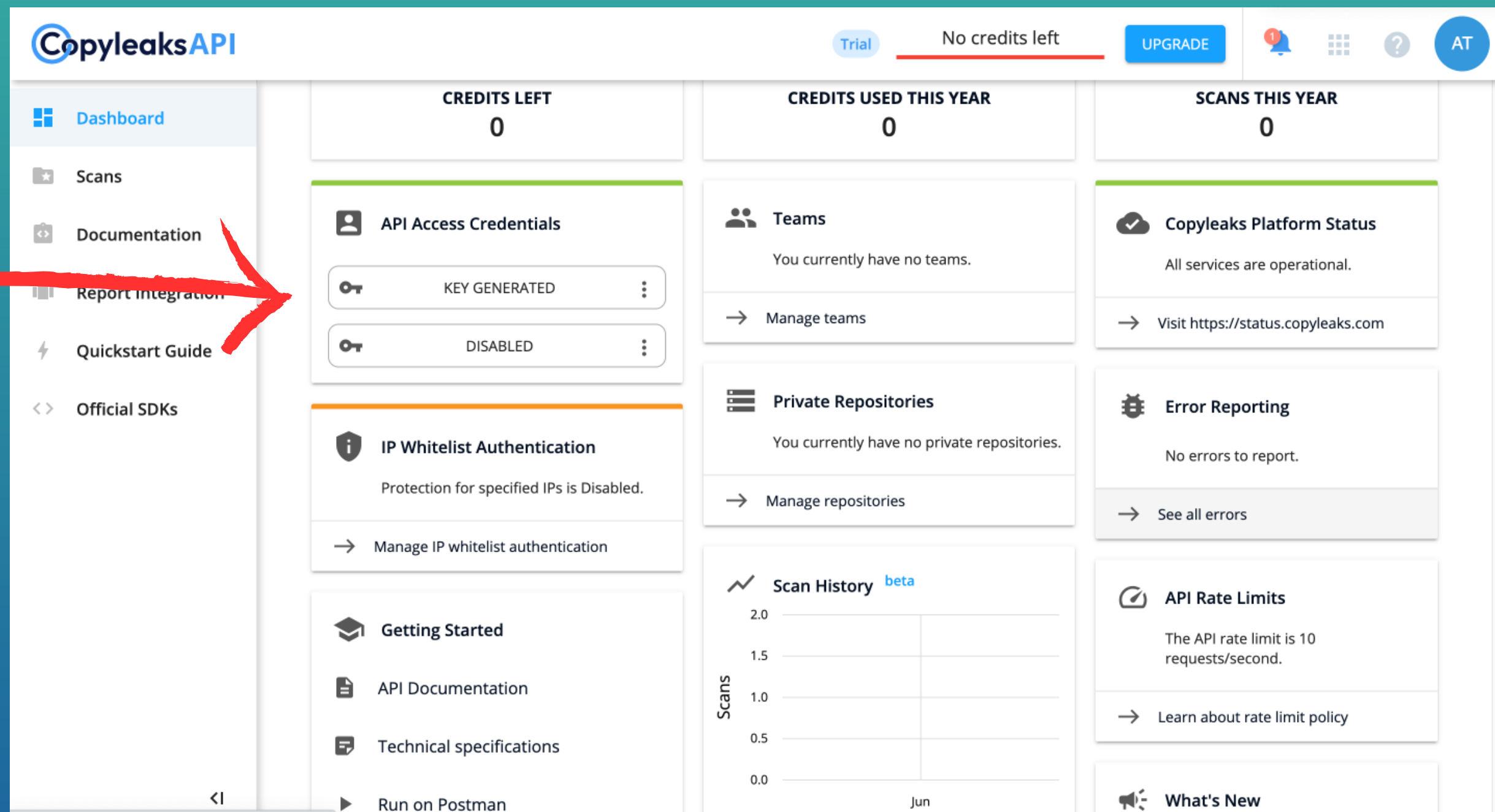
checkOriginality();
```



Conexión Copyleaks API

<https://copyleaks.com/>

Al igual que con OpenAI, hay que crear una KEY para la conexión:



The screenshot displays the Copyleaks API dashboard. The top navigation bar includes the CopyleaksAPI logo, a 'Trial' badge, a 'No credits left' warning, an 'UPGRADE' button, and user profile information (AT). The left sidebar contains links to Dashboard, Scans, Documentation, Report Integration, Quickstart Guide, and Official SDKs. A large red arrow points from the 'Documentation' link to the 'API Access Credentials' section in the main content area. This section shows two API keys: one 'KEY GENERATED' and one 'DISABLED'. Other sections include 'Teams' (no teams), 'Private Repositories' (no repositories), 'Copyleaks Platform Status' (operational), 'Error Reporting' (no errors), 'API Rate Limits' (10 requests/second), and 'Scan History' (a line graph showing zero scans for June).

CopyleaksAPI Trial No credits left UPGRADE AT

Dashboard

- Scans
- Documentation
- Report Integration
- Quickstart Guide
- Official SDKs

CREDITS LEFT
0

CREDITS USED THIS YEAR
0

SCANS THIS YEAR
0

API Access Credentials

- KEY GENERATED
- DISABLED

Teams
You currently have no teams.
→ Manage teams

Private Repositories
You currently have no private repositories.
→ Manage repositories

Copyleaks Platform Status
All services are operational.
→ Visit <https://status.copyleaks.com>

Error Reporting
No errors to report.
→ See all errors

API Rate Limits
The API rate limit is 10 requests/second.
→ Learn about rate limit policy

Scan History beta

Scans

2.0
1.5
1.0
0.5
0.0

Jun

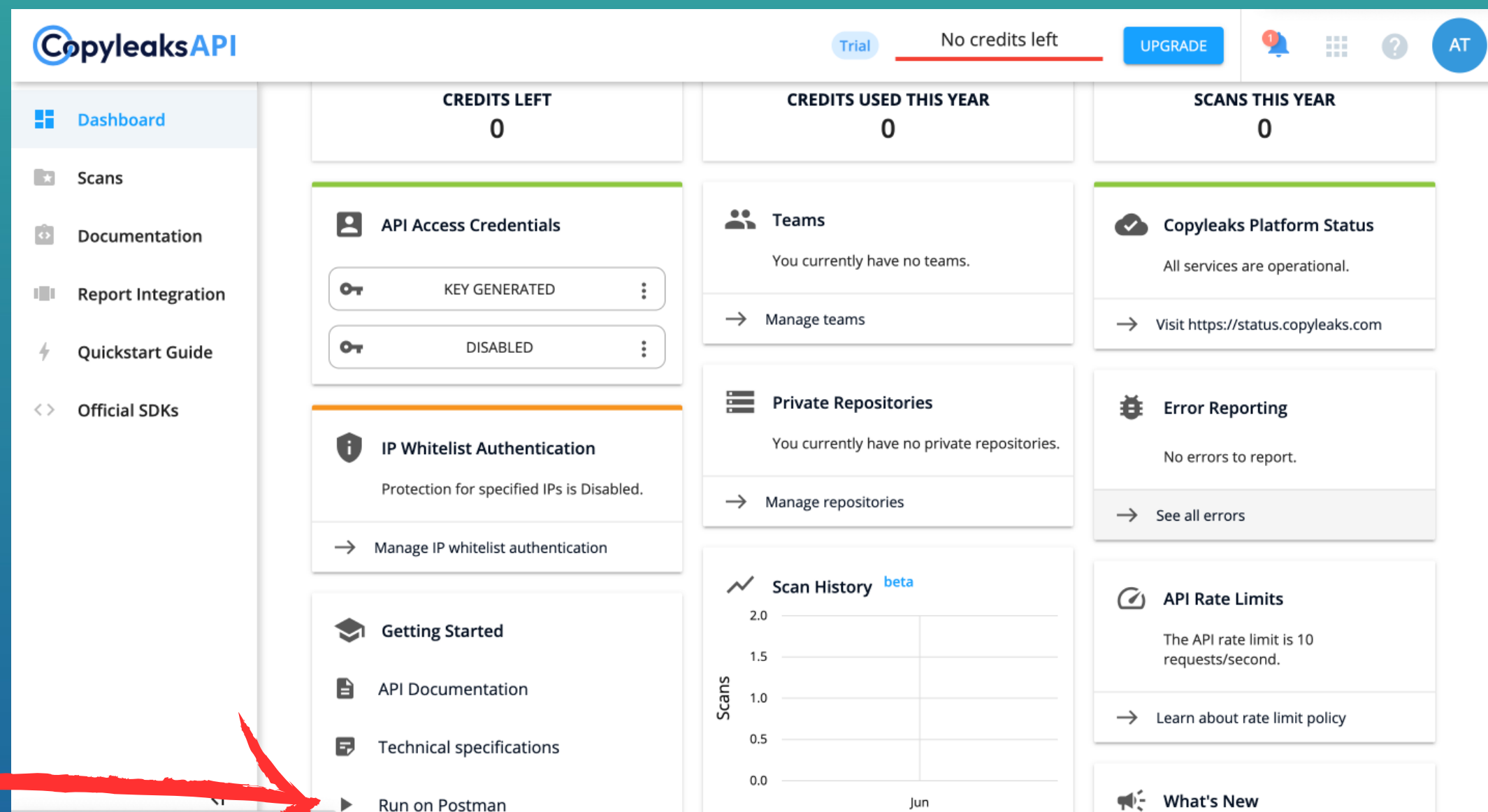
Getting Started

- API Documentation
- Technical specifications
- Run on Postman

What's New

Conexión Copyleaks API

Debido a que es algo mas complejo el uso de la API usamos POSTMAN. Nos facilitan ya las consultas que se importan directamente y sólo hay que especificar las opciones.

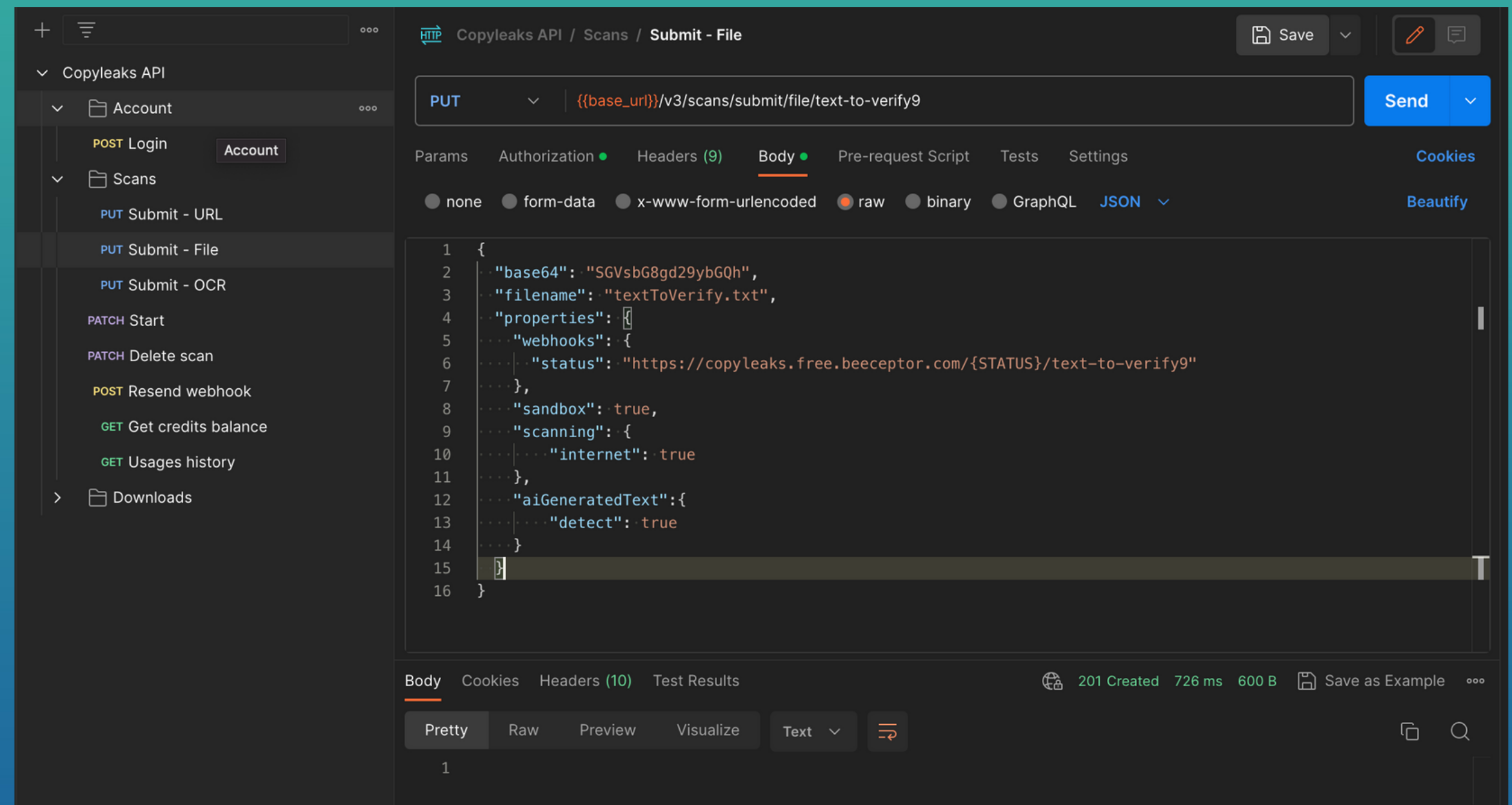


Uso Copyleaks API

Esta API nos da la opción de enviar diferentes fuentes para comprobar su originalidad:

- Archivo
- URL
- OCR (imagen)

Nosotros usamos la función de file, donde incorporamos diferentes textos a verificar su originalidad.



Uso Copyleaks API

Webhooks

Es necesario usar un webhook, que es un interceptor de la respuesta de la API que nos muestra el resultado devuelto por la consulta:

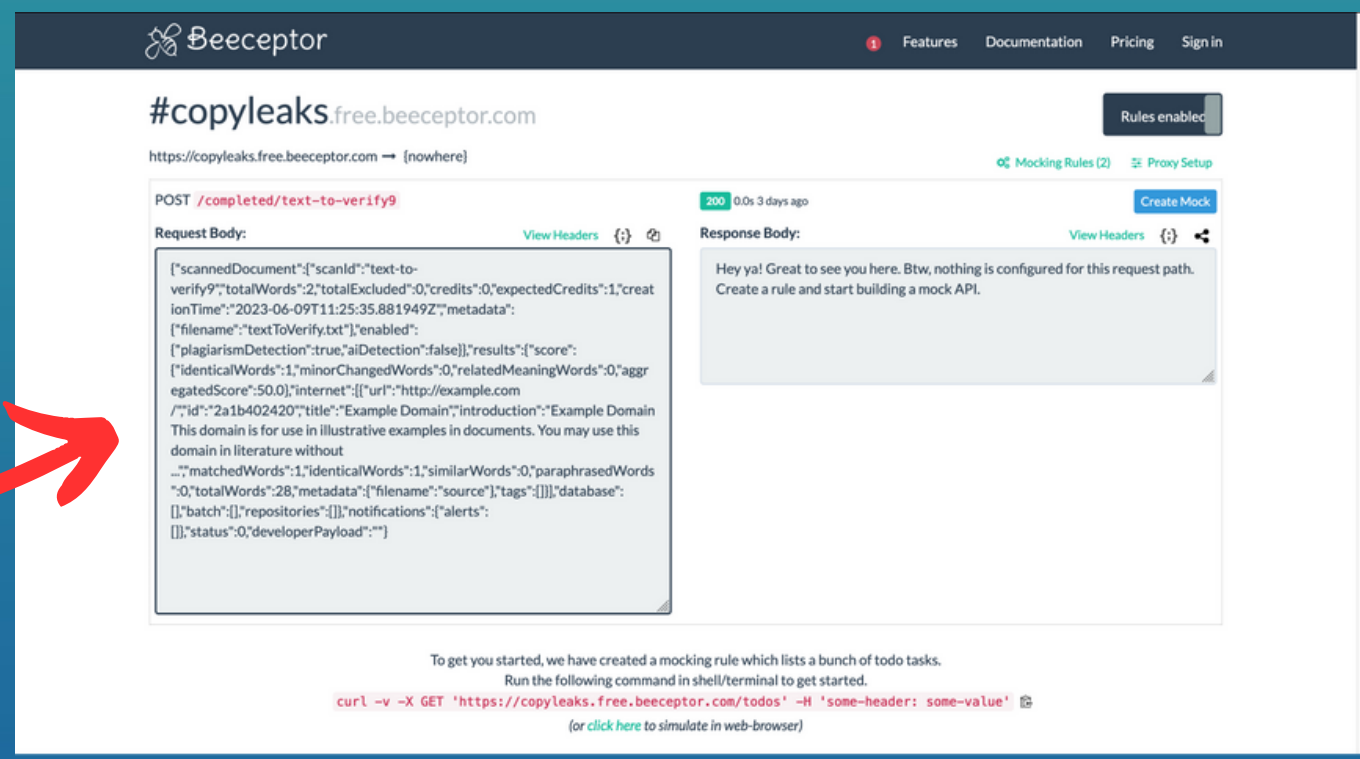
Beeceptor

<https://beeceptor.com/>

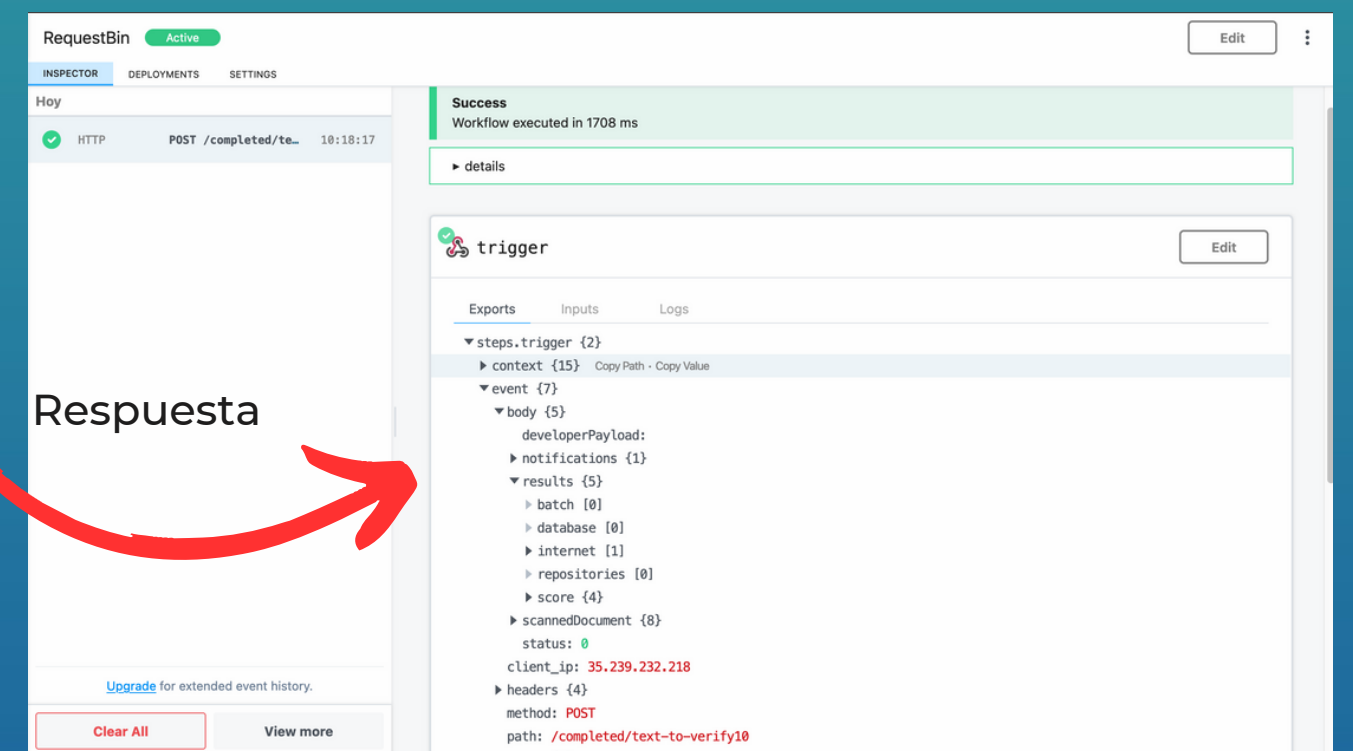
Requestbin

<https://pipedream.com/>

Respuesta



Respuesta



Conclusiones

OPENAI API

En el caso del uso de la API de OpenAI, no hemos conseguido un prompt, un texto suficientemente confiable para que nos devuelva un resultado coherente con lo que le pedimos. Esto junto con la falta de capacidad de los modelos usados para detectar si el texto está plagado o no, la conclusión es que no es viable el uso para la detección de plagio.

COPYLEAKS API

Debido a la complejidad del uso de la API no conseguimos obtener datos que pudiésemos cuantificar y que nos sirvan para determinar si es un modelo valido para la detección de plagio.

NOTA: Hay que tener en cuenta que hemos usado la versión gratuita de la API de OpenAI y el Sandbox en el caso de Copyleaks.

Gracias por escuchar!

