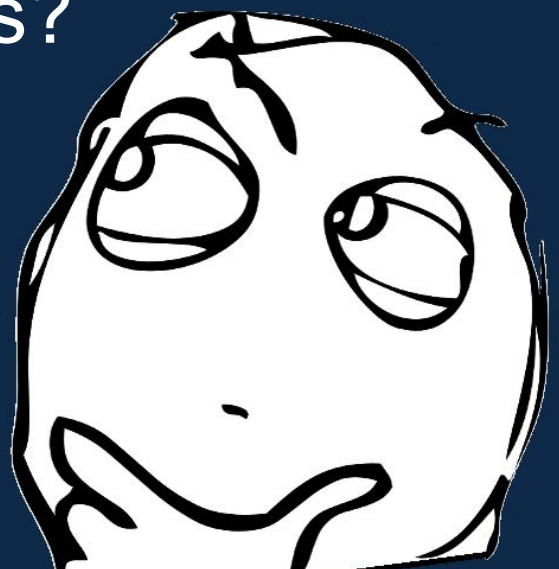
Artes Marciales con JavaScript

SERGIO GARZÓN

- Desarrollador de Software
- Desarrollador de Videojuegos
- Docente de Programación

Automatización de pruebas de API

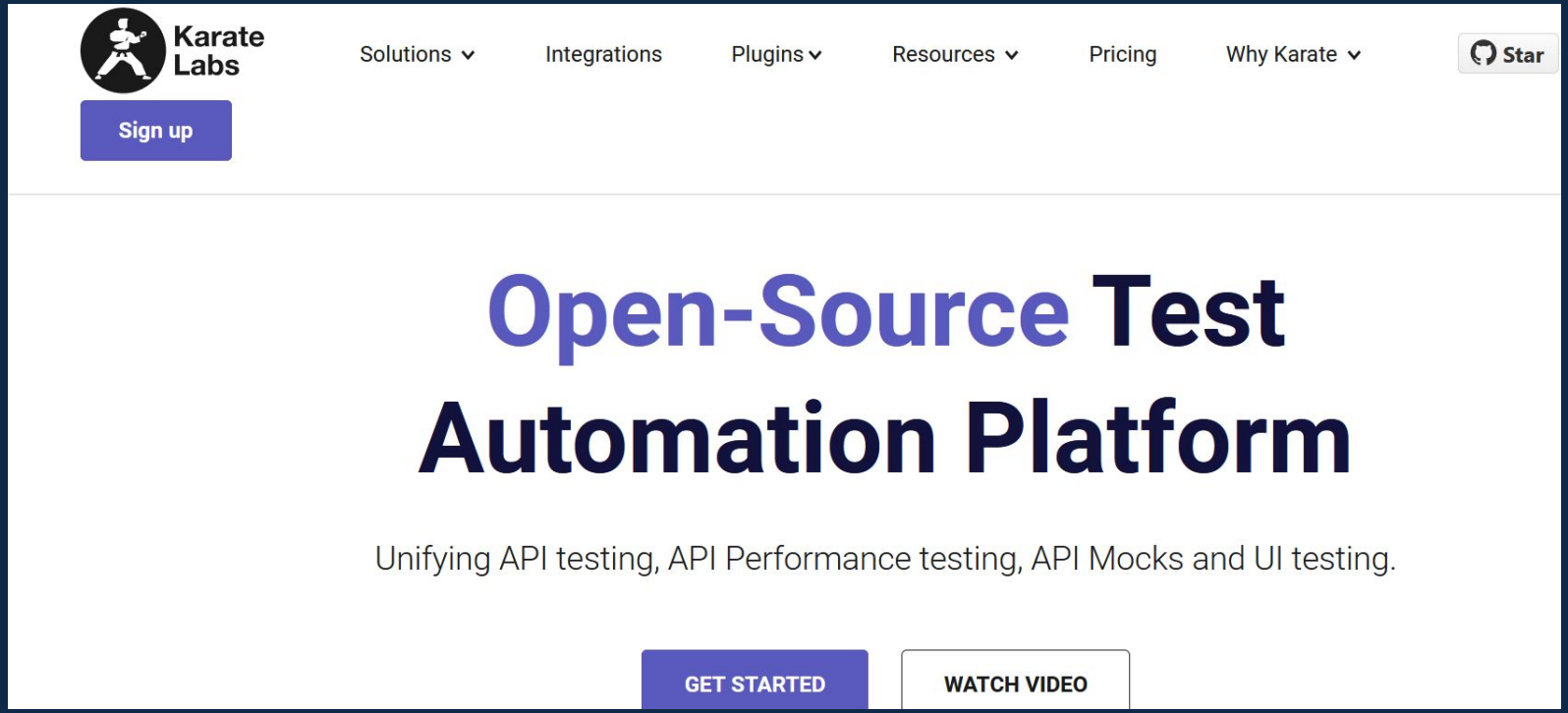
¿Por qué el título de la charla es
“Artes Marciales con JavaScript”
si me estas mostrando una
herramienta para testear APIs?
No entiendo…



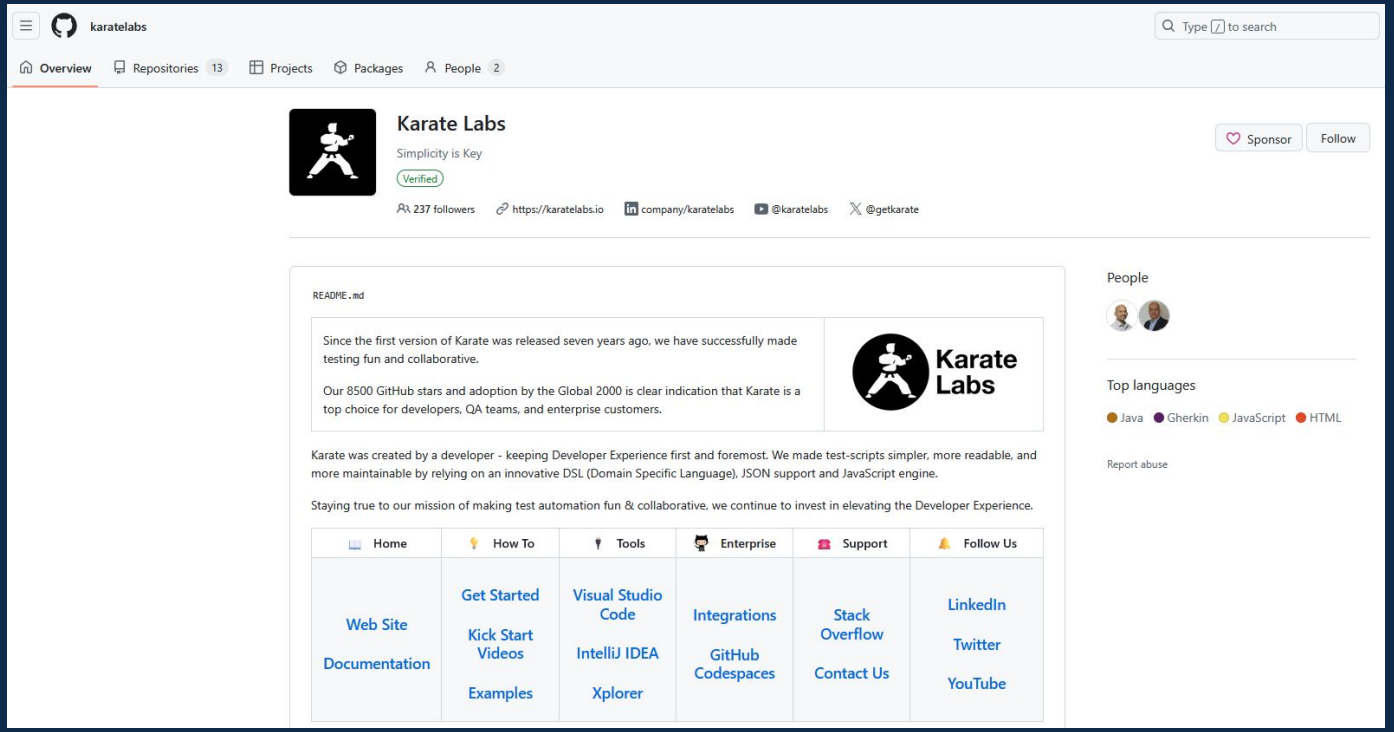
Karate Framework / Karate DSL

Karate Labs

Empresa de software
Sugar Land, Texas



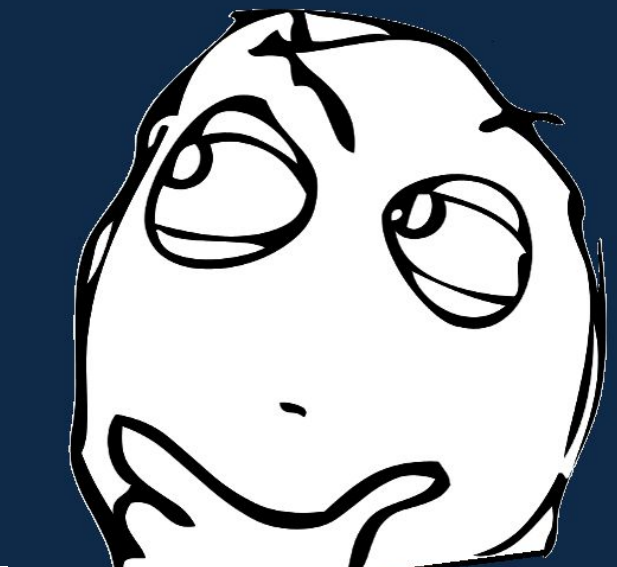
Sitio web: https://www.karatelabs.io/



Sitio web: https://github.com/karatelabs

👍 Interesante….

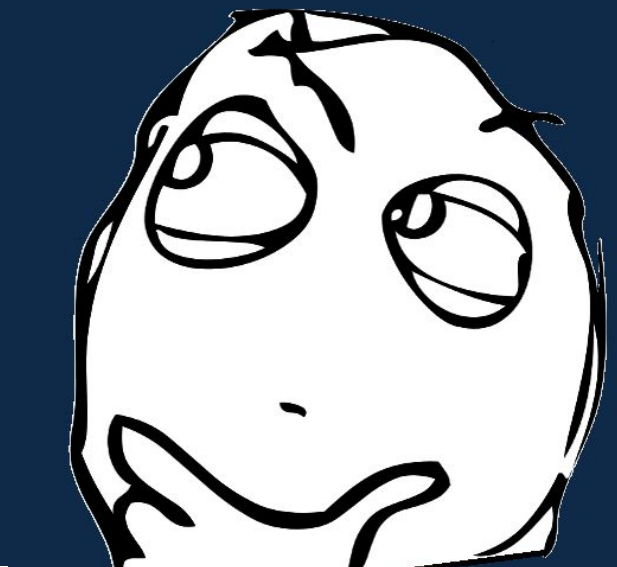
Pero ¿Para que me sirve? 🤷



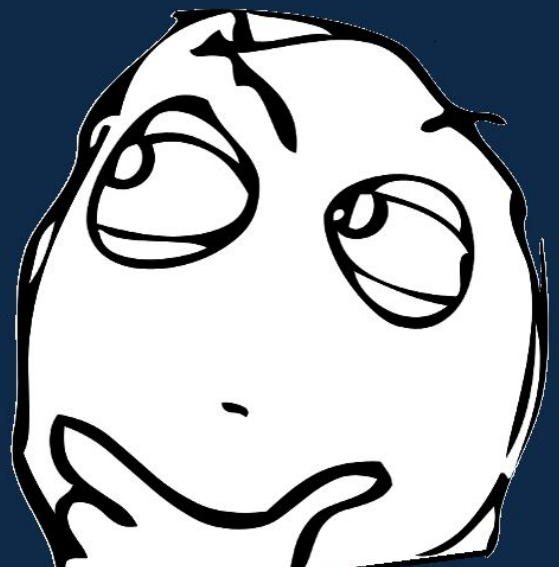
KARATE sirve para muchas cosas:

- Nos permite automatizar pruebas de APIs
- Crear script pruebas de APIs
- Simular APIs
- Realizar pruebas de UI
- Automatizar pruebas web, de performance y demás.

¿Y es fácil? 😱



Obvio, no es difícil, es fácil aprenderlo, pero puede volverse complejo, depende del escenario.
😊



KARATE utiliza:

- Utiliza Cucumber (Software para procesar y ejecutar pruebas de APIs)

- Cucumber utiliza el lenguaje Gherkin (Lenguaje DSL - Lenguaje de dominio específico)

- Cucumber se centra en la sintaxis BDD (Behavior Driven Development / Desarrollo dirigidos por comportamiento)

KARATE utiliza:

- Está escrito en el lenguaje de programación de Java

- Lo podemos utilizar en cualquier proyecto, no necesariamente de Java

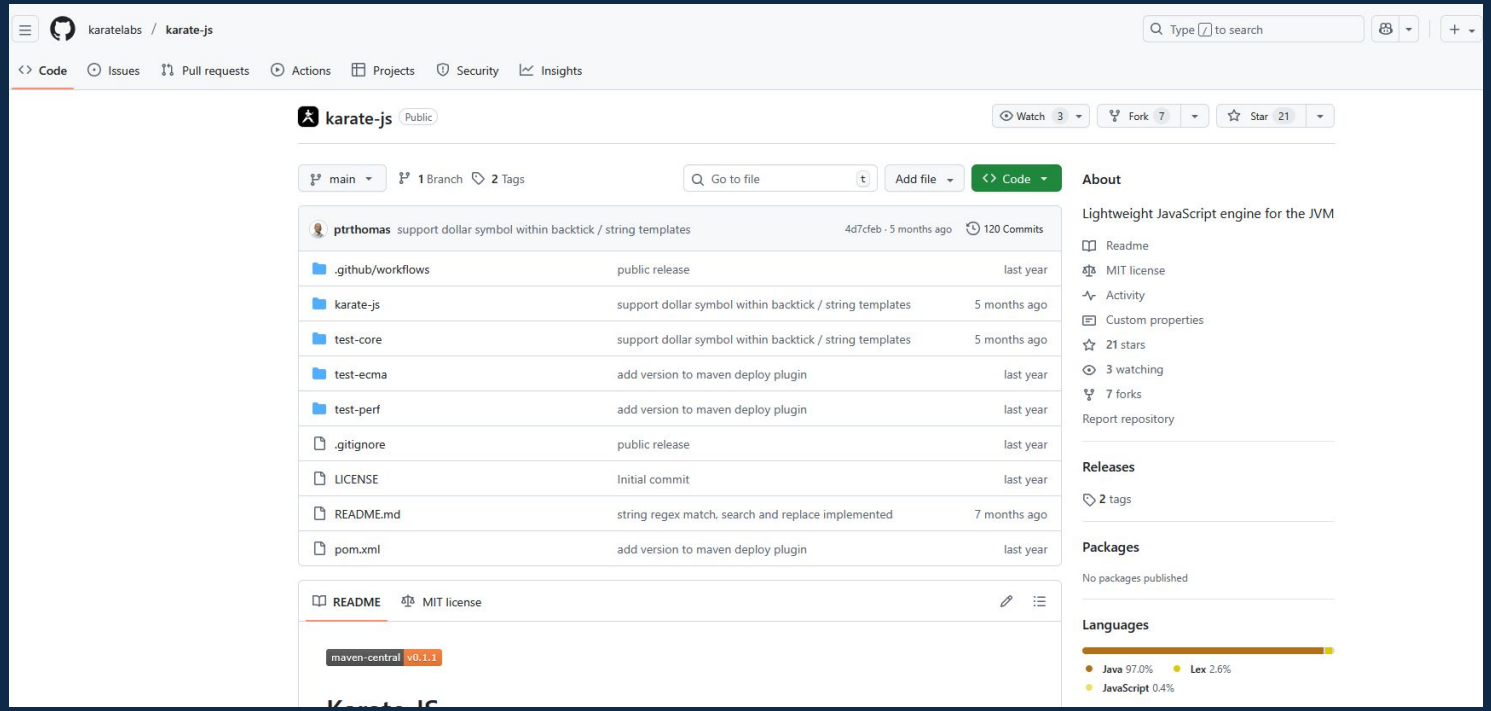
- Es de código abierto, es gratuito, pero en su mayoría

Y lo más importante de todo:

- Karate tiene funciones internas, escritas en JavaScript, porque cuenta con el motor Karate-JS, que está basado en el motor GraalVM.

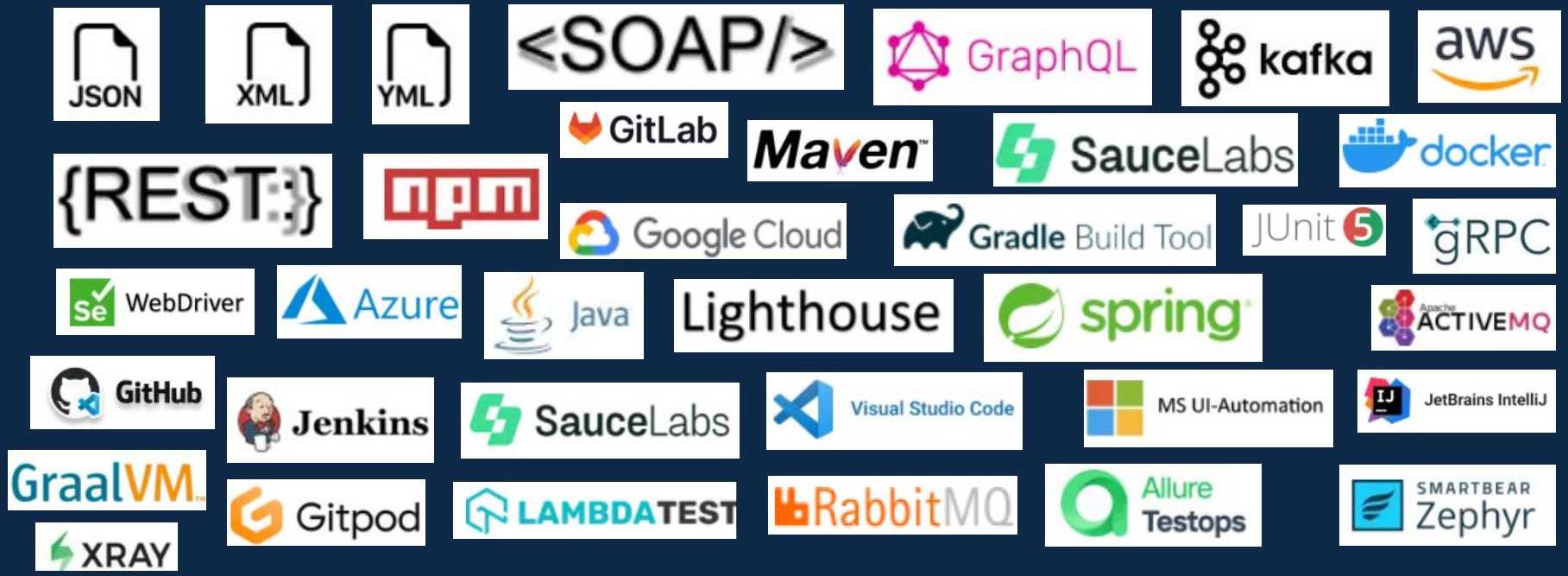
- Se pueden crear más funciones de JavaScript para realizar configuraciones. Dentro del archivo .feature o en un archivo .js

Repo del motor Karate-JS



Sitio web: https://github.com/karatelabs/karate-js

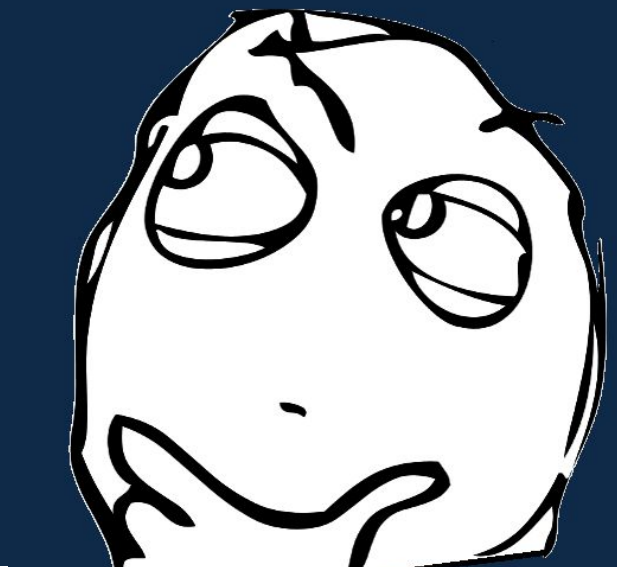
Se puede integrar con varias tecnologías:



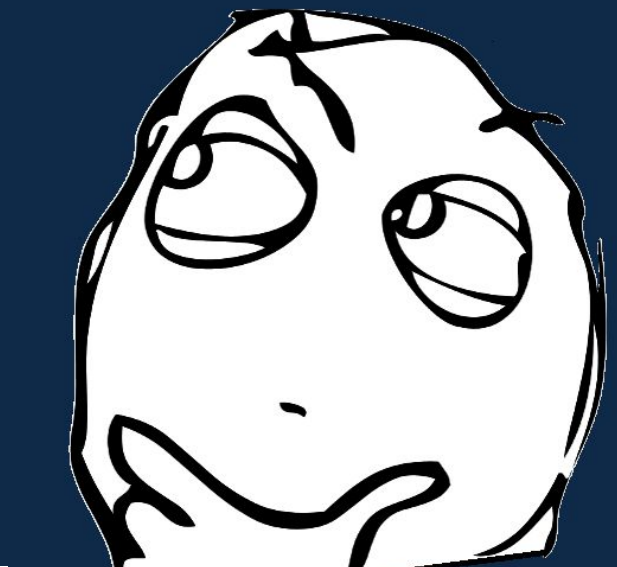
Sitio web: https://www.karatelabs.io/integrations

Karate está escrito en Java y está relacionado con la Java Virtual Machine

🤔 Interesante todo lo que me contas ¿Y tenes alguna demostración para ver?



Si, mas vale papa. 😊



Si tenemos un proyecto de Java con Maven o Gradle es posible, de hecho mucho mejor porque Karate DSL está escrito en java

Teniendo el JDK instalado:

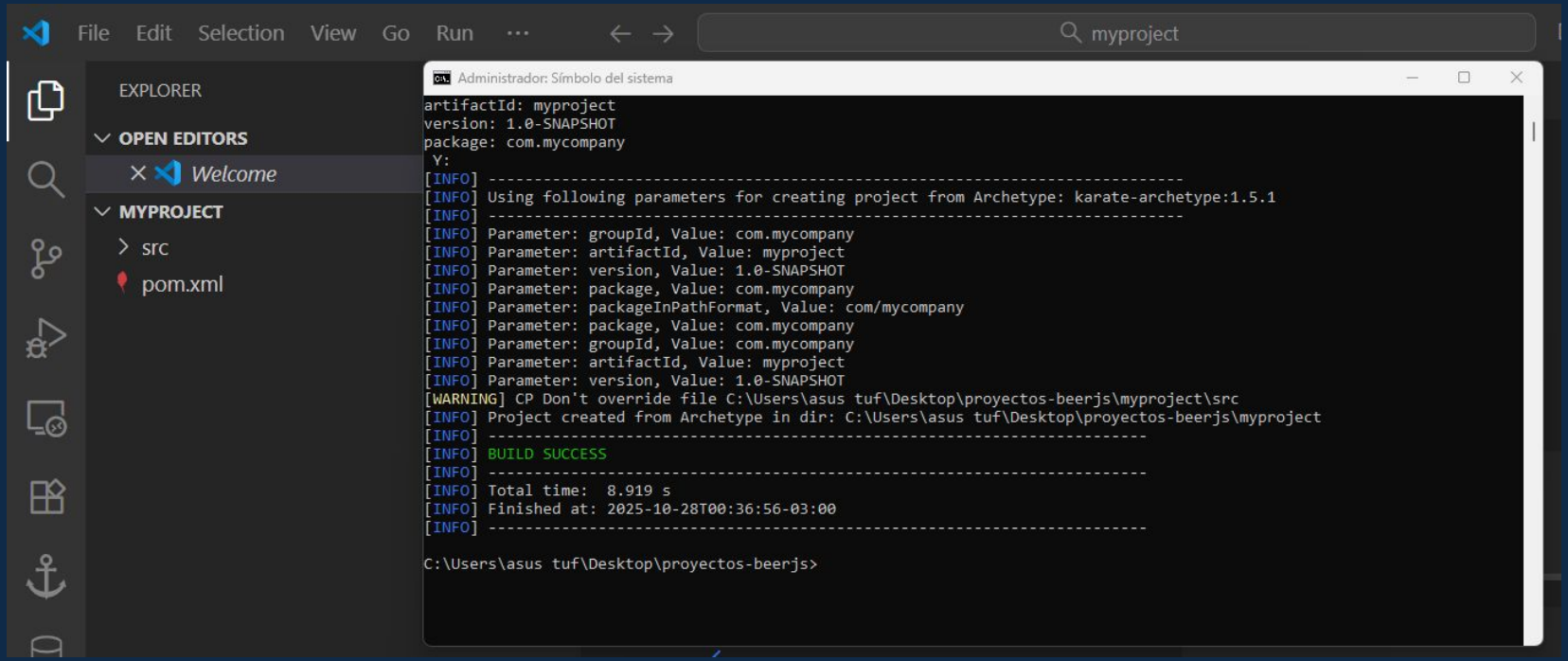
En el repo nos da el comando para generar un proyecto de Maven

```
mvn archetype:generate \
-DarchetypeGroupId=io.karatelabs \
-DarchetypeArtifactId=karate-archetype \
-DarchetypeVersion=1.5.1 \
-DgroupId=com.mycompany \
-DartifactId=myproject
```

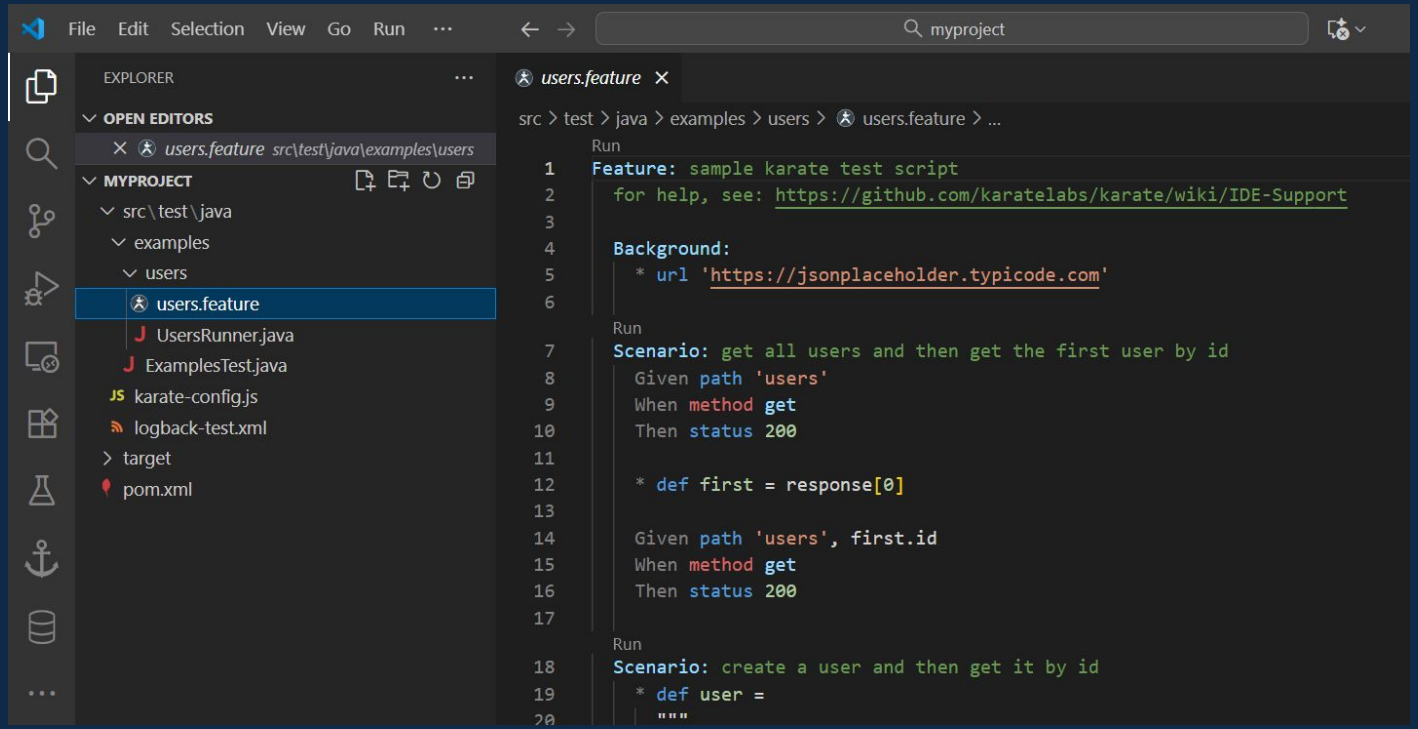
También con Gradle:

```
testImplementation 'io.karatelabs:karate-junit5:1.5.1'
```

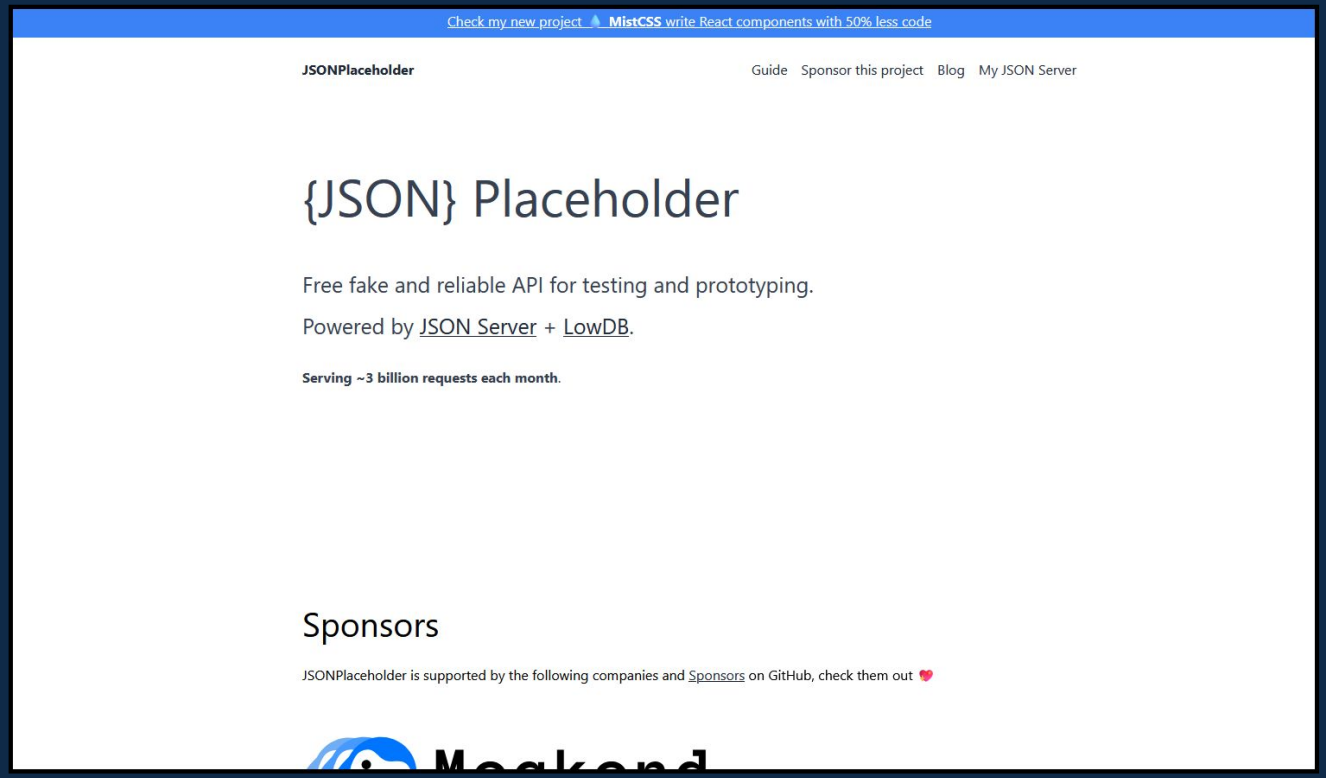
Creamos el proyecto de java con Maven



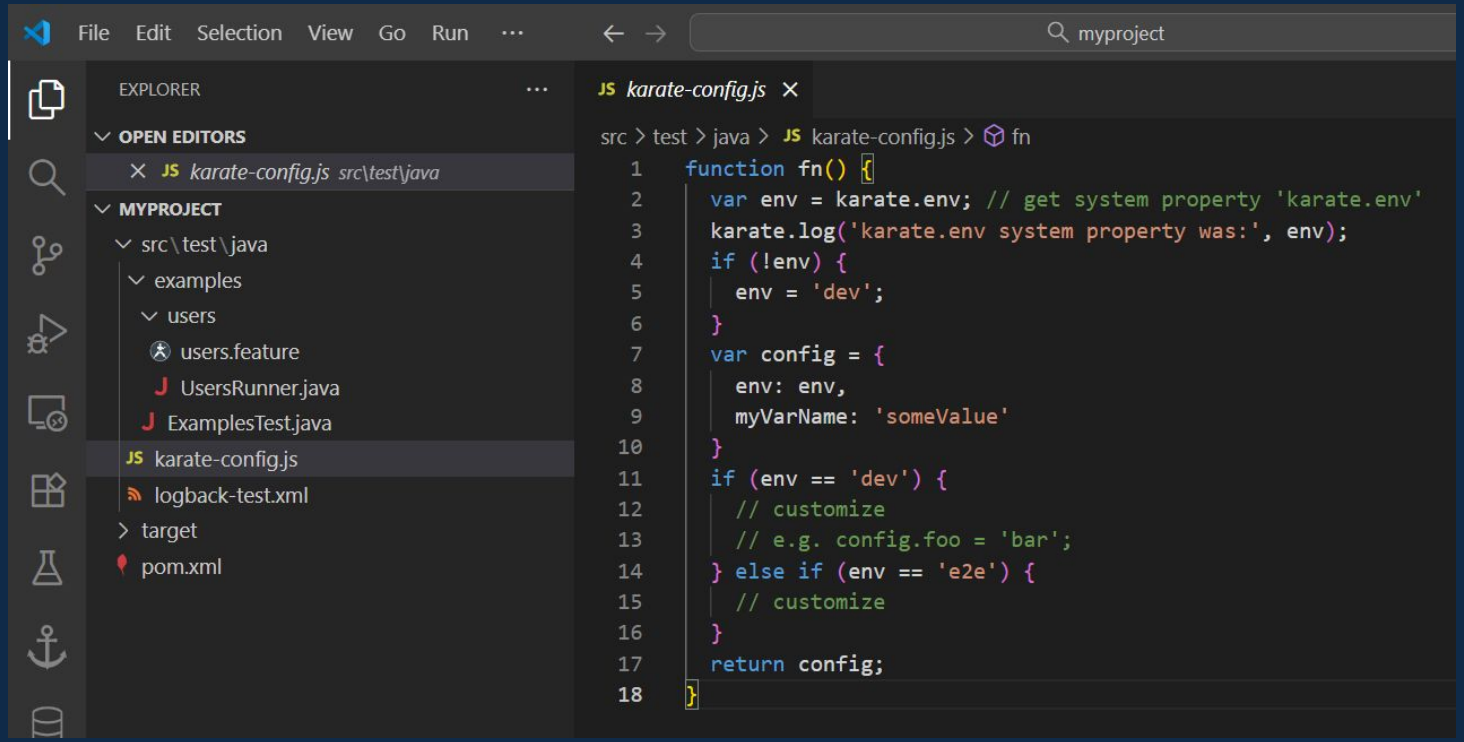
En la estructura del proyecto tenemos los escenarios (archivos .feature)



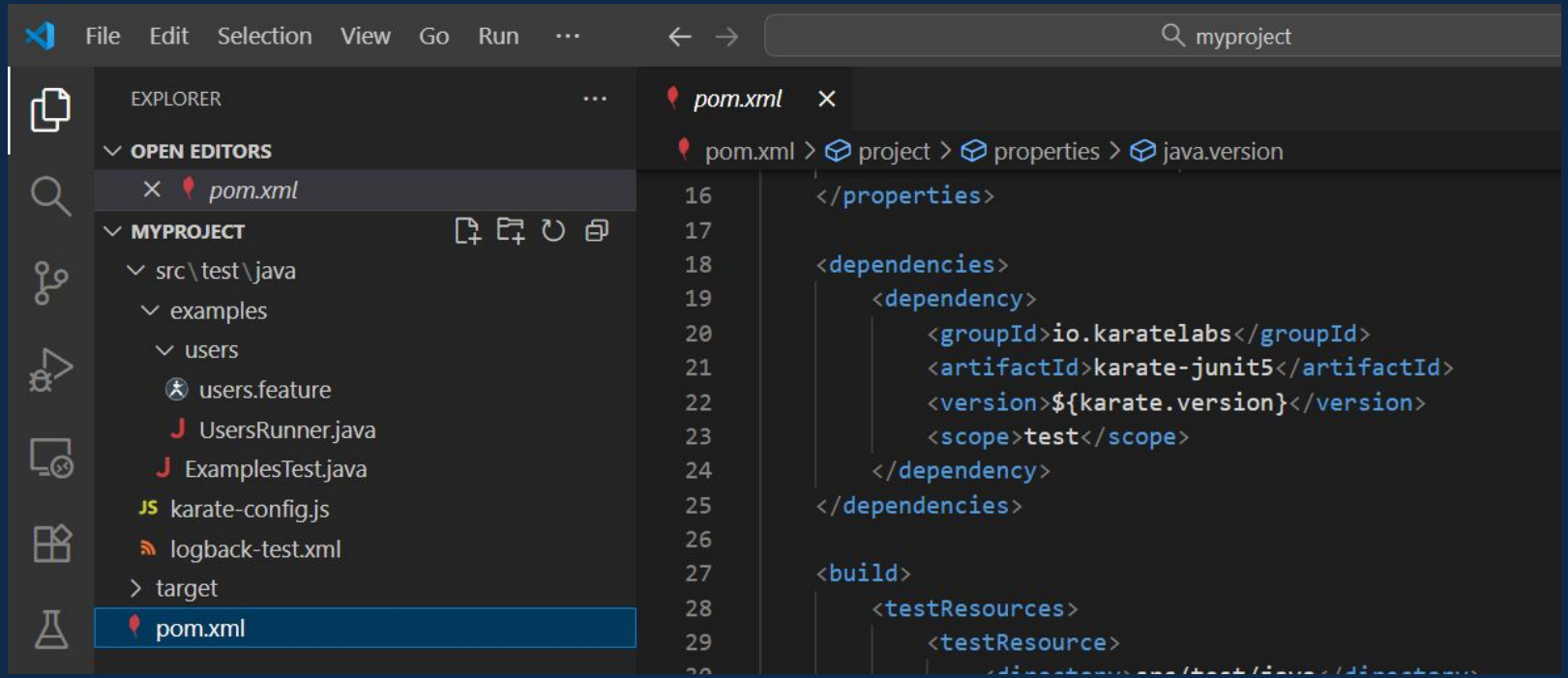
Ya nos trae por defecto una url de una API (users) de JSONPlaceHolder



y el archivo de configuración en JavaScript que usa Karate (archivo karate-config.js)



En el archivo pom.xml podemos ver que Karate Framework / DSL se integra con JUnit para hacer test de pruebas



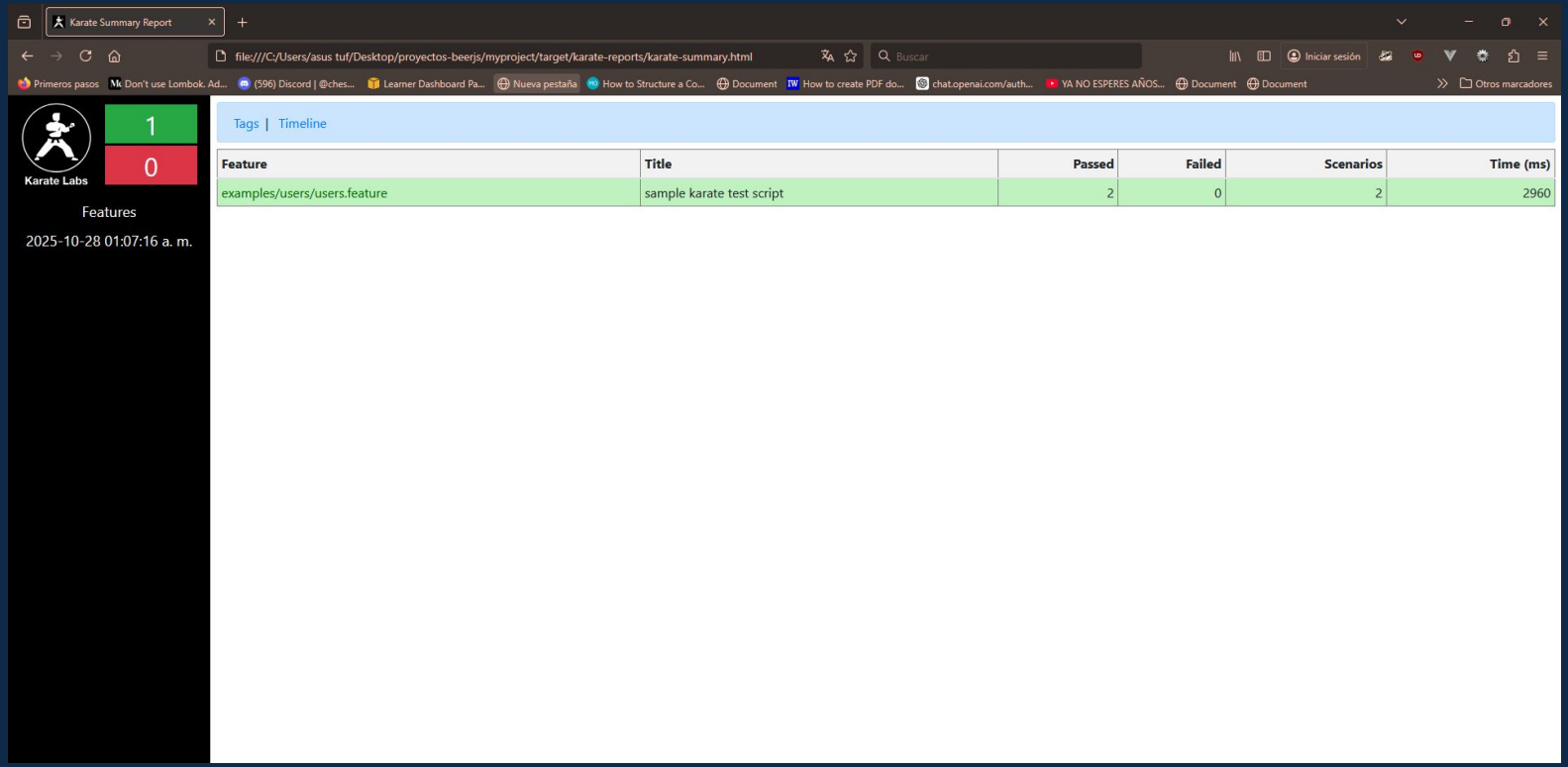
Para instalar las dependencias de Karate, junto con las otras, entonces ejecuto el comando de Maven

```
PS C:\Users\asus tuf\Desktop\proyectos-beerjs\myproject> mvn compile
[INFO] Scanning for projects...
[INFO]
[INFO] ----------------------< com.mycompany:myproject >----------------------
[INFO] Building myproject 1.0-SNAPSHOT
```

Luego ejecuto el test con Maven

```
PS C:\Users\asus tuf\Desktop\proyectos-beerjs\myproject> mvn clean test
[INFO] Scanning for projects...
[INFO]
[INFO] ----------------------< com.mycompany:myproject >----------------------
[INFO] Building myproject 1.0-SNAPSHOT
[INFO]   from pom.xml
```

En la terminal se genera el log, y a su vez nos da el reporte de las pruebas que hizo



Dentro de los archivos .feature de Karate, tenemos sintaxis de Cucumber

```
Run
Feature: Todo lo que contiene el test

Background: Precondiciones (url, token, variables de cada escenario y demas...)
Run
Scenario: Nombre del escenario

    Given
    #Dado
    And
    # Y
    When
    # Cuando
    Then
    # Entonces

Run
Scenario: Nombre del escenario 2

    # Multiples escenarios
```

Dentro de los archivos .feature de Karate, es posible escribir JSON dentro de una variable:

```
Scenario: Escenario de eventos
  * def createNewEvent =
  """
  {
  "nombreEvento": "BeerJS",
  "tematica": "JavaScript"
  }
  """
  Given path '/api/events'
  And request createNewEvent
  When method post
  Then status 200
```

Dentro de los archivos .feature de Karate, es posible escribir funciones de javascript:

```
* def calcularSuma =
  """
    function(num1, num2){
      var result = num1 + num2
      return result
    }
  """
* karate.log(calcularSuma(12, 2))
```

No se puede utilizar console.log() de javascript, si no que karate.log()

También es posible utilizar simplemente “print” para mostrar por consola valores:

```
* def calcularSuma =
  """
    function(num1, num2){
      var result = num1 + num2
      return result
    }
  """
* print "El resultado de la suma es: ", calcularSuma(12, 2)
```

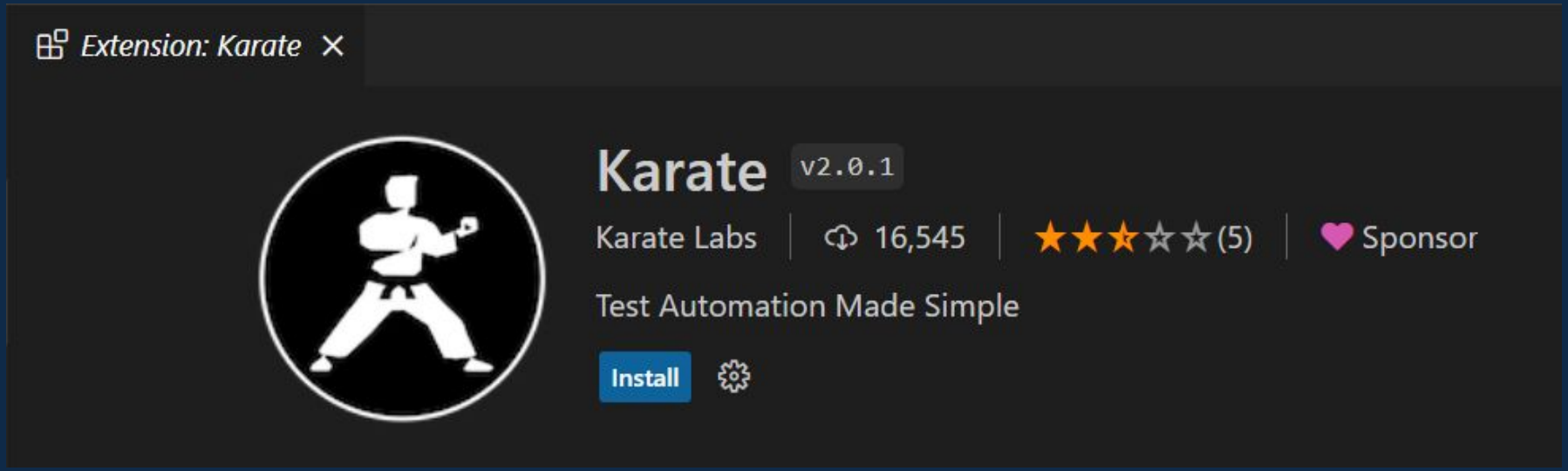
Para poder ejecutar los test, se trata en conjunto la JVM y el motor de Karate-JS, por medio de esto se generan los reportes

JVM

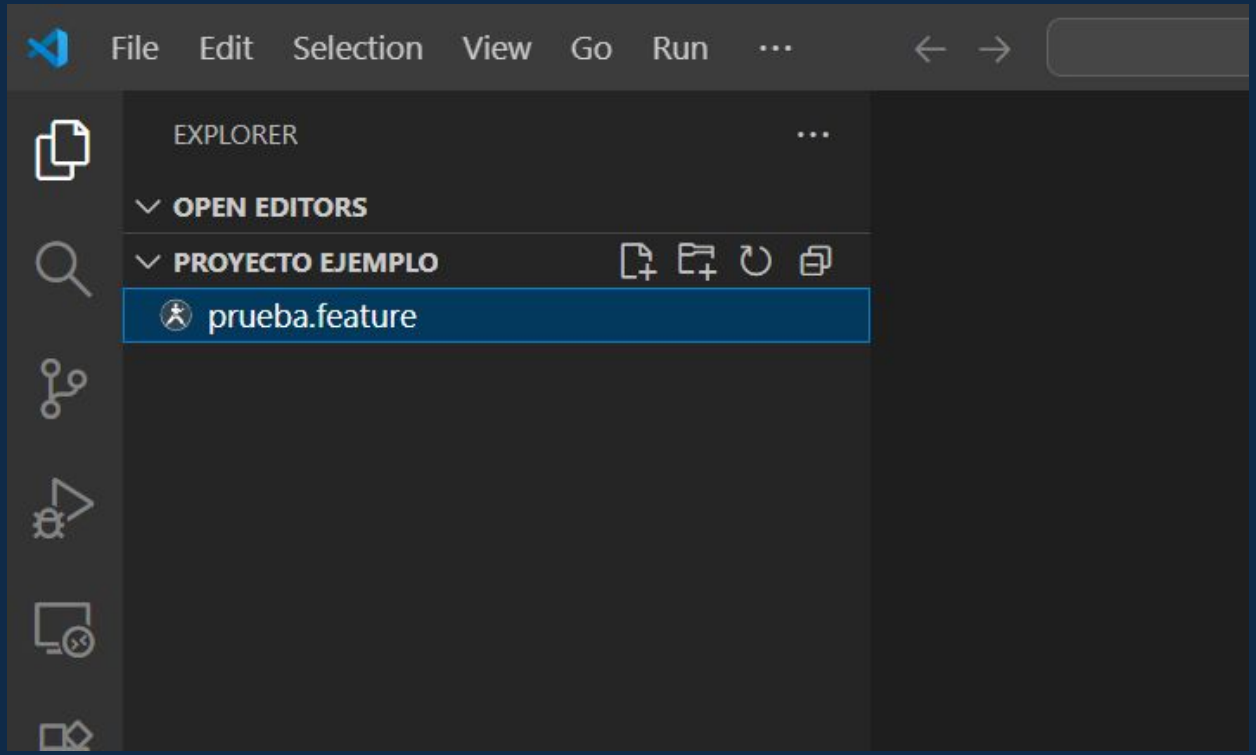
(Java Virtual Machine)

Karate-JS

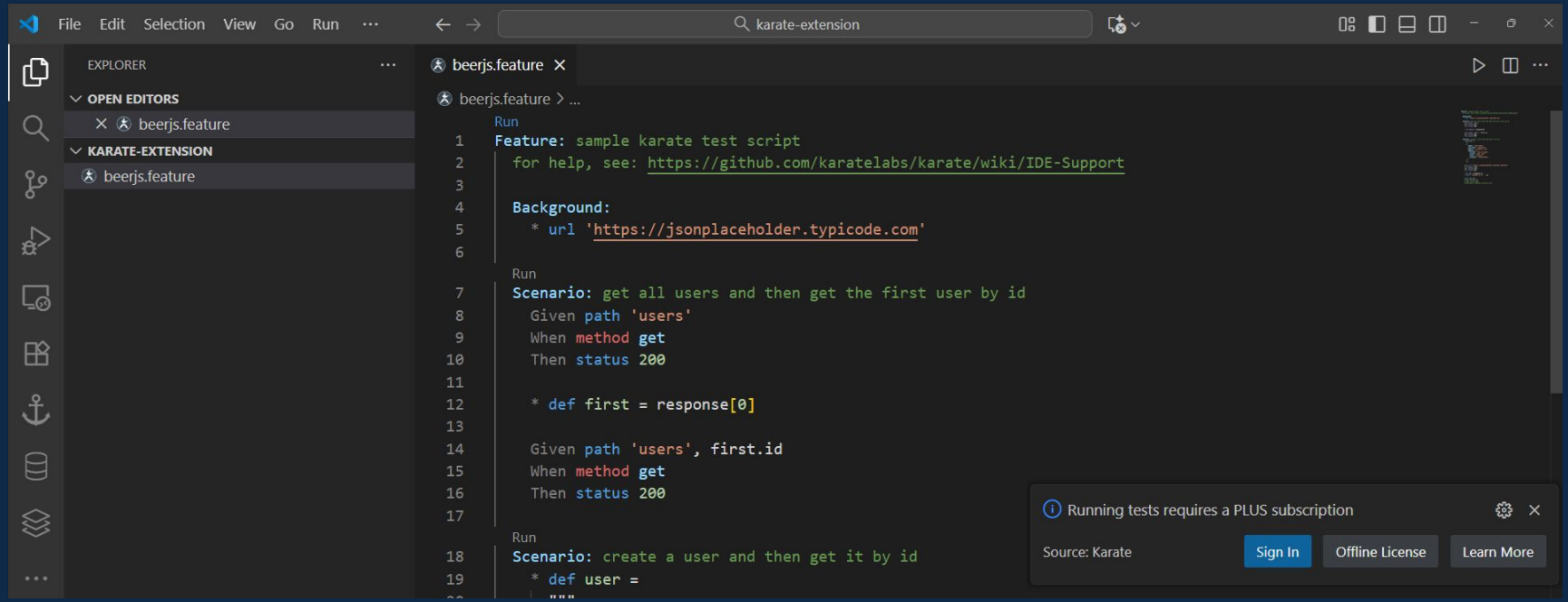
Ahora, sin la necesidad de que sea un proyecto de Java, y tener Maven o Gradle instalado, viene una extensión de VSCode:



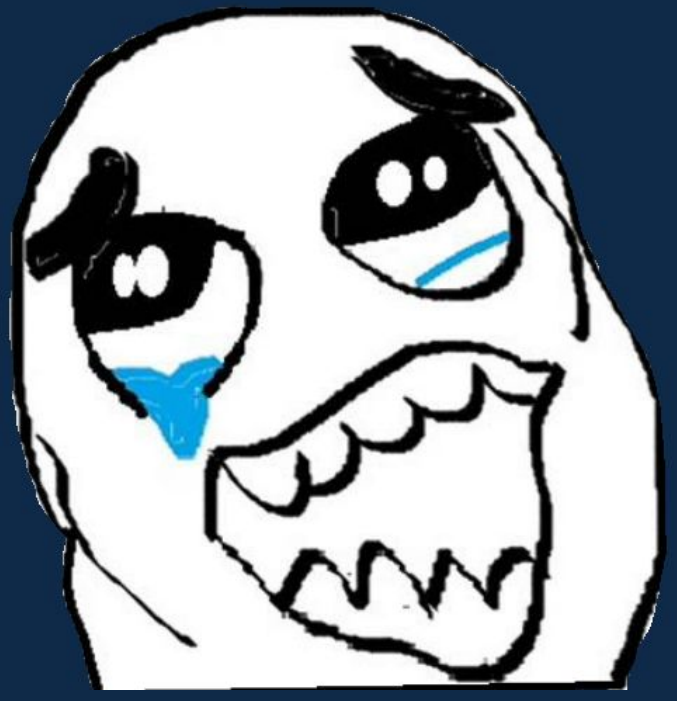
Se puede ejecutar directamente archivos de karate .feature y agregarle código



Se puede ejecutar directamente archivos de karate .feature desde el botón de Run

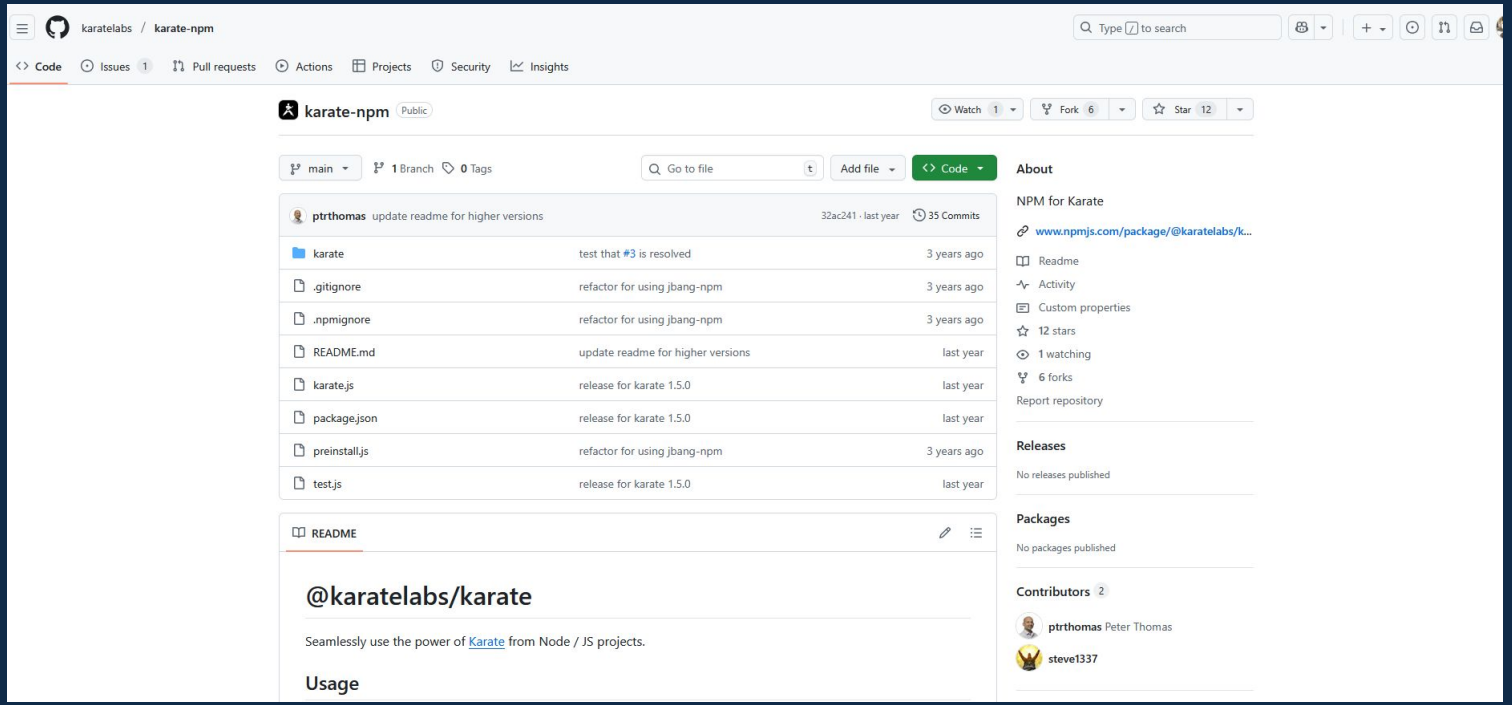


Pero el tema es que esta característica no la podemos utilizar gratuitamente, la han hecho paga hace 2 años atras (2023).



Otra opción para trabajar con Karate en proyectos de javascript con NodeJS con Karate-npm

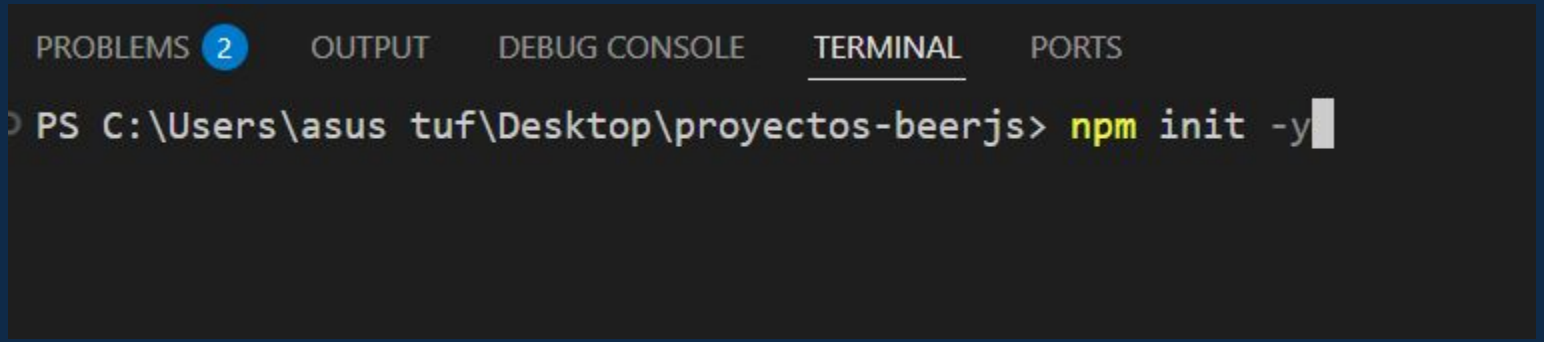
Repo del proyecto Karate-npm



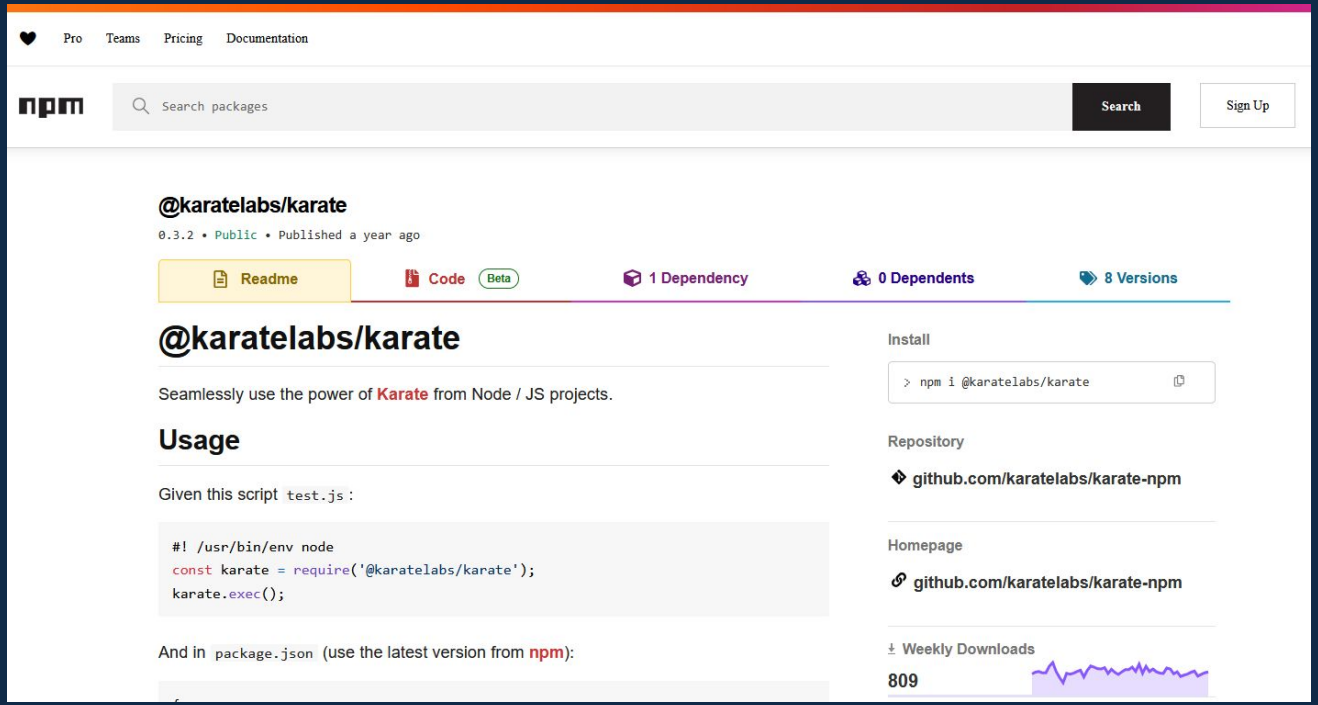
Sitio web: https://github.com/karatelabs/karate-npm

Empezamos a trabajar

Creamos un proyecto de node

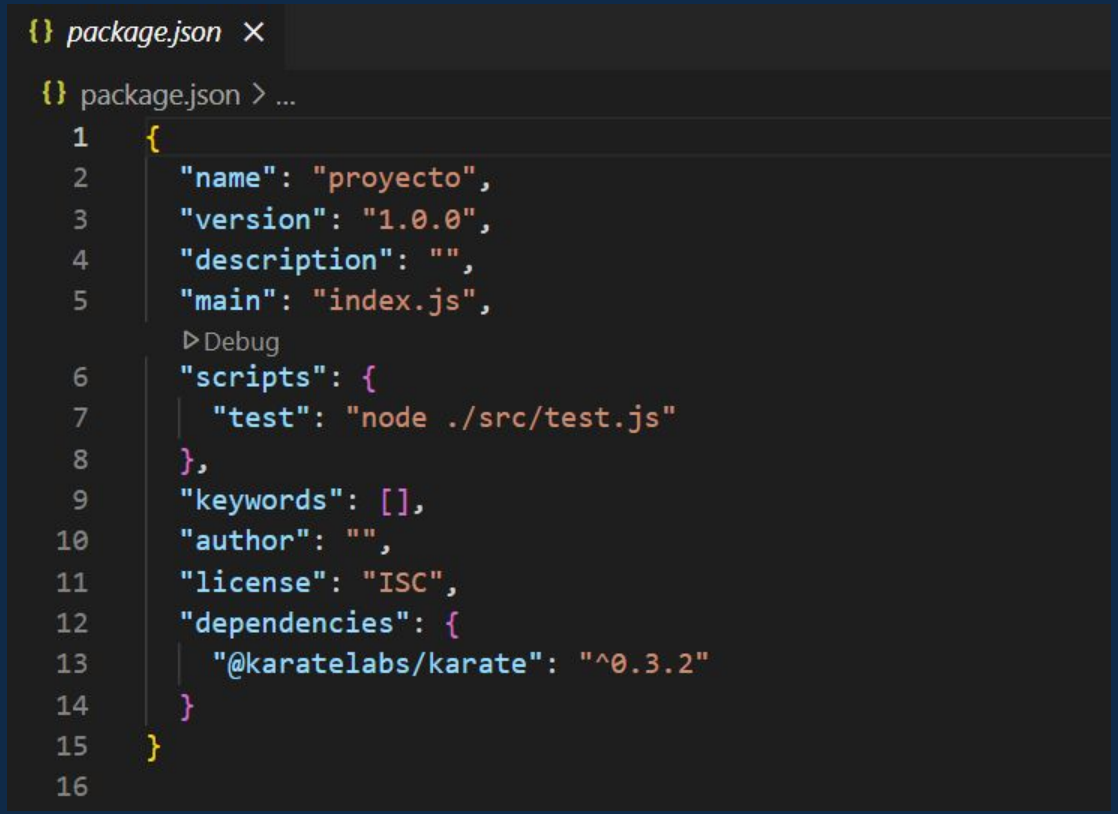


Primero debemos instalar la dependencia de Karate

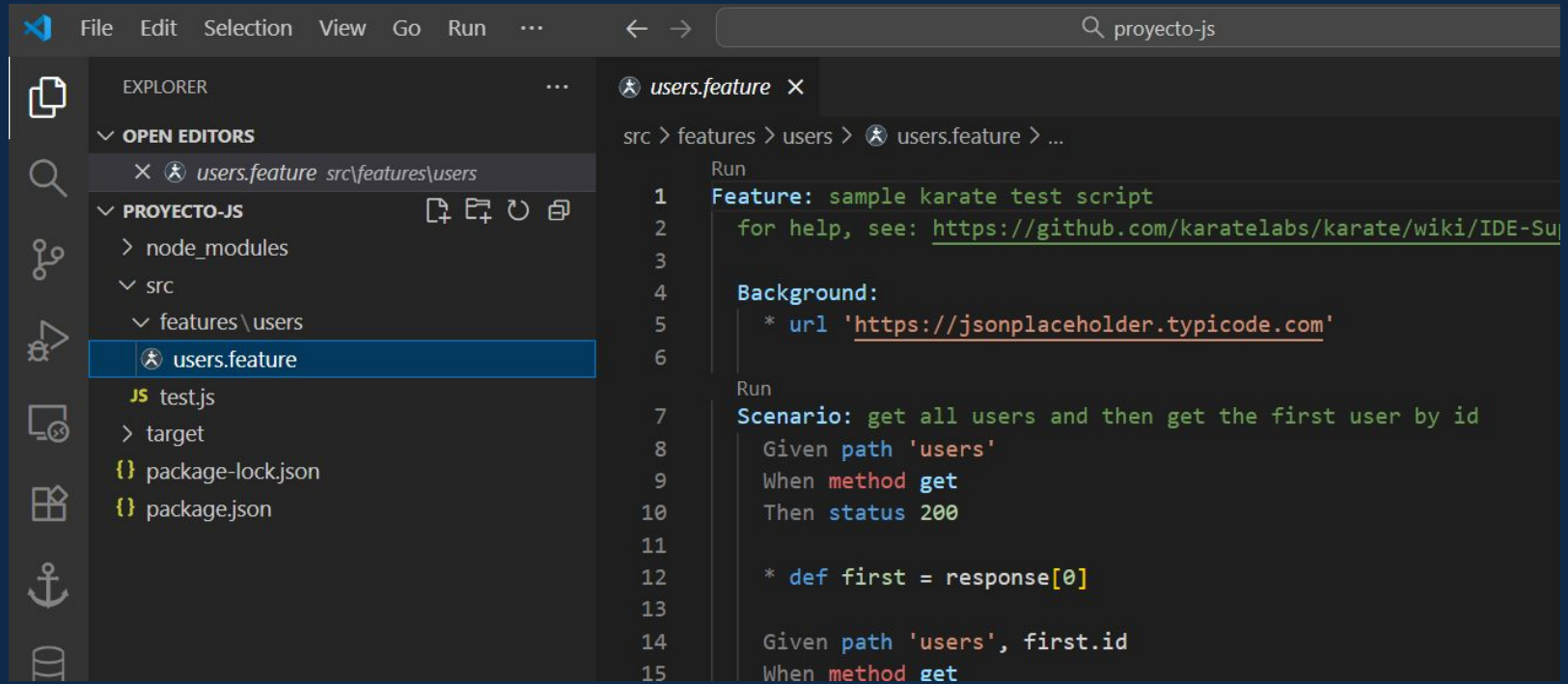


Link: https://www.npmjs.com/package/@karatelabs/karate

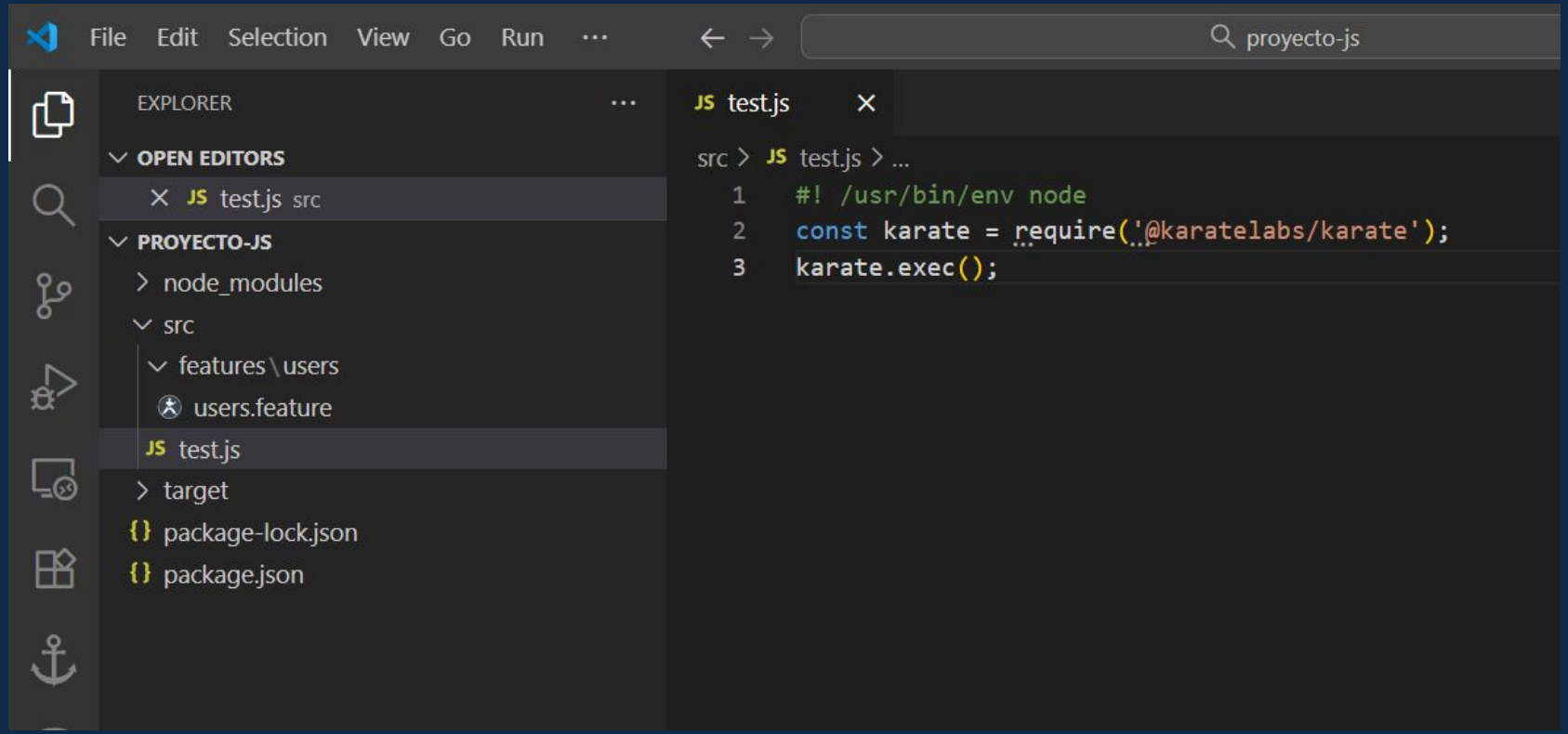
Se guarda en el package.json la dependencia de Karate



Agregamos un archivo de test de karate (.feature)



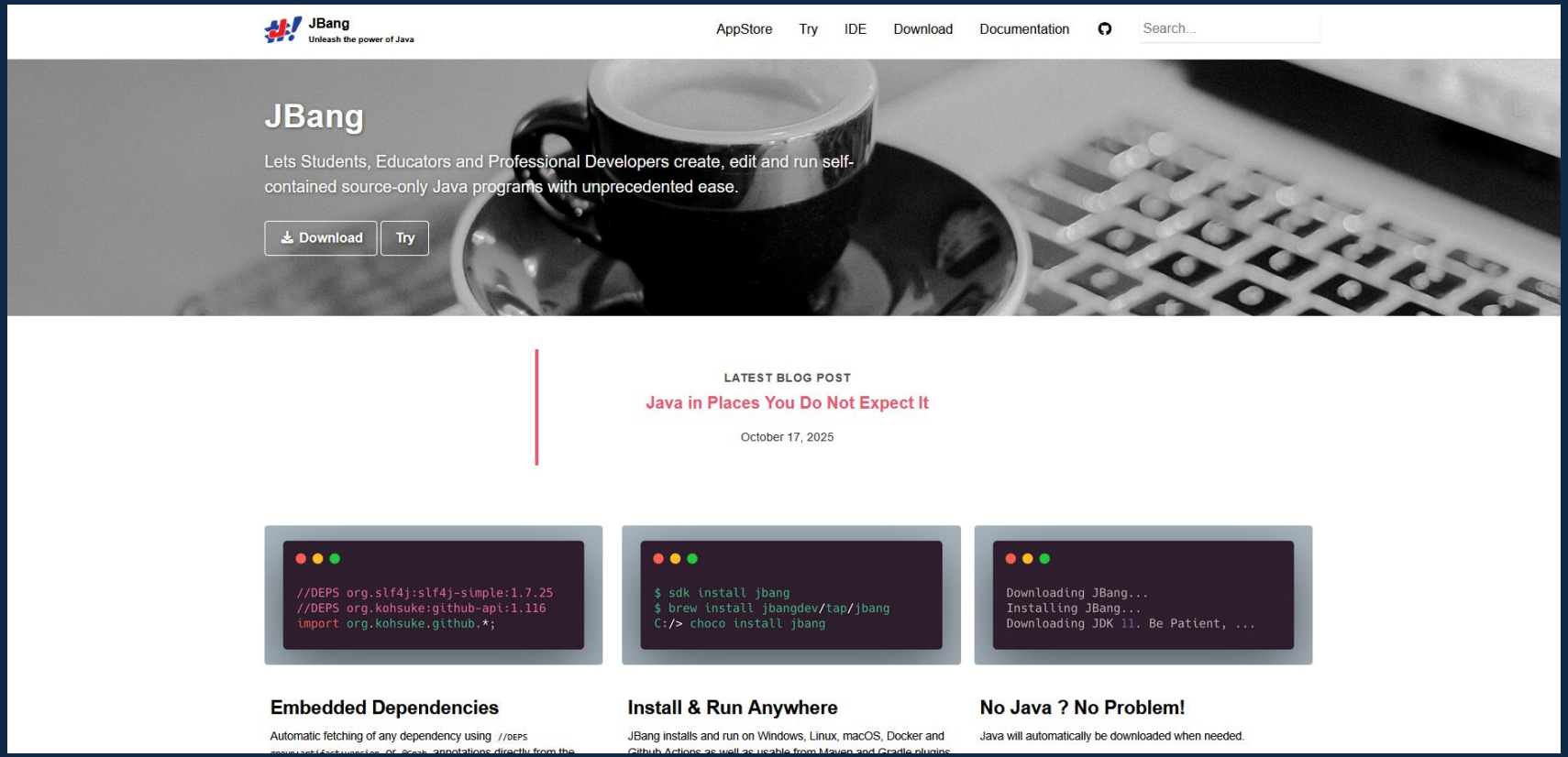
Agregamos un archivo de test de javascript



Para ejecutar los test de Karate, hay que instalar una dependencia que es @jbangdev/jbang que es un orquestador de JVM.

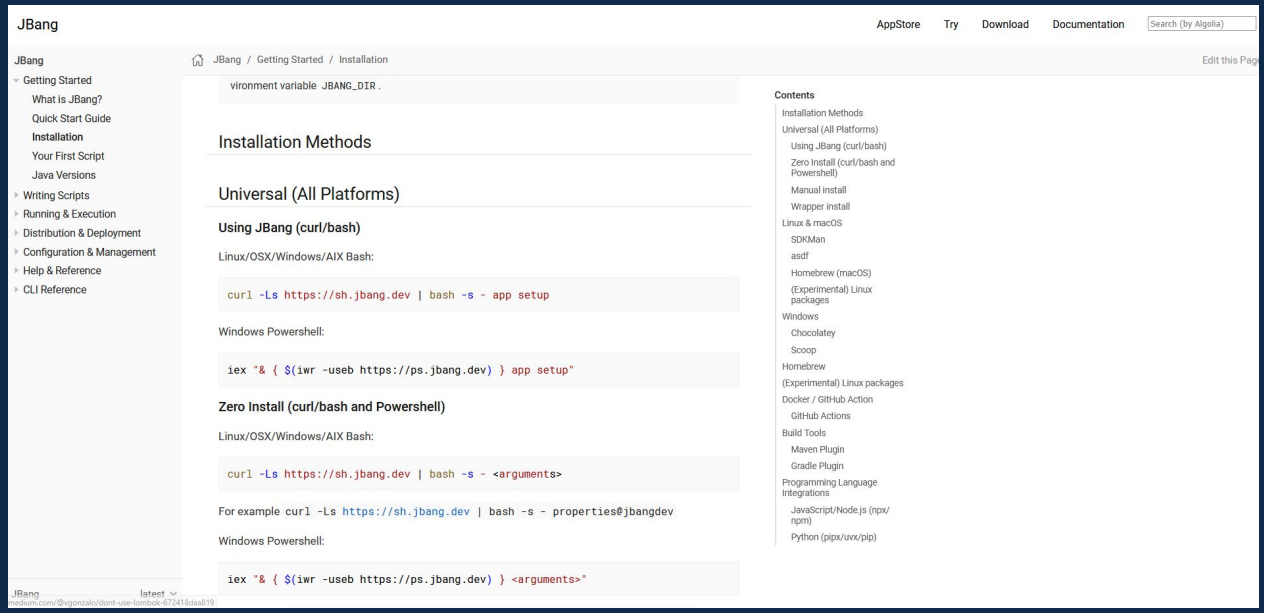
Dos formas de instalar:

- Consultar documentación
- Consultar la dependencia en la página de npm



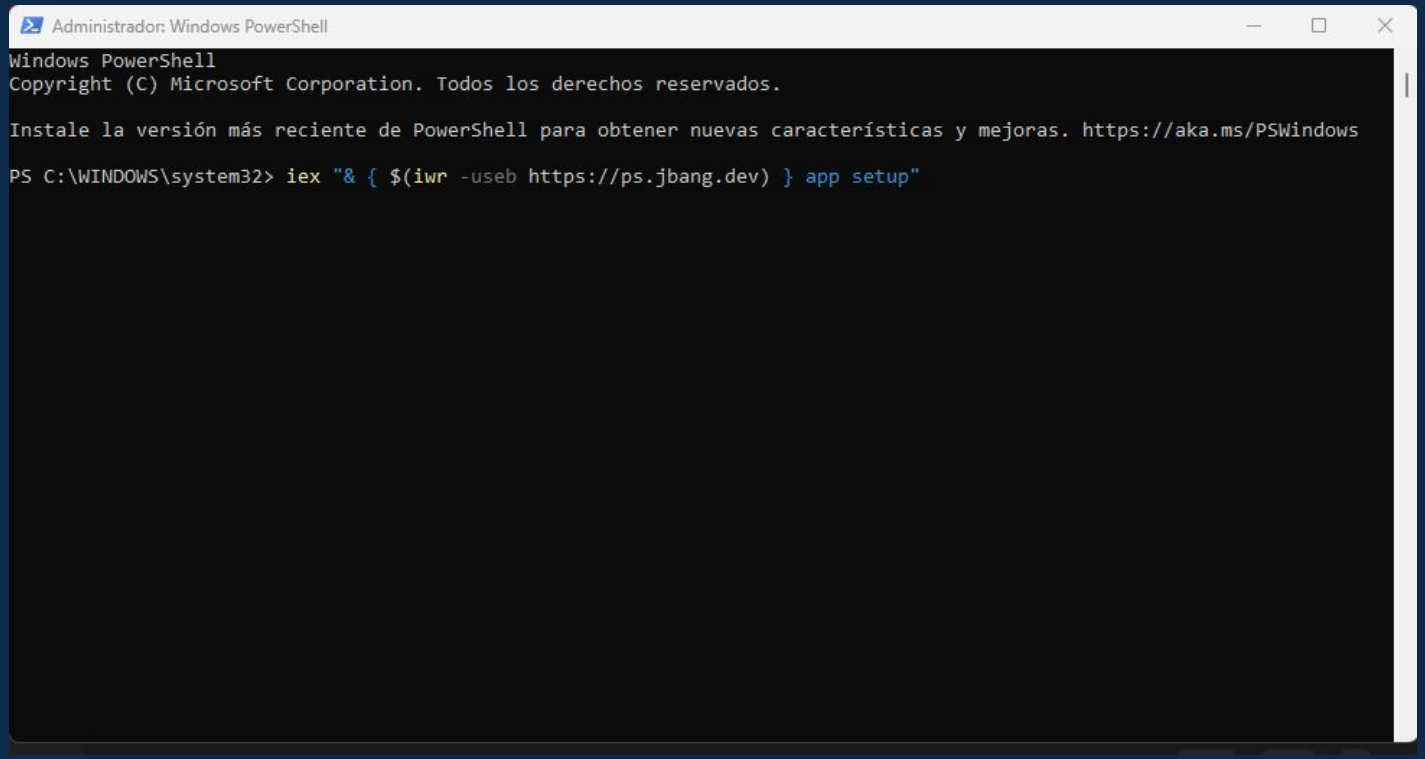
Sitio web: https://www.jbang.dev/

En la documentación nos salen los comandos para instalar JBang en terminal de Linux, PowerShell de Windows y demás

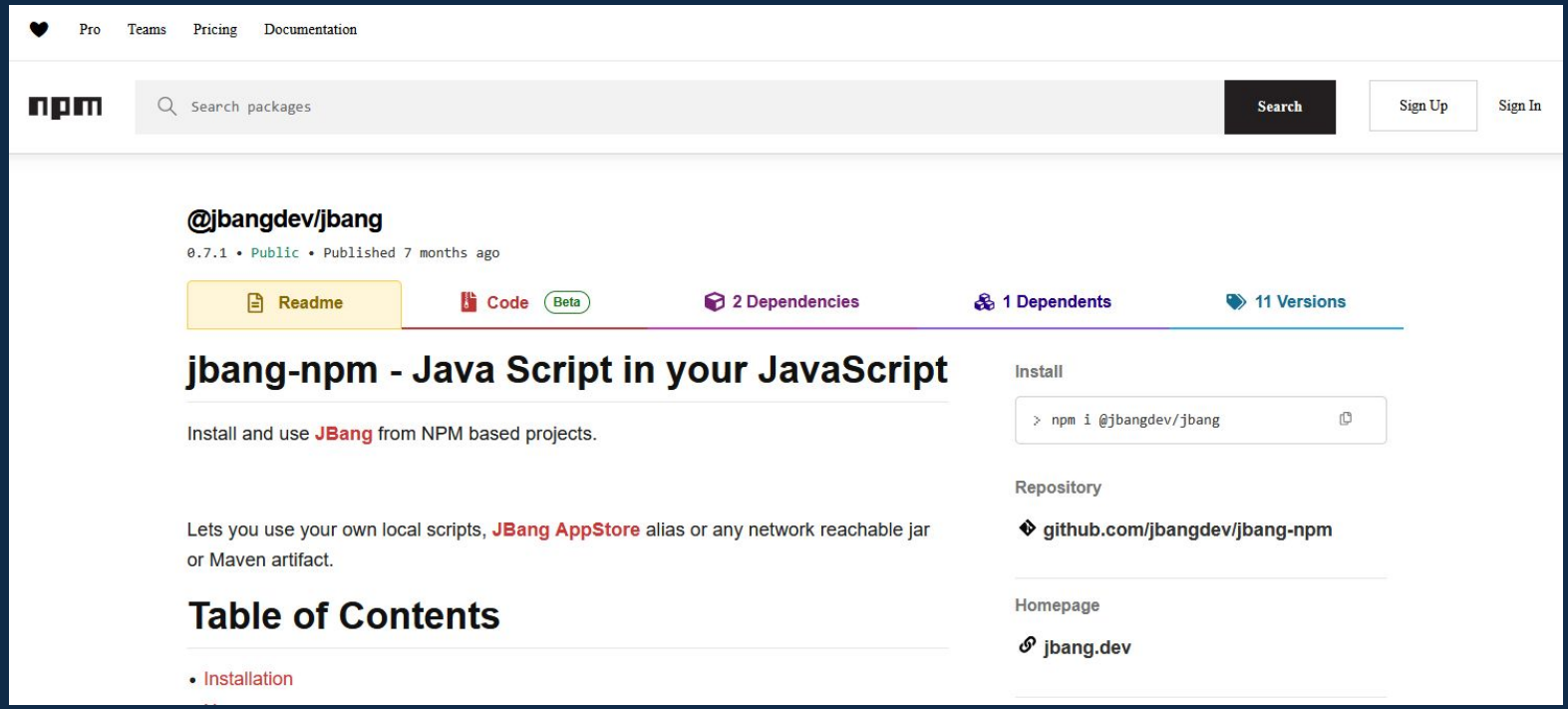


Link: https://www.jbang.dev/documentation/jbang/latest/installation.html

En el PowerShell de Windows instalamos JBang

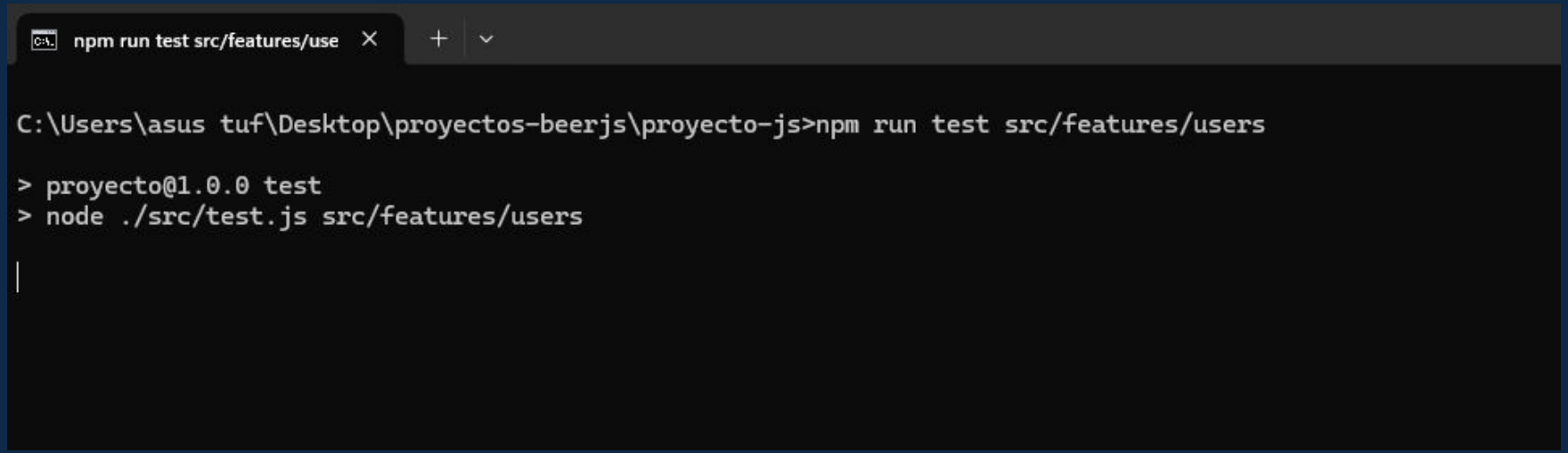


Otra opción para instalar es desde la página de npm

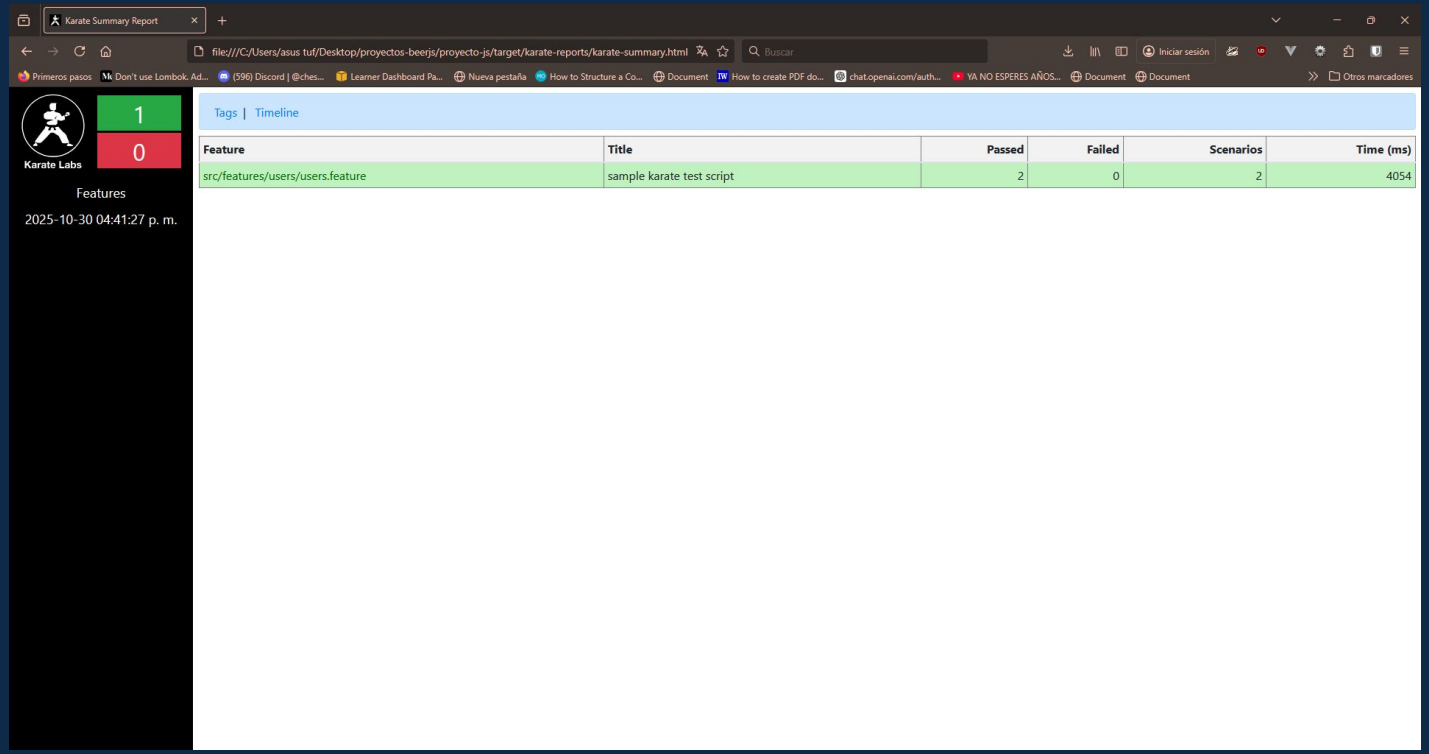


Link: https://www.npmjs.com/package/@jbangdev/jbang

Por último ejecutamos el test desde la terminal de Linux o Windows



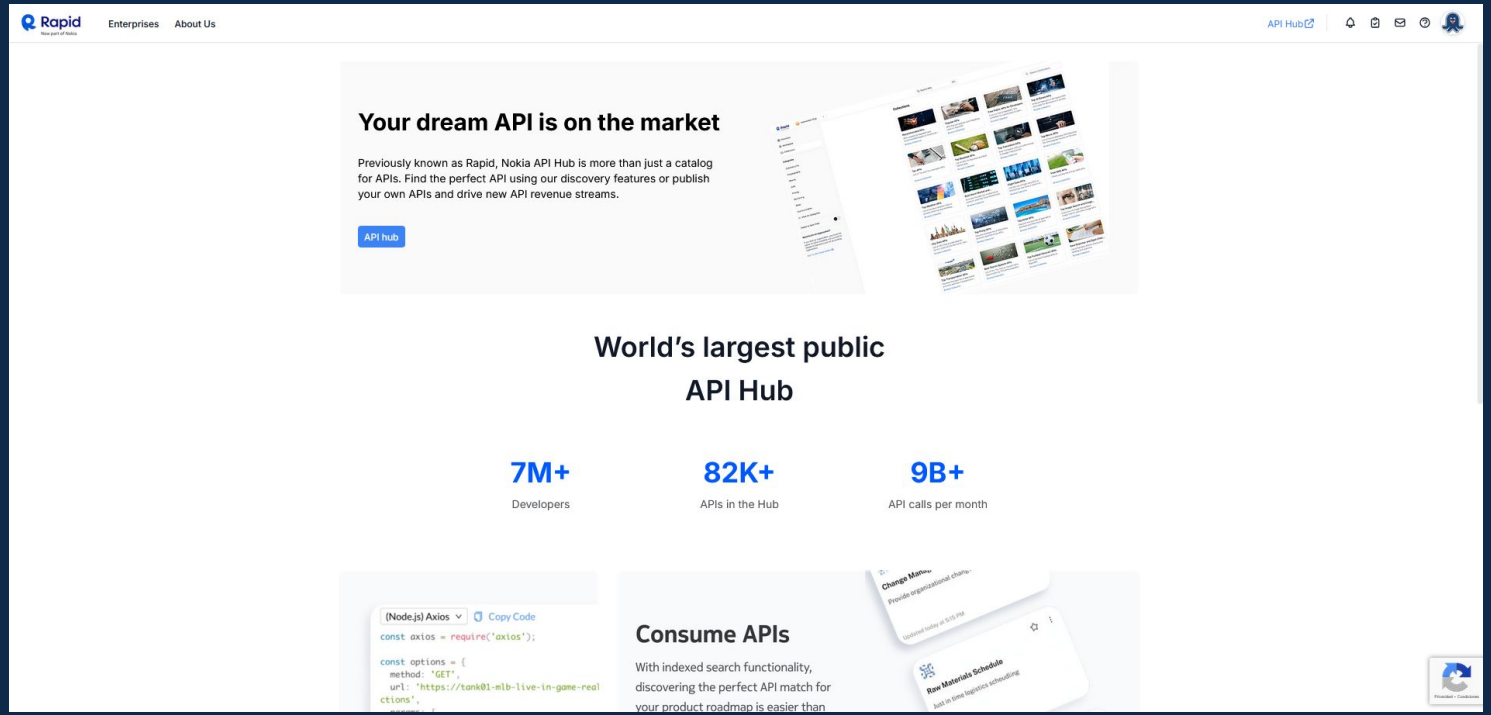
Podremos ver el log en la terminal y genera el reporte que es igual a Karate Framework



Puedo utilizar otras API para realizar pruebas que no sea el genérico de JSONPlaceHolder

Puedo poner las API que yo quiera, pero tengo esta pagina que tiene muchísimas APIs para realizar pruebas, es un mercado de APIs:

RapidAPI: Sitio web para consultar APIs



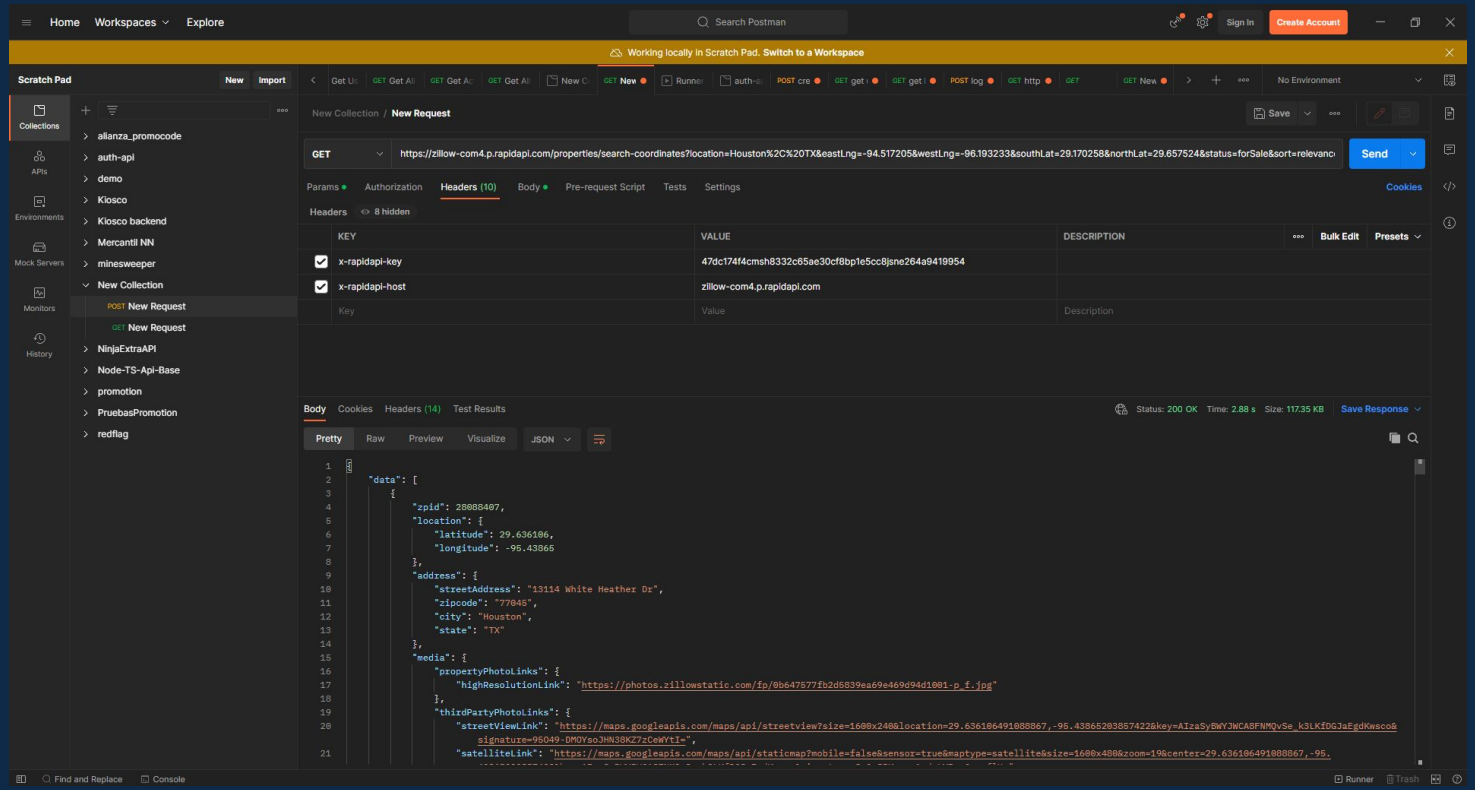
Link: https://rapidapi.com/

Iddo Gino

2015

NOKIA

2024



POSTMAN

Bruno
Reinventing the API Client

Insomnia

¿¿¿Preguntas???

Muchas Gracias

Sergio Gabriel Garzón

sergiog90arg