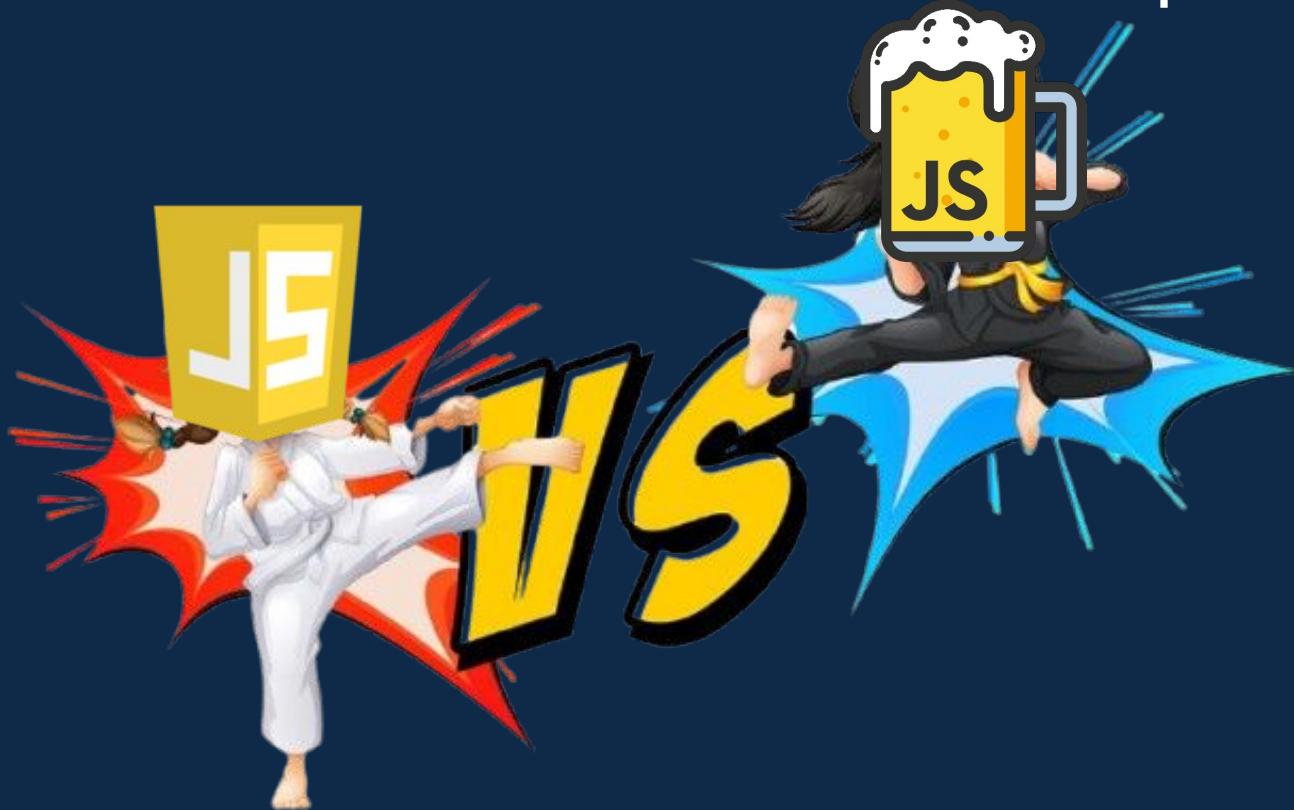


Artes Marciales con JavaScript



SERGIO GARZÓN

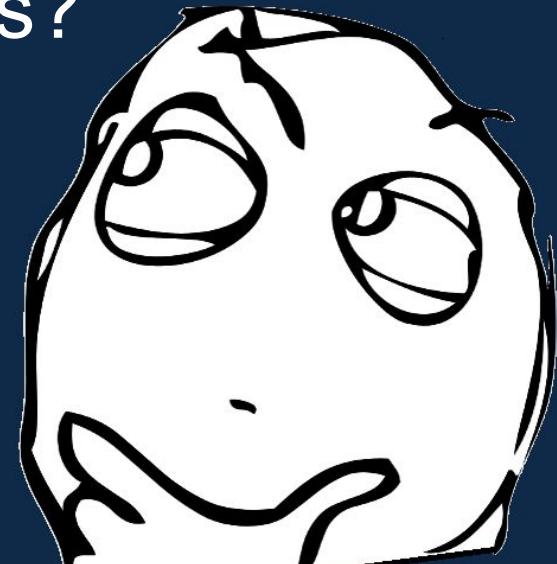
- Desarrollador de Software
- Desarrollador de Videojuegos
- Docente de Programación





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





Karate
Labs

Solutions ▾

Integrations

Plugins ▾

Resources ▾

Pricing

Why Karate ▾

Star

Sign up

Open-Source Test Automation Platform

Unifying API testing, API Performance testing, API Mocks and UI testing.

GET STARTED

WATCH VIDEO

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



karatelabs

Overview Repositories 13 Projects Packages People 2

Karate Labs
Simplicity is Key
Verified

237 followers <https://karatelabs.io> company/karatelabs @karatelabs @getkarate

README.md

Since the first version of Karate was released seven years ago, we have successfully made testing fun and collaborative.

Our 8500 GitHub stars and adoption by the Global 2000 is clear indication that Karate is a top choice for developers, QA teams, and enterprise customers.

Karate was created by a developer - keeping Developer Experience first and foremost. We made test-scripts simpler, more readable, and more maintainable by relying on an innovative DSL (Domain Specific Language), JSON support and JavaScript engine.

Staying true to our mission of making test automation fun & collaborative, we continue to invest in elevating the Developer Experience.

Home How To Tools Enterprise Support Follow Us

Web Site Documentation	Get Started Kick Start Videos Examples	Visual Studio Code IntelliJ IDEA Xplorer	Integrations GitHub Codespaces	Stack Overflow Contact Us	LinkedIn Twitter YouTube
------------------------	--	--	-----------------------------------	------------------------------	--------------------------------

People

Top languages

Java Gherkin JavaScript HTML

Report abuse

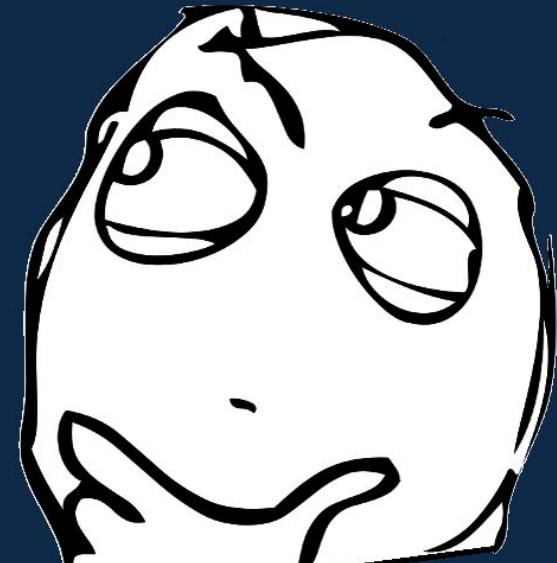
<https://github.com/karatelabs>

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



Interesante....

Pero ¿Para que me sirve?

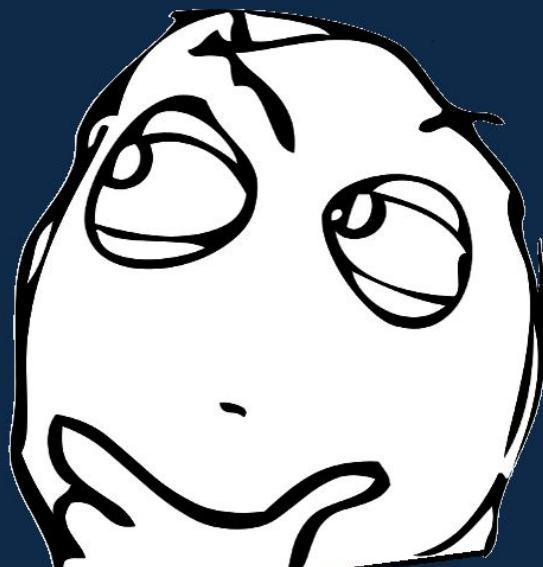


KARATE sirve para muchas cosas:

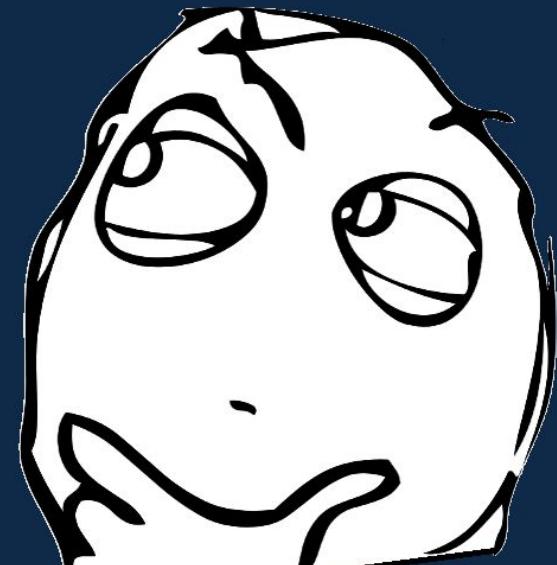
- Nos permite automatizar pruebas de APIs
- Crear script pruebas de APIs
- Simular APIs
- Realizar pruebas de UI
- Automatizar pruebas web, de performance y demás.



¿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

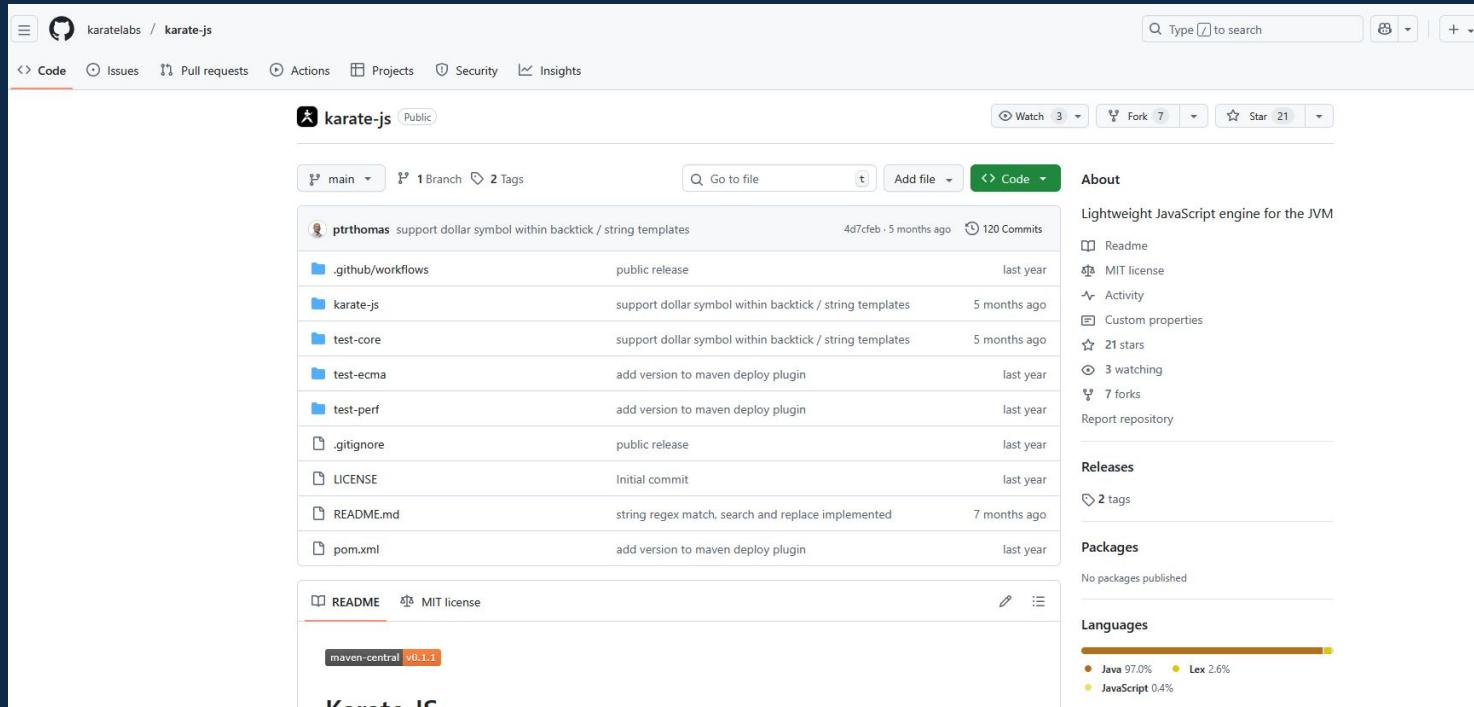


Y lo más importante de todo:

- 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



The screenshot shows the GitHub repository page for 'karatelabs / karate-js'. The repository is public and has 1 branch and 2 tags. The main branch has 120 commits. The repository description is 'Lightweight JavaScript engine for the JVM'. It includes links to Readme, MIT license, Activity, Custom properties, 21 stars, 3 watching, 7 forks, and a Report repository. The Releases section shows 2 tags. The Packages section indicates 'No packages published'. The Languages section shows Java at 97.0%, Lex at 2.6%, and JavaScript at 0.4%.

karatelabs / karate-js

Code Issues Pull requests Actions Projects Security Insights

karate-js Public

main 1 Branch 2 Tags

ptrthomas support dollar symbol within backtick / string templates 4d7feb · 5 months ago 120 Commits

.github/workflows public release last year

karate-js support dollar symbol within backtick / string templates 5 months ago

test-core support dollar symbol within backtick / string templates 5 months ago

test-ecma add version to maven deploy plugin last year

test-perf add version to maven deploy plugin last year

.gitignore public release last year

LICENSE Initial commit last year

README.md string regex match, search and replace implemented 7 months ago

pom.xml add version to maven deploy plugin last year

About

Lightweight JavaScript engine for the JVM

Readme MIT license Activity Custom properties 21 stars 3 watching 7 forks Report repository

Releases

2 tags

Packages

No packages published

Languages

Java 97.0% Lex 2.6% JavaScript 0.4%

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

Se puede integrar con varias tecnologías:



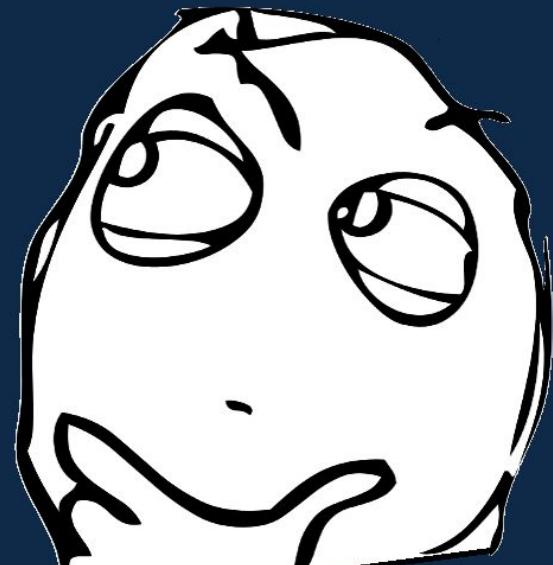
Sitio web: <https://www.karatelabs.io/integrations>

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

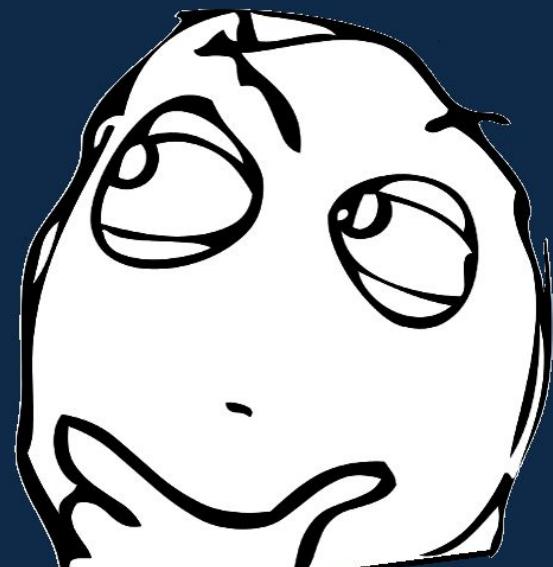




Interesante todo lo que me contas
¿Y tenes alguna demostración para
ver?



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:



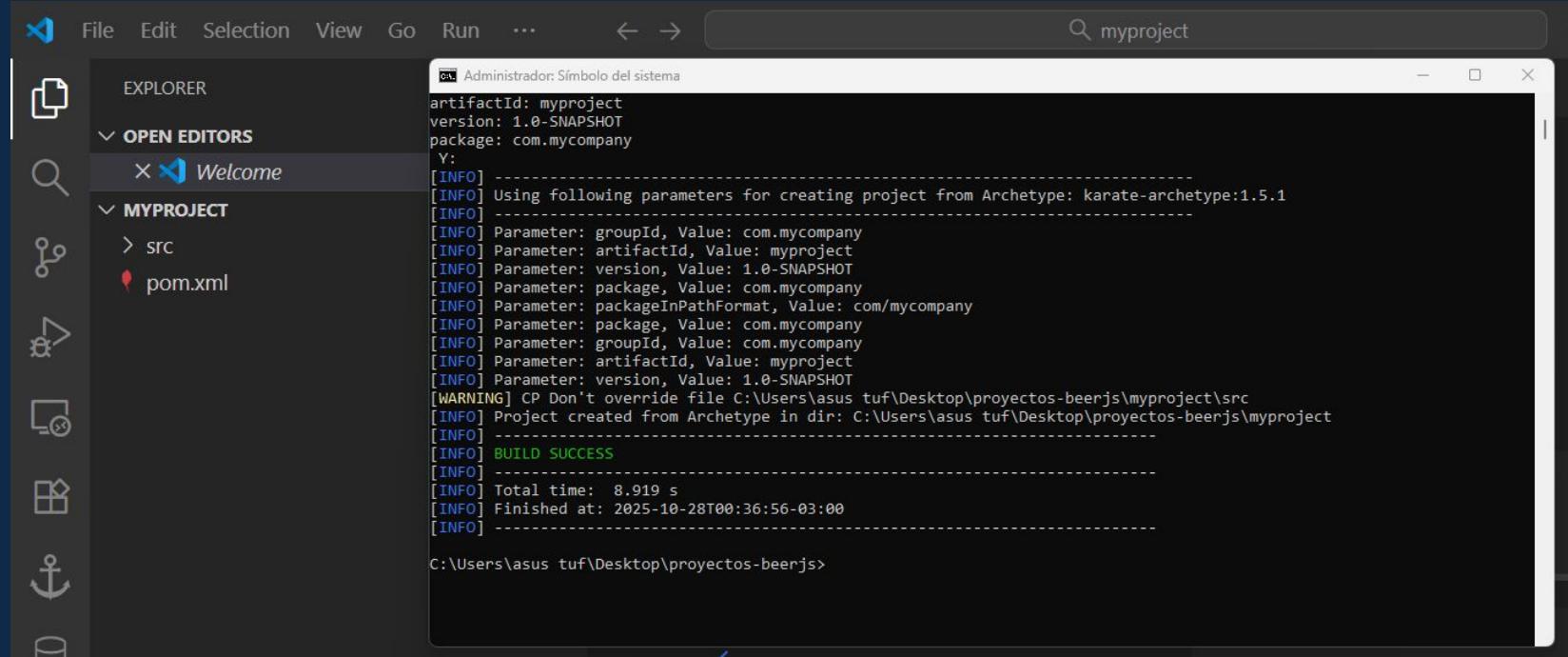
En el repo nos da el comando para generar un proyecto de Maven

```
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



File Edit Selection View Go Run ... ← → myproject

EXPLORER

OPEN EDITORS

Welcome

MYPROJECT

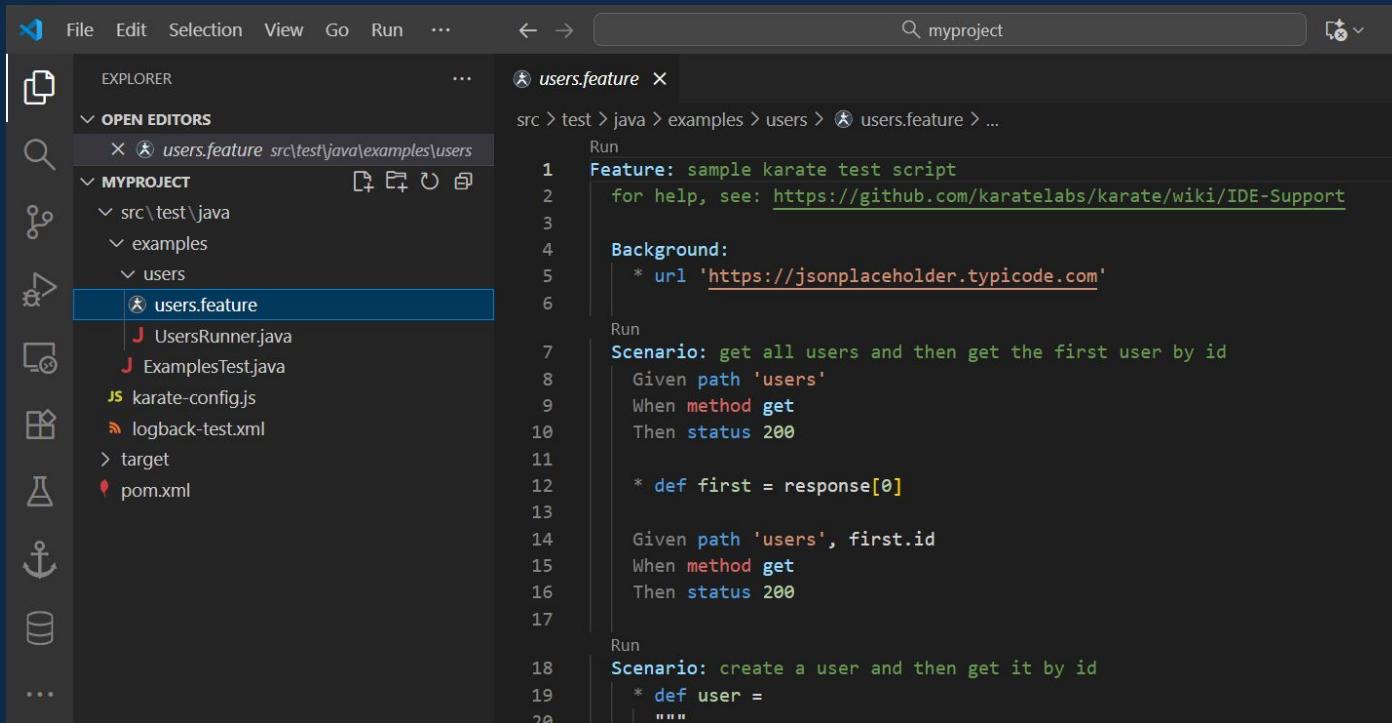
src

pom.xml

```
artifactId: myproject
version: 1.0-SNAPSHOT
package: com.mycompany
Y:
[INFO] -----
[INFO] Using following parameters for creating project from Archetype: karate-archetype:1.5.1
[INFO] -----
[INFO] Parameter: groupId, Value: com.mycompany
[INFO] Parameter: artifactId, Value: myproject
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[INFO] Parameter: package, Value: com.mycompany
[INFO] Parameter: packageInPathFormat, Value: com/mycompany
[INFO] Parameter: package, Value: com.mycompany
[INFO] Parameter: groupId, Value: com.mycompany
[INFO] Parameter: artifactId, Value: myproject
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[WARNING] CP Don't override file C:\Users\asus tuf\Desktop\proyectos-beerjs\myproject\src
[INFO] Project created from Archetype in dir: C:\Users\asus tuf\Desktop\proyectos-beerjs\myproject
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 8.919 s
[INFO] Finished at: 2025-10-28T00:36:56-03:00
[INFO] -----
```

C:\Users\asus tuf\Desktop\proyectos-beerjs>

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

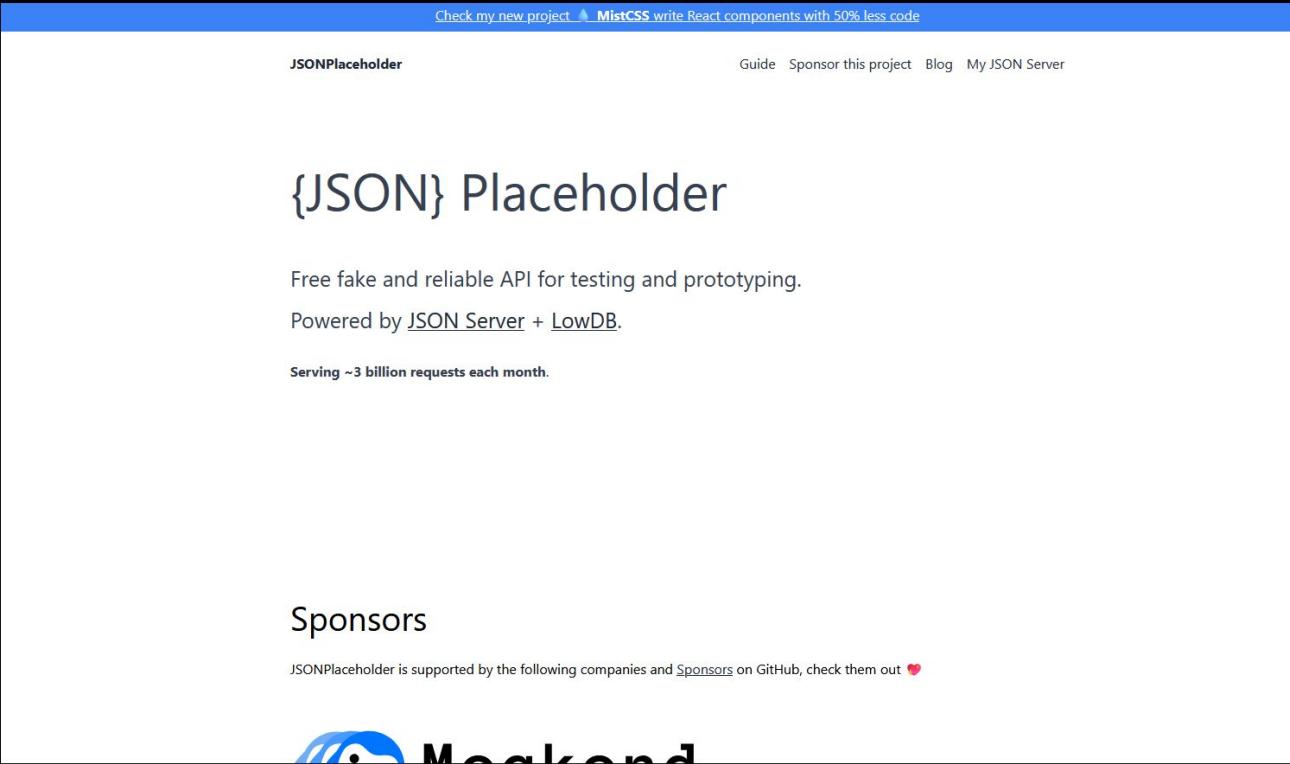


The screenshot shows a dark-themed IDE interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** myproject
- Explorer:** Shows the project structure:
 - OPEN EDITORS: users.feature
 - MYPROJECT:
 - src\test\java
 - examples
 - users
 - File list:
 - UsersRunner.java
 - ExamplesTest.java
 - karate-config.js
 - logback-test.xml
 - target
 - pom.xml
- Code Editor:** The users.feature file is open and visible:

```
1 Feature: sample karate test script
2   for help, see: https://github.com/karatelabs/karate/wiki/IDE-Support
3
4 Background:
5   * url 'https://jsonplaceholder.typicode.com'
6
7 Run
8 Scenario: get all users and then get the first user by id
9   Given path 'users'
10  When method get
11  Then status 200
12
13  * def first = response[0]
14
15  Given path 'users', first.id
16  When method get
17  Then status 200
18
19 Scenario: create a user and then get it by id
20   * def user =
21   """
22   
```

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



The screenshot shows the homepage of the JSONPlaceholder API. At the top, there is a blue header bar with the text "Check my new project" and "MistCSS write React components with 50% less code". Below the header, the "JSONPlaceholder" logo is on the left, and a navigation bar with links to "Guide", "Sponsor this project", "Blog", and "My JSON Server" is on the right. The main content area features a large, bold title "JSON Placeholder" and a subtext "Free fake and reliable API for testing and prototyping." Below this, it says "Powered by [JSON Server](#) + [LowDB](#)." A small note at the bottom states "Serving ~3 billion requests each month." In the bottom left corner, there is a section titled "Sponsors" with a list of companies. At the very bottom, there is a footer with the "Mockapi" logo and the text "Mockapi is a free API for testing and prototyping".

Check my new project  MistCSS write React components with 50% less code

JSONPlaceholder

Guide Sponsor this project Blog My JSON Server

{JSON} Placeholder

Free fake and reliable API for testing and prototyping.

Powered by [JSON Server](#) + [LowDB](#).

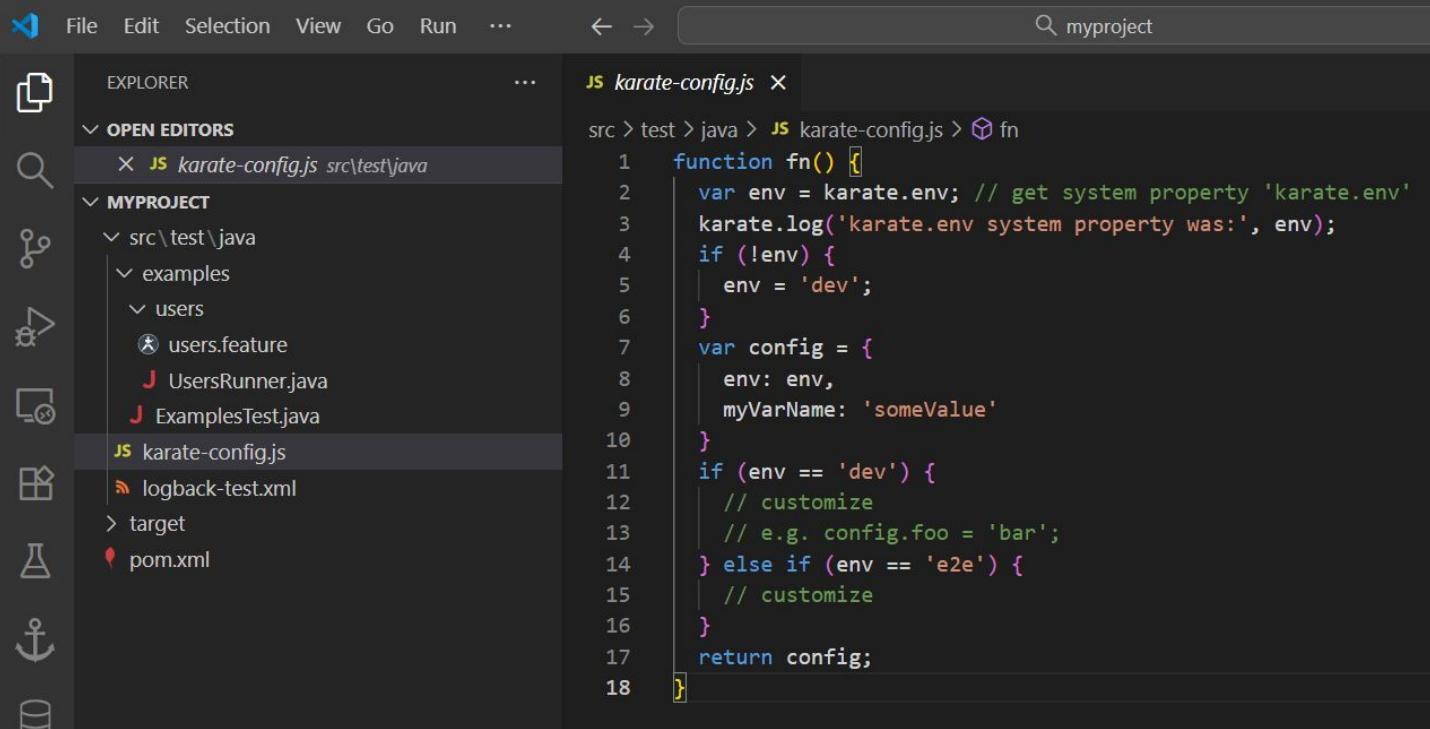
Serving ~3 billion requests each month.

Sponsors

JSONPlaceholder is supported by the following companies and [Sponsors](#) on GitHub, check them out 

 **Mockapi**

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



The screenshot shows a code editor interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** myproject
- Explorer:** Shows the project structure under 'OPEN EDITORS':
 - src\test\java (selected)
 - examples
 - users
 - users.feature
 - UsersRunner.java
 - ExamplesTest.java
 - karate-config.js (selected)
 - logback-test.xml
 - target
 - pom.xml
- Code Editor:** The 'karate-config.js' file content is displayed:

```
1  function fn() {
2    var env = karate.env; // get system property 'karate.env'
3    karate.log('karate.env system property was:', env);
4    if (!env) {
5      env = 'dev';
6    }
7    var config = {
8      env: env,
9      myVarName: 'someValue'
10   }
11   if (env == 'dev') {
12     // customize
13     // e.g. config.foo = 'bar';
14   } else if (env == 'e2e') {
15     // customize
16   }
17   return config;
18 }
```

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

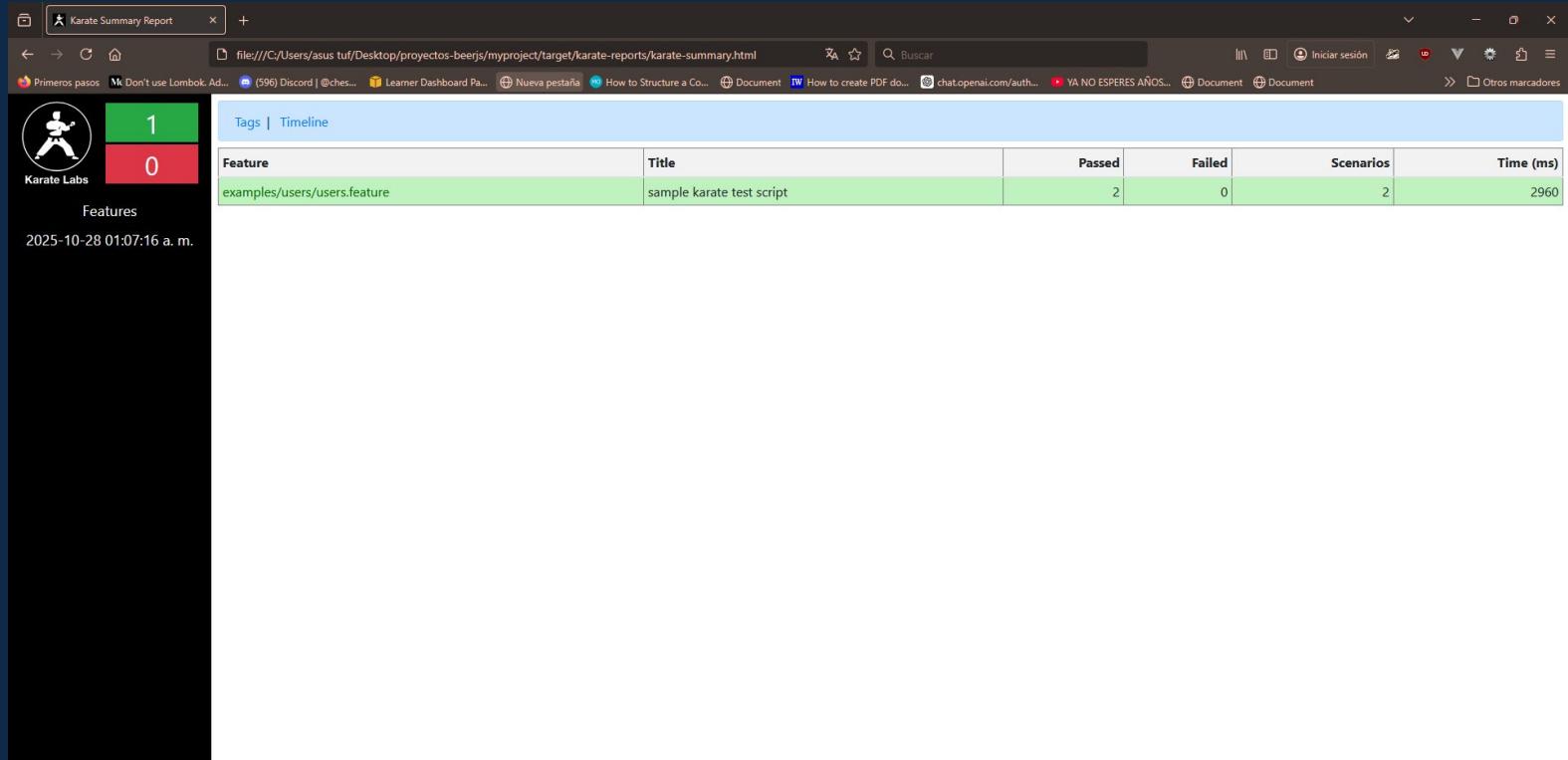
Para instalar las dependencias de Karate, junto con las otras, entonces ejecuto el comando de Maven

```
● 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



Karate Summary Report

file:///C:/Users/asus tuf/Desktop/proyectos-beerjs/myproject/target/karate-reports/karate-summary.html

Primeros pasos M₆ Don't use Lombok. Ad... (596) Discord | @ches... Learner Dashboard Pa... Nueva pestaña How to Structure a Co... Document How to create PDF do... chat.openai.com/auth... YA NO ESPERES AÑOS... Document Document Otros marcadores

Karate Labs

1

0

Features

2025-10-28 01:07:16 a.m.

Feature	Title	Passed	Failed	Scenarios	Time (ms)
examples/users/users.feature	sample karate test script	2	0	2	2960

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 demás...)
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:

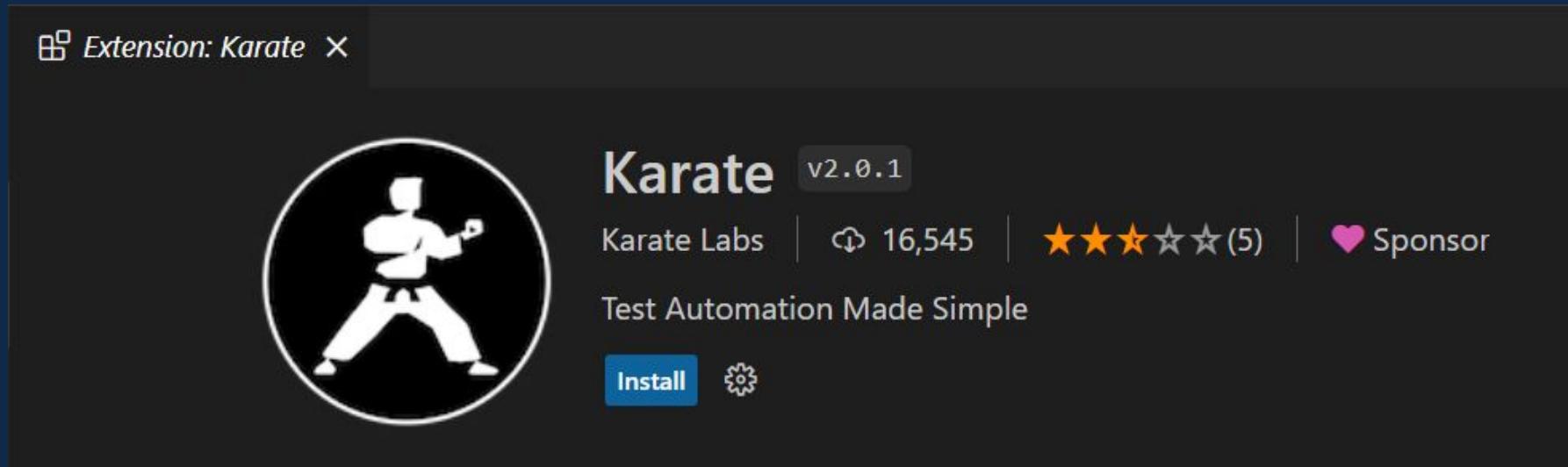
```
* def calcularSuma =
"""
    function(num1, num2){
        var result = num1 + num2
        return result
    }
"""
* print "El resultado de la suma es: ", calcularSuma(12, 2)
```

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

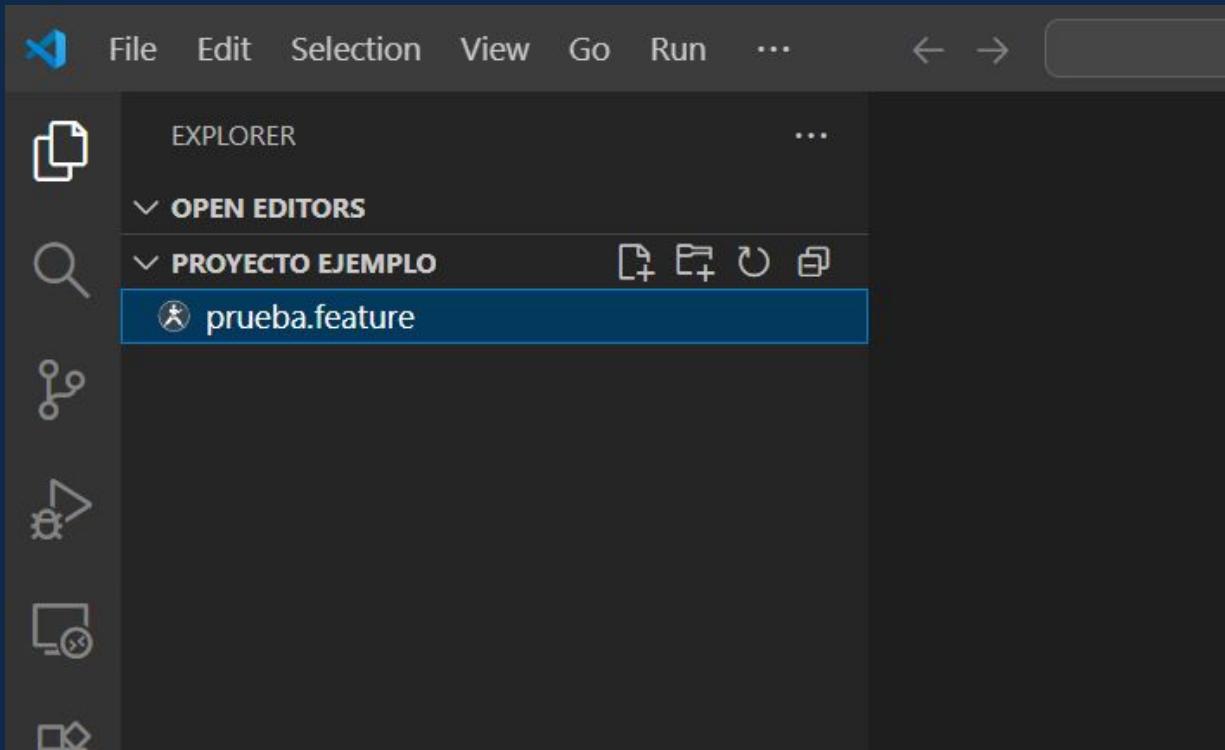


Karate-JS

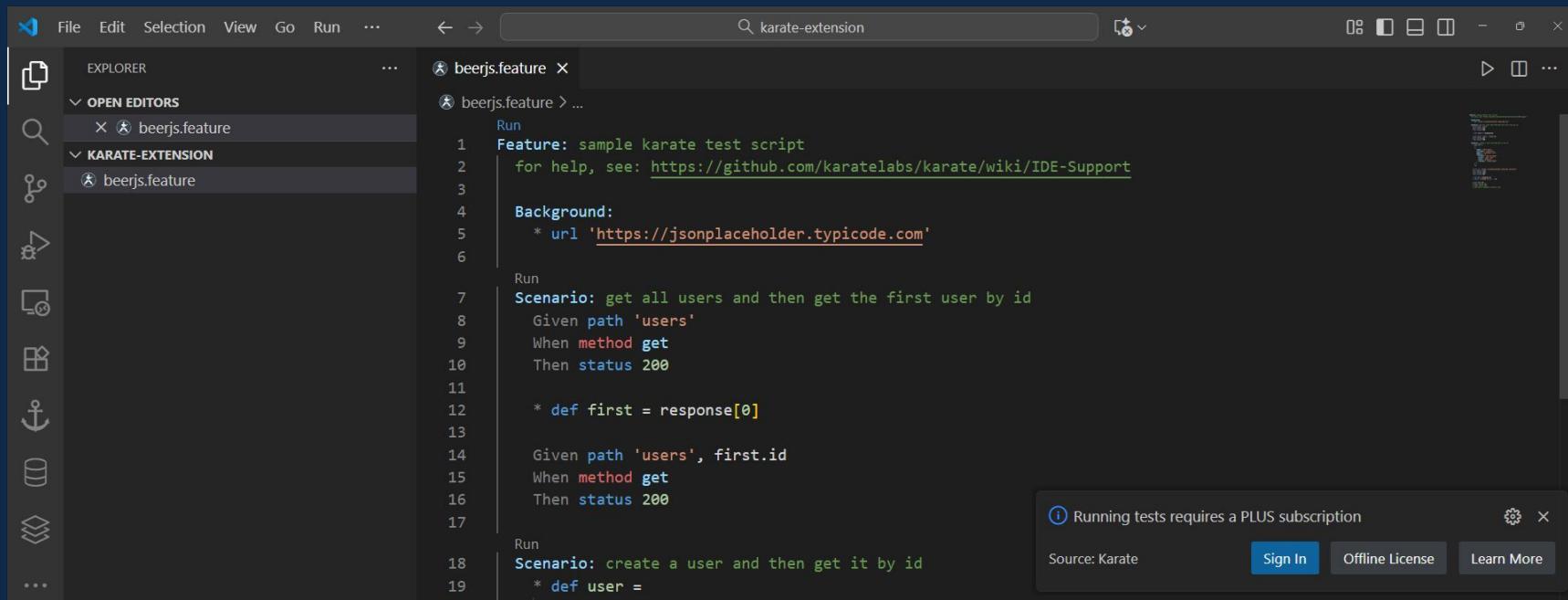
Ahora, sin la necesidad de que sea un proyecto de Java, y tener Maven o Gradle instalado, viene una extensión de VSCode:



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



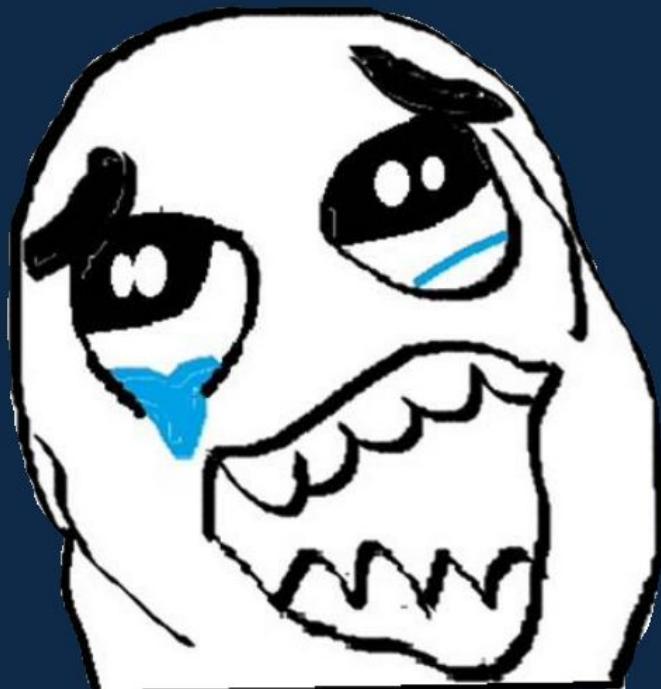
Se puede ejecutar directamente archivos de karate .feature desde el botón de Run



The screenshot shows the Visual Studio Code (VS Code) interface with the Karate extension installed. The left sidebar has icons for file operations, the Explorer (with 'beerjs.feature' listed), and the KARATE-EXTENSION section (also showing 'beerjs.feature'). The main editor area displays a Karate test script named 'beerjs.feature'. The script includes a Feature block with a sample test, a Background block with a URL, and two Scenarios: one for getting users and another for creating a user. A 'Run' button is visible above the scenarios. A status bar at the bottom right indicates 'Source: Karate' and shows options to 'Sign In', 'Offline License', and 'Learn More'. A notification in the bottom right corner states 'Running tests requires a PLUS subscription'.

```
File Edit Selection View Go Run ... ← → 🔍 karate-extension ⚙️ ...  
EXPLORER beerjs.feature ...  
OPEN EDITORS beerjs.feature > ...  
KARATE-EXTENSION beerjs.feature  
Run  
1 Feature: sample karate test script  
2   for help, see: https://github.com/karatelabs/karate/wiki/IDE-Support  
3  
4 Background:  
5   * url 'https://jsonplaceholder.typicode.com'  
6  
Run  
7 Scenario: get all users and then get the first user by id  
8   Given path 'users'  
9   When method get  
10  Then status 200  
11  
12  * def first = response[0]  
13  
14  Given path 'users', first.id  
15  When method get  
16  Then status 200  
17  
Run  
18 Scenario: create a user and then get it by id  
19   * def user =  
20   | ...  
...  
Running tests requires a PLUS subscription  
Source: Karate Sign In Offline License Learn More
```

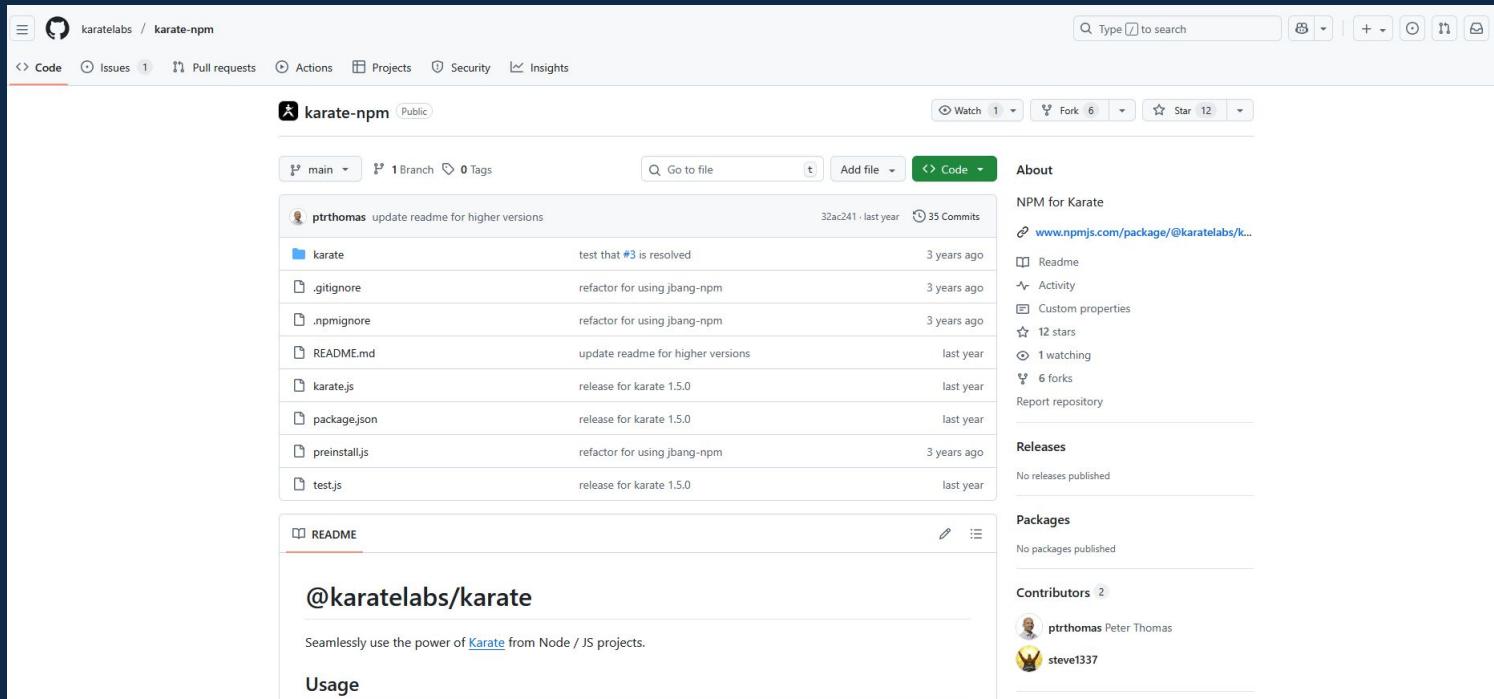
Pero el tema es que esta característica no la podemos utilizar gratuitamente, la han hecho paga hace 2 años atrás (2023).



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



Repo del proyecto Karate-npm



The screenshot shows the GitHub repository page for 'karatelabs/karate-npm'. The repository is public and has 1 branch and 0 tags. The main file listed is 'main'. The repository was last updated 3 years ago. The README file is present and contains the text: '@karatelabs/karate'. The README also states: 'Seamlessly use the power of [Karate](#) from Node / JS projects.' The repository has 12 stars, 6 forks, and 12 watching. It has 35 commits from the user 'ptrthomas'. The repository has no releases or packages published. The contributors are 'ptrthomas' and 'steve1337'.

karatelabs / karate-npm

Code Issues Pull requests Actions Projects Security Insights

karate-npm Public

main 1 Branch 0 Tags

ptrthomas update readme for higher versions 32ac241 · last year 35 Commits

karate test that #3 is resolved 3 years ago

.gitignore refactor for using jbang-npm 3 years ago

.npmignore refactor for using jbang-npm 3 years ago

README.md update readme for higher versions last year

karate.js release for karate 1.5.0 last year

package.json release for karate 1.5.0 last year

preinstall.js refactor for using jbang-npm 3 years ago

test.js release for karate 1.5.0 last year

About

NPM for Karate

www.npmjs.com/package/@karatelabs/karate

Readme Activity Custom properties 12 stars 1 watching 6 forks Report repository

Releases

No releases published

Packages

No packages published

Contributors 2

ptrthomas Peter Thomas

steve1337

README

@karatelabs/karate

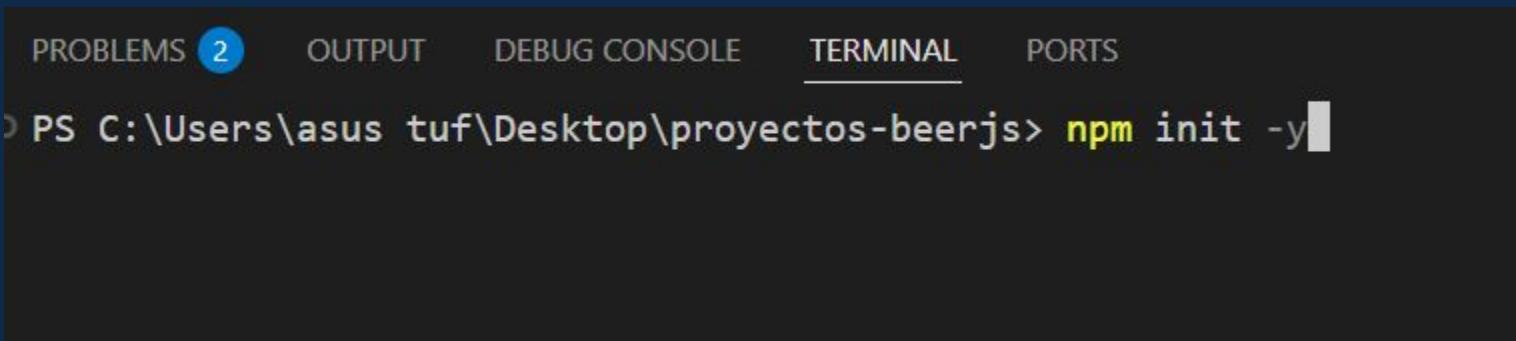
Seamlessly use the power of [Karate](#) from Node / JS projects.

Usage

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

Empezamos a trabajar

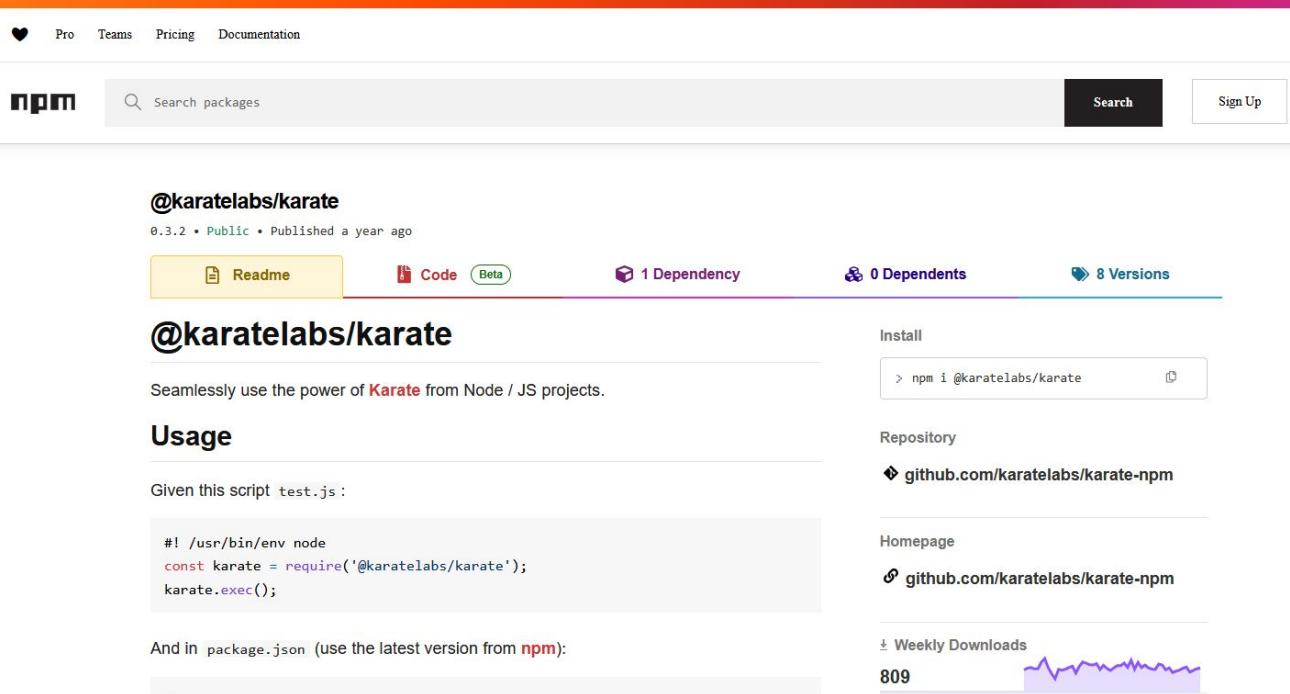
Creamos un proyecto de node



PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\asus tuf\Desktop\proyectos-beerjs> npm init -y
```

Primero debemos instalar la dependencia de Karate



The screenshot shows the npmjs.com package page for the `@karatelabs/karate` package. The page has a dark header with a light bar containing navigation links: **Pro**, **Teams**, **Pricing**, and **Documentation**. Below the header is the npm logo and a search bar with the placeholder "Search packages". To the right of the search bar are a "Search" button and a "Sign Up" button. The main content area has a white background. At the top of the content area, the package name `@karatelabs/karate` is displayed in bold, with a version of `0.3.2`, a `Public` status, and a note that it was published a year ago. Below this, there are tabs: **Readme** (highlighted in yellow), **Code** (Beta), **1 Dependency**, **0 Dependents**, and **8 Versions**. The **Readme** tab contains the following text: **@karatelabs/karate**, *Seamlessly use the power of Karate from Node / JS projects.* The **Usage** section includes a code snippet for a `test.js` file:

```
#! /usr/bin/env node
const karate = require('@karatelabs/karate');
karate.exec();
```

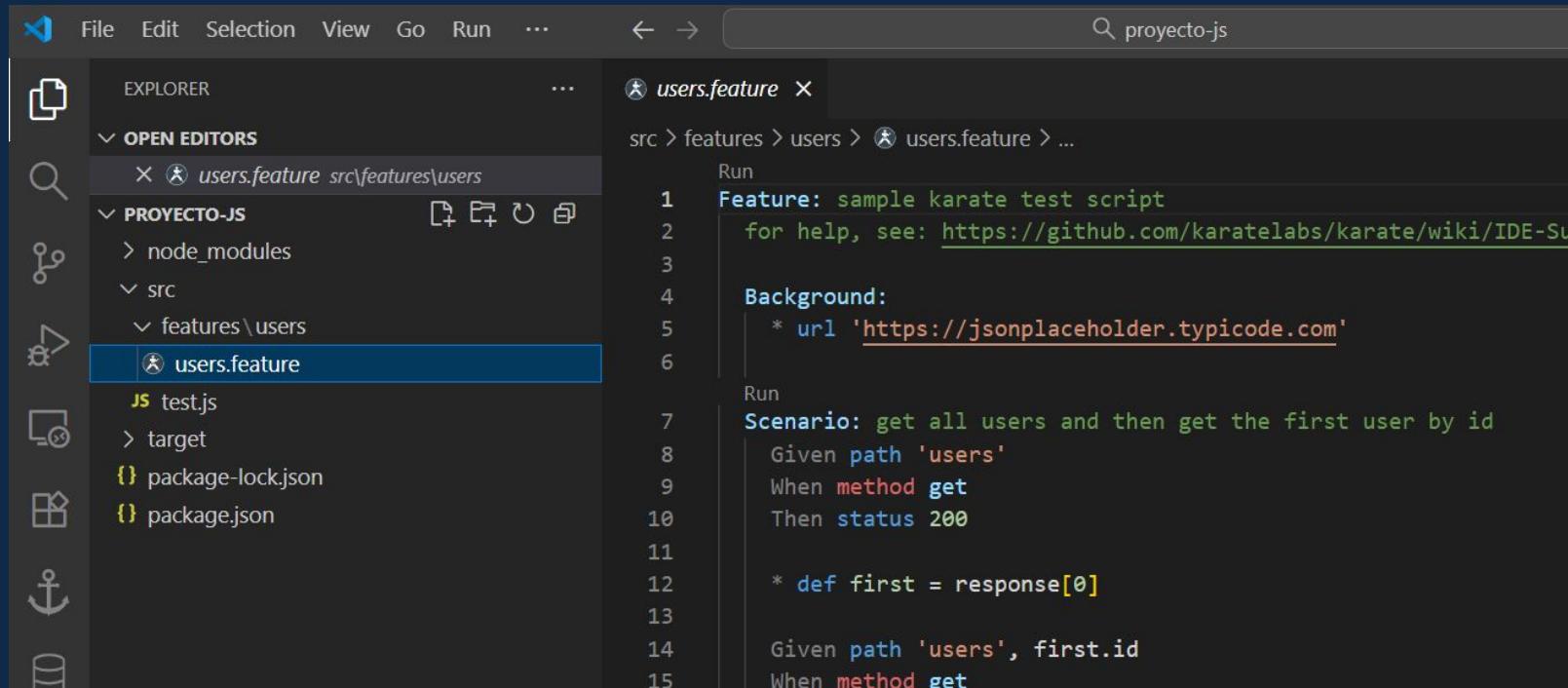
Below the usage section, it says "And in `package.json` (use the latest version from [npm](#)):". To the right of the usage section, there are sections for "Install" (with a command line icon and the command `npm i @karatelabs/karate`), "Repository" (with a GitHub icon and the URL `github.com/karatelabs/karate-npm`), "Homepage" (with a GitHub icon and the URL `github.com/karatelabs/karate-npm`), and "Weekly Downloads" (with a purple line graph showing the number 809).

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

Se guarda en el package.json la dependencia de Karate

```
{ } package.json ×  
{ } package.json > ...  
1  {  
2    "name": "proyecto",  
3    "version": "1.0.0",  
4    "description": "",  
5    "main": "index.js",  
6    "scripts": {  
7      "test": "node ./src/test.js"  
8    },  
9    "keywords": [],  
10   "author": "",  
11   "license": "ISC",  
12   "dependencies": {  
13     "@karatelabs/karate": "^0.3.2"  
14   }  
15 }  
16 }
```

Agregamos un archivo de test de karate (.feature)



The image shows a screenshot of a code editor, likely Visual Studio Code, with a dark theme. The left sidebar (Explorer) shows a file structure for a project named 'PROYECTO-JS'. Inside 'src/features/users', there is a file named 'users.feature'. This file is currently selected and highlighted with a blue bar at the bottom of the sidebar. The main editor area displays the content of 'users.feature'.

```
Feature: sample karate test script
for help, see: https://github.com/karatelabs/karate/wiki/IDE-Support

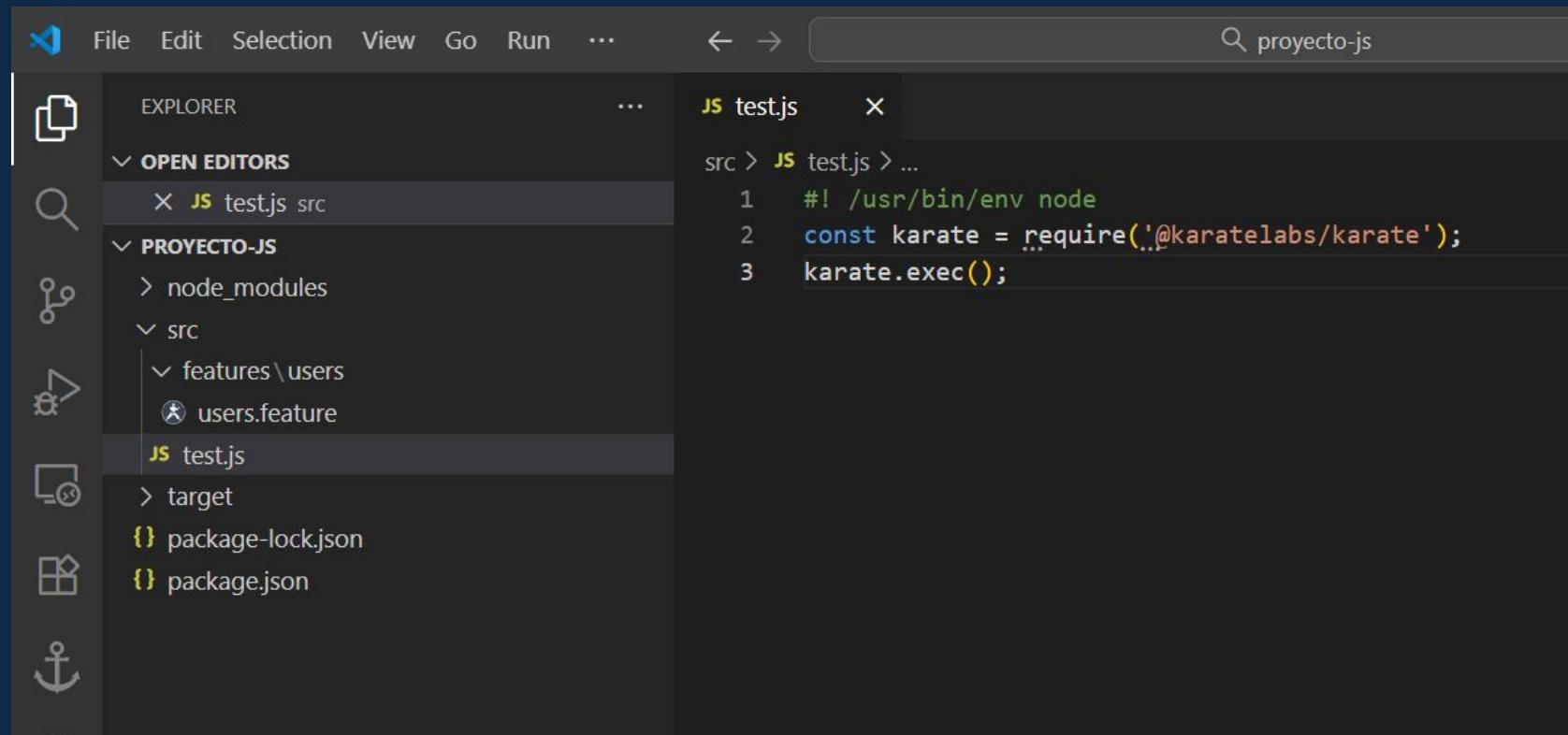
Background:
* url 'https://jsonplaceholder.typicode.com'

Scenario: get all users and then get the first user by id
Given path 'users'
When method get
Then status 200

* def first = response[0]

Given path 'users', first.id
When method get
```

Agregamos un archivo de test de javascript



File Edit Selection View Go Run ... ← → 🔍 proyecto-js

EXPLORER

OPEN EDITORS

PROYECTO-JS

- node_modules
- src
 - features\users
 - users.feature
 - test.js
- target
- package-lock.json
- package.json

test.js

```
src > JS test.js > ...
1  #! /usr/bin/env node
2  const karate = require('@karatelabs/karate');
3  karate.exec();
```

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

JBang

Lets Students, Educators and Professional Developers create, edit and run self-contained source-only Java programs with unprecedented ease.

[Download](#)[Try](#)

LATEST BLOG POST

Java in Places You Do Not Expect It

October 17, 2025



```
● ● ●
//DEPS org.slf4j:slf4j-simple:1.7.25
//DEPS org.kohsuke:github-api:1.116
import org.kohsuke.github.*;
```

Embedded Dependencies

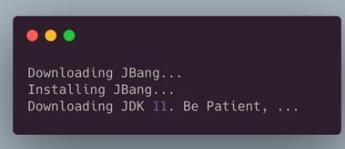
Automatic fetching of any dependency using `//DEPS`
Annotations or `@JBang` annotations directly from the



```
● ● ●
$ sdk install jbang
$ brew install jbangdev/tap/jbang
C:/> choco install jbang
```

Install & Run Anywhere

JBang installs and runs on Windows, Linux, macOS, Docker and
GitHub Actions as well as usable from Maven and Gradle plugins



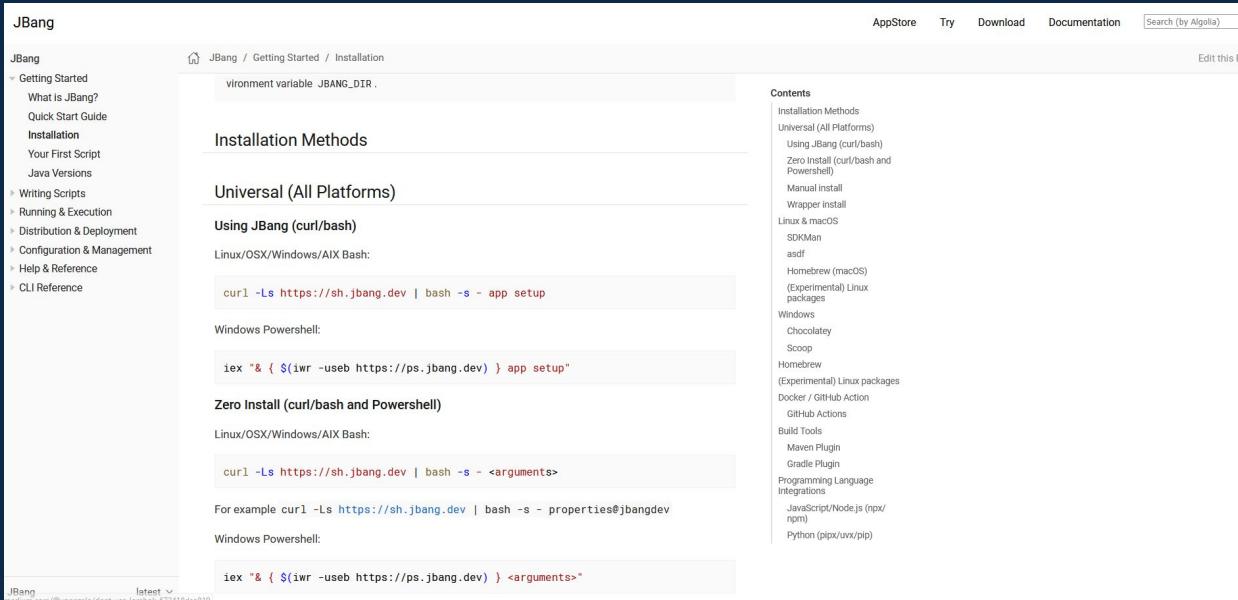
```
● ● ●
Downloading JBang...
Installing JBang...
Downloading JDK 11. Be Patient, ...
```

No Java ? No Problem!

Java will automatically be downloaded when needed.

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

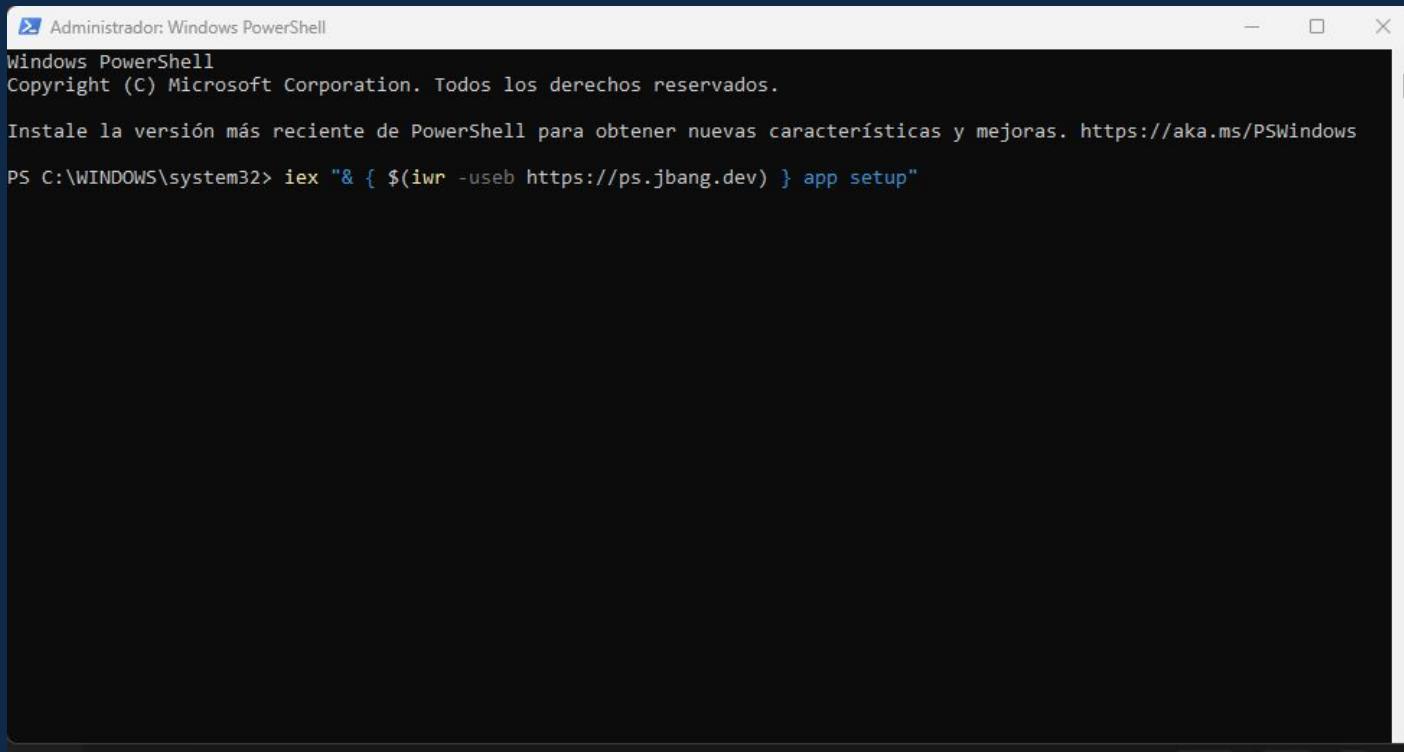


The screenshot shows the JBang documentation website. The left sidebar has a 'JBang' header and a 'Getting Started' section with links to 'What Is JBang?', 'Quick Start Guide', 'Installation', 'Your First Script', 'Java Versions', 'Writing Scripts', 'Running & Execution', 'Distribution & Deployment', 'Configuration & Management', 'Help & Reference', and 'CLI Reference'. The main content area is titled 'JBang / Getting Started / Installation'. It contains a 'vironment variable JBANG_DIR.' section, an 'Installation Methods' section with a 'Universal (All Platforms)' heading, and sections for 'Using JBang (curl/bash)', 'Windows Powershell', 'Zero Install (curl/bash and Powershell)', and 'Windows'. The right sidebar has a 'Contents' section with links to various installation methods and platforms, including 'Universal (All Platforms)', 'Linux & macOS', 'Windows', and 'Programming Language Integrations'.

Link:

<https://www.jbang.dev/documentation/jbang/latest/installation.html>

En el PowerShell de Windows instalamos JBang



Administrator: Windows PowerShell

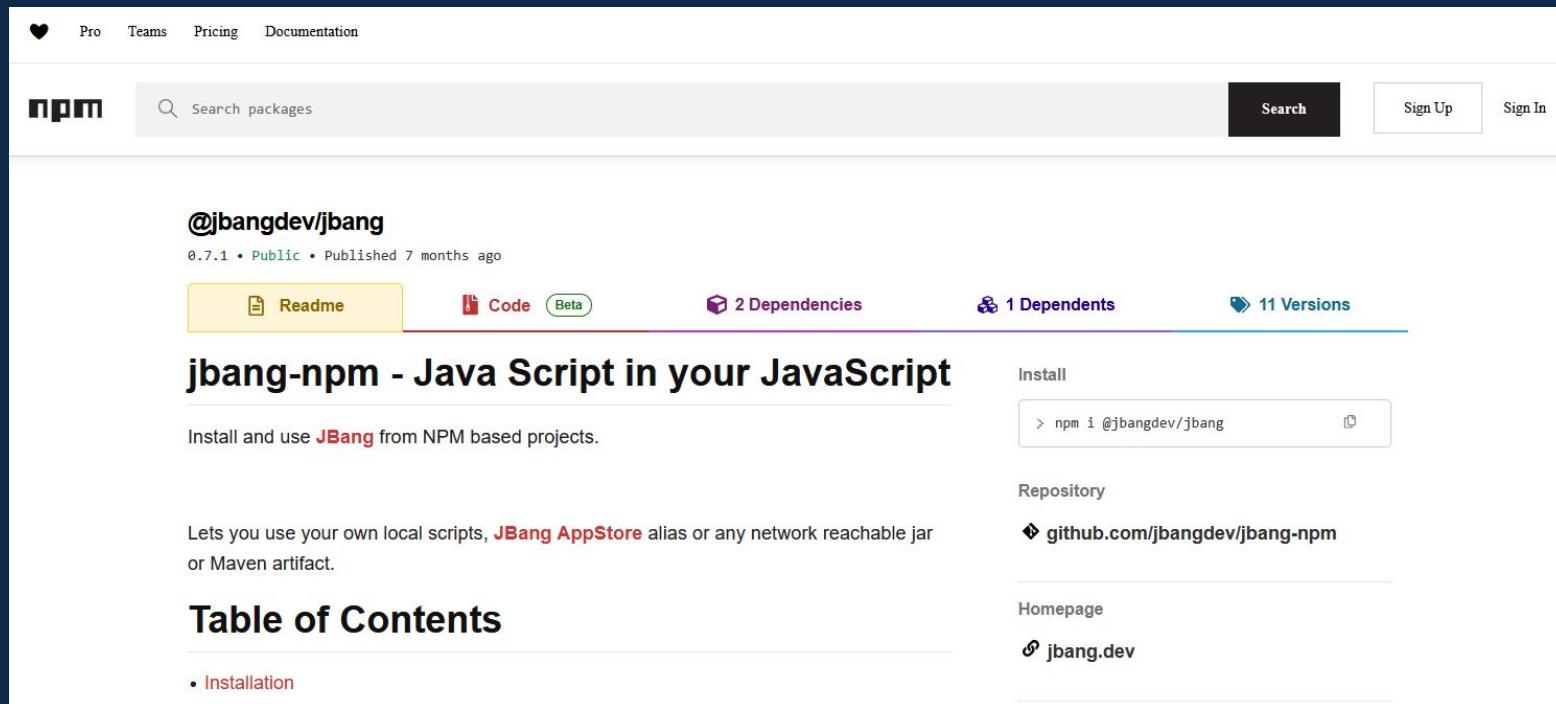
Windows PowerShell

Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Instale la versión más reciente de PowerShell para obtener nuevas características y mejoras. <https://aka.ms/PSWindows>

```
PS C:\WINDOWS\system32> iex "& { $(iwr -useb https://ps.jbang.dev) } app setup"
```

Otra opción para instalar es desde la página de npm

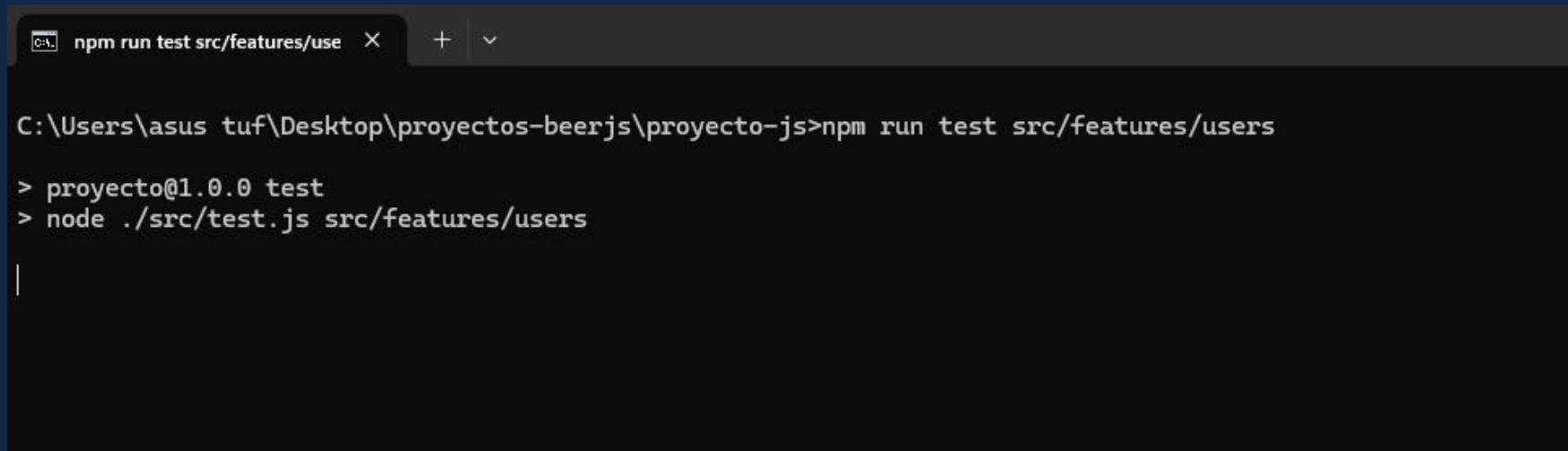


The screenshot shows the npmjs.com package page for the package `@jbangdev/jbang`. The page includes the following elements:

- Header:** A navigation bar with links for **Pro**, **Teams**, **Pricing**, and **Documentation**. It also features a heart icon for favoriting, a search bar with the placeholder `Search packages`, and buttons for **Search**, **Sign Up**, and **Sign In**.
- Package Information:** The package name `@jbangdev/jbang` is displayed, along with its version `0.7.1`, `Public` status, and `Published 7 months ago`.
- Navigation Tabs:** A horizontal bar with tabs for **Readme** (highlighted in yellow), **Code** (Beta), **2 Dependencies**, **1 Dependents**, and **11 Versions**.
- Section Headers:** **jbang-npm - Java Script in your JavaScript**, **Install**, **Repository**, **Homepage**.
- Install Section:** A box containing the command `> npm i @jbangdev/jbang` with a copy icon.
- Repository Section:** A link to the GitHub repository `github.com/jbangdev/jbang-npm`.
- Homepage Section:** A link to the official homepage `jbang.dev`.
- Table of Contents:** A section titled **Table of Contents** with a single item: **Installation**.

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

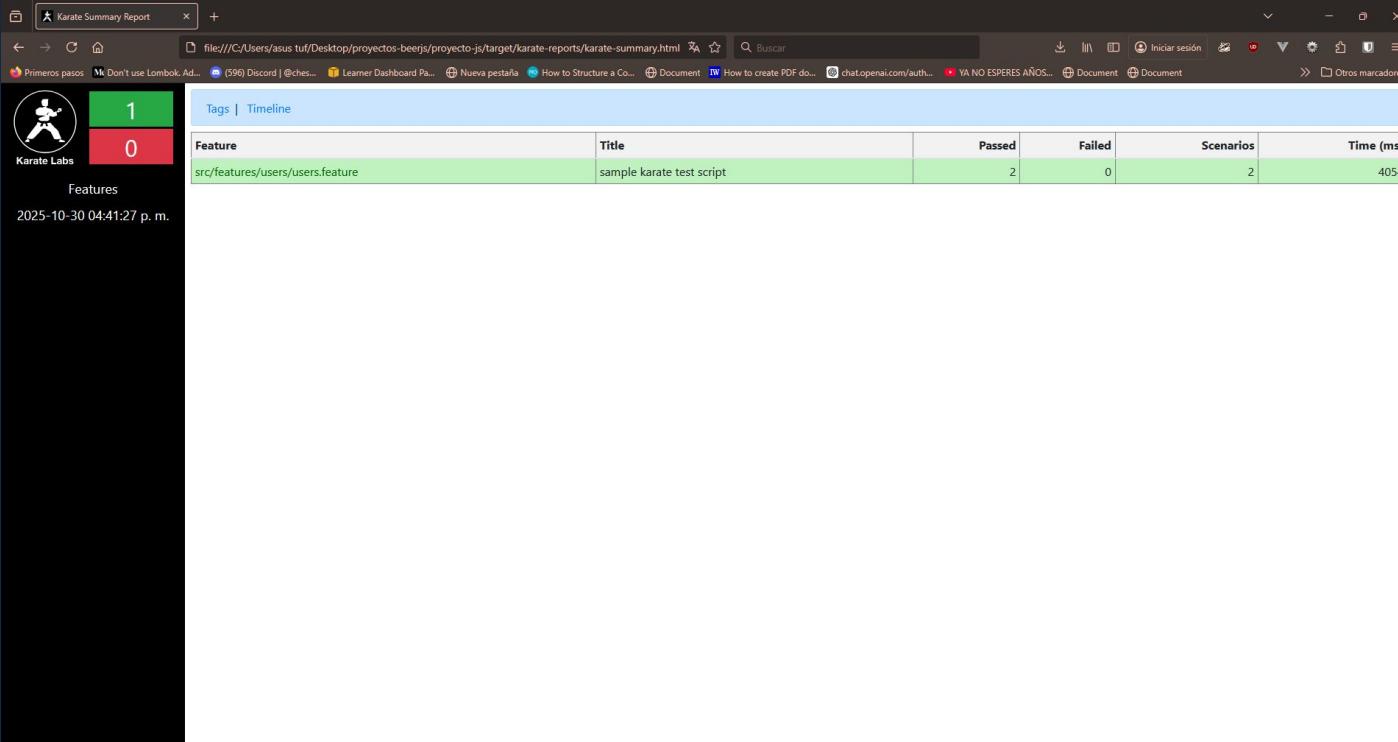
Por último ejecutamos el test desde la terminal de Linux o Windows



A screenshot of a terminal window with a dark background. The title bar shows the command: "npm run test src/features/use". The main area of the terminal displays the following text:

```
C:\Users\asus tuf\Desktop\proyectos-beerjs\proyecto-js>npm run test src/features/users
> proyecto@1.0.0 test
> node ./src/test.js src/features/users
|
```

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



Karate Summary Report

file:///C:/Users/asus tuf/Desktop/proyectos-beerjs/proyecto-js/target/karate-reports/karate-summary.html

Primeros pasos, MK Don't use Lombok, Ad..., (596) Discord @ches..., Learner Dashboard Pa..., Nueva pestaña, How to Structure a Co..., Document, How to create PDF do..., chat.openai.com/auth..., YA NO ESPERES AÑOS..., Document, Document, Otros marcadores

Tags | Timeline

Feature	Title	Passed	Failed	Scenarios	Time (ms)
src/features/users/users.feature	sample karate test script	2	0	2	4054

Karate Labs

Features

2025-10-30 04:41:27 p. m.

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



{ api }

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

The screenshot shows the homepage of the RapidAPI website. At the top, there is a navigation bar with the Rapid logo, 'Enterprises', and 'About Us' on the left, and 'API Hub' with a search icon and user profile on the right. The main headline is 'Your dream API is on the market', followed by a subtext: 'Previously known as Rapid, Nokia API Hub is more than just a catalog for APIs. Find the perfect API using our discovery features or publish your own APIs and drive new API revenue streams.' Below this is a 'API hub' button. To the right, there are two images: one showing a search interface and another showing a grid of API categories like 'Finance', 'Sports', 'Weather', 'Transportation', 'Entertainment', 'Business', 'Technology', and 'Science'. The central part of the page features the text 'World's largest public API Hub' above three statistics: '7M+' developers, '82K+' APIs in the Hub, and '9B+' API calls per month. At the bottom, there are two sections: 'Consume APIs' with a code snippet for Node.js Axios and 'Provide APIs' with a screenshot of a mobile app interface.

Your dream API is on the market

Previously known as Rapid, Nokia API Hub is more than just a catalog for APIs. Find the perfect API using our discovery features or publish your own APIs and drive new API revenue streams.

API hub

World's largest public API Hub

7M+ Developers

82K+ APIs in the Hub

9B+ API calls per month

Consume APIs

```
[Node.js Axios] Copy Code
const axios = require('axios');

const options = {
  method: 'GET',
  url: 'https://rank01-mlb-live-in-game-real
  citions',
  headers: {
```

Provide APIs

Change Master
Provide organizational chart
Updated today at 9:51 PM

Raw Materials Schedule
Plan in time logistics scheduling

Powered by Cloudflare

Link: <https://rapidapi.com/>



← [Iddo Gino](#)
2015



2024



POSTMAN





¿¿¿Preguntas???

Muchas Gracias



[Sergio Gabriel Garzón](#)



[sergiog90arg](#)