

Automating Postman Tests With JMeter



Author: Rafael Alcocer Caldera

Introduction: Postman & JMeter

Date: 25-Aug-2023

Tabla de Contenido

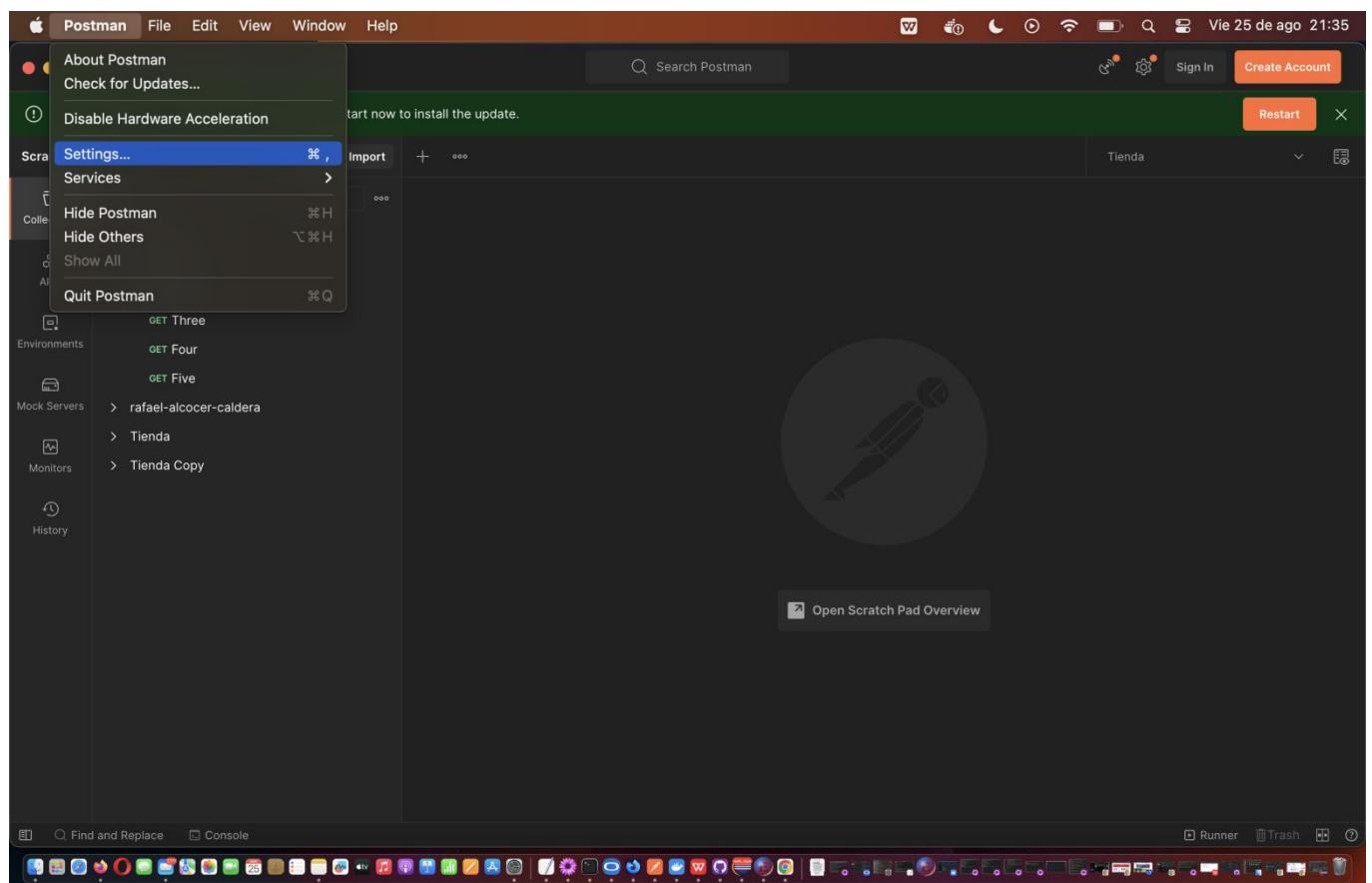
■ Introduction	3
■ Postman Settings	3
■ JMeter Configuration	7
■ Running the Tests	18

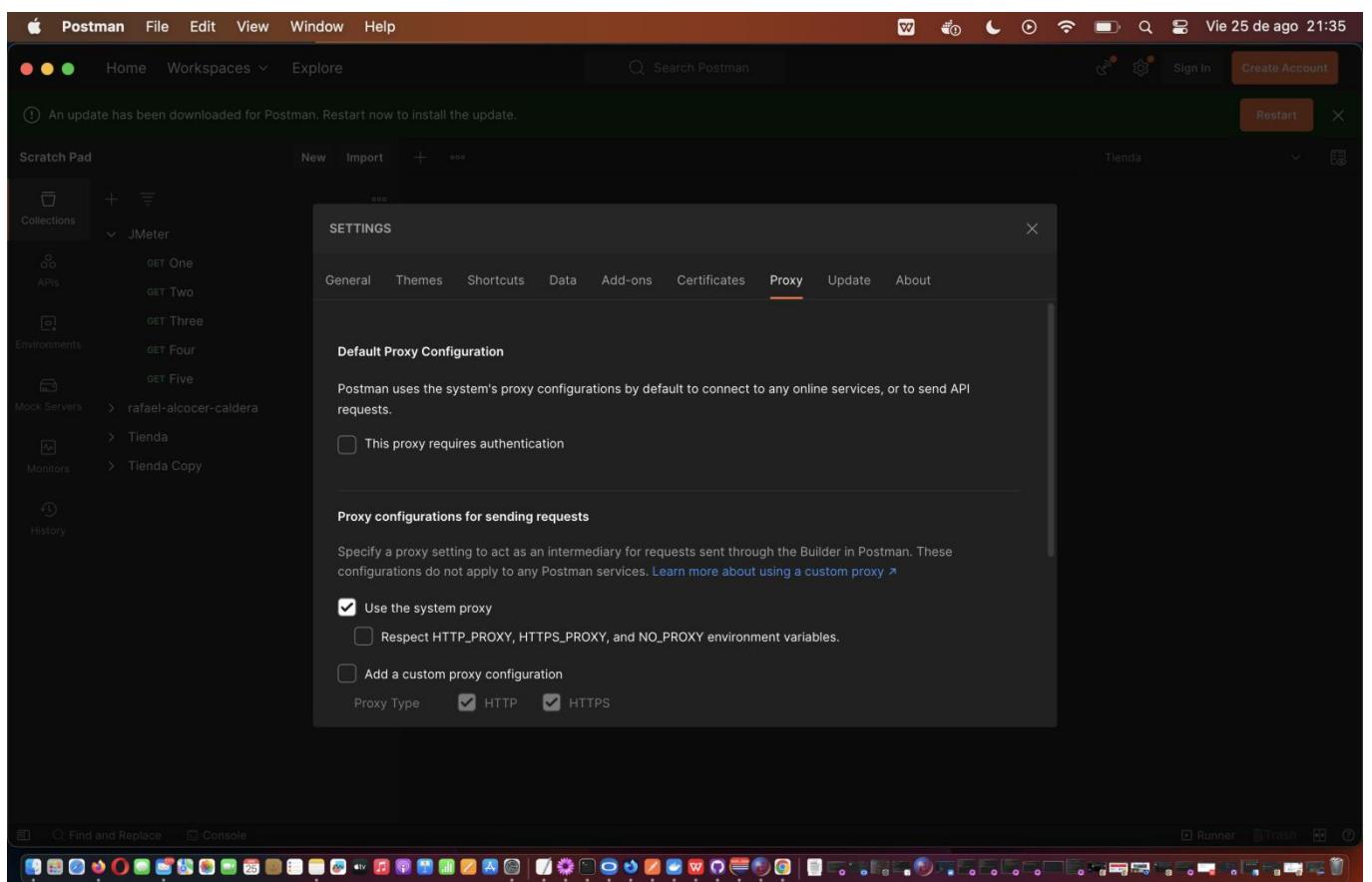
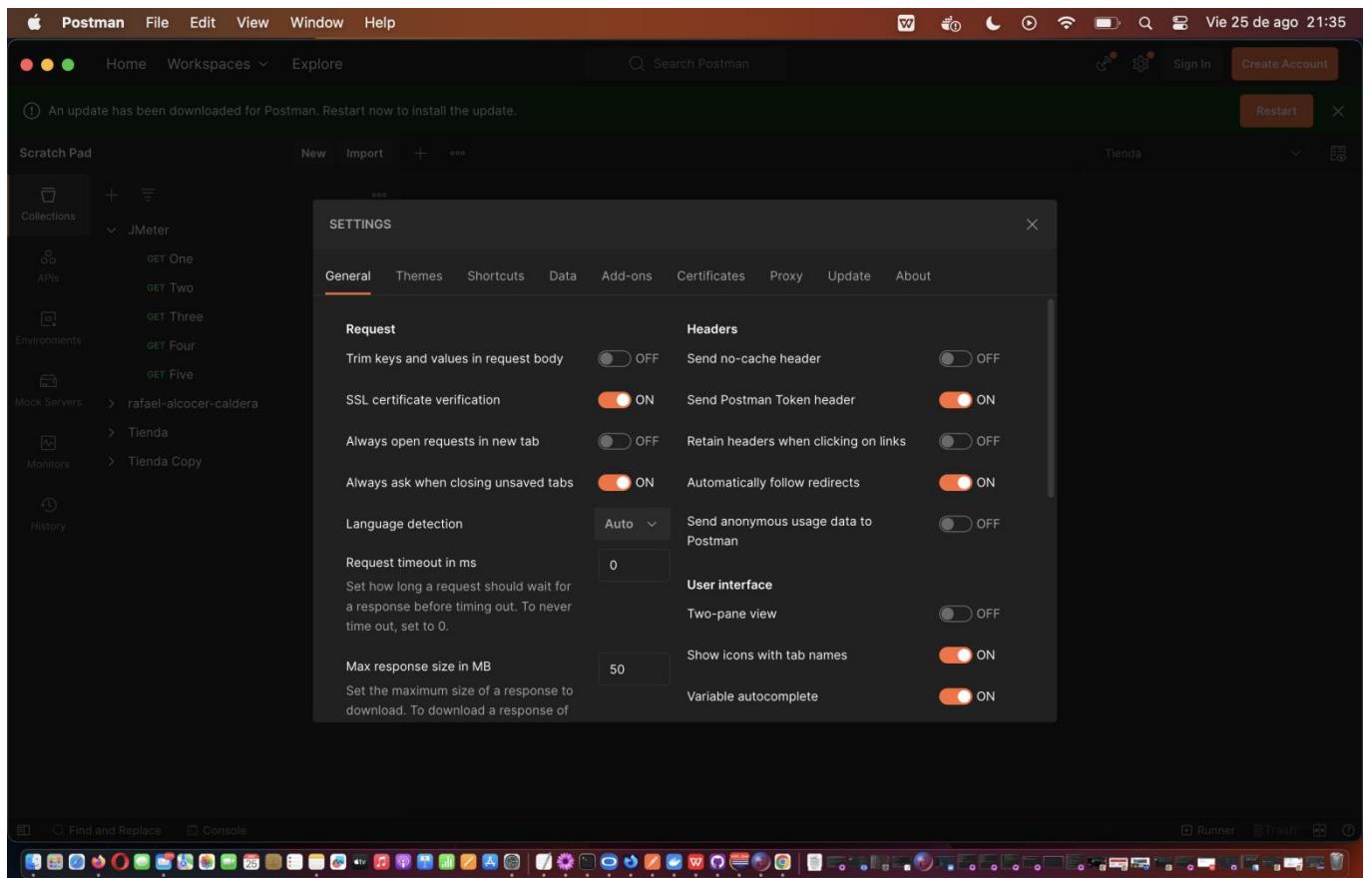
■ Introduction

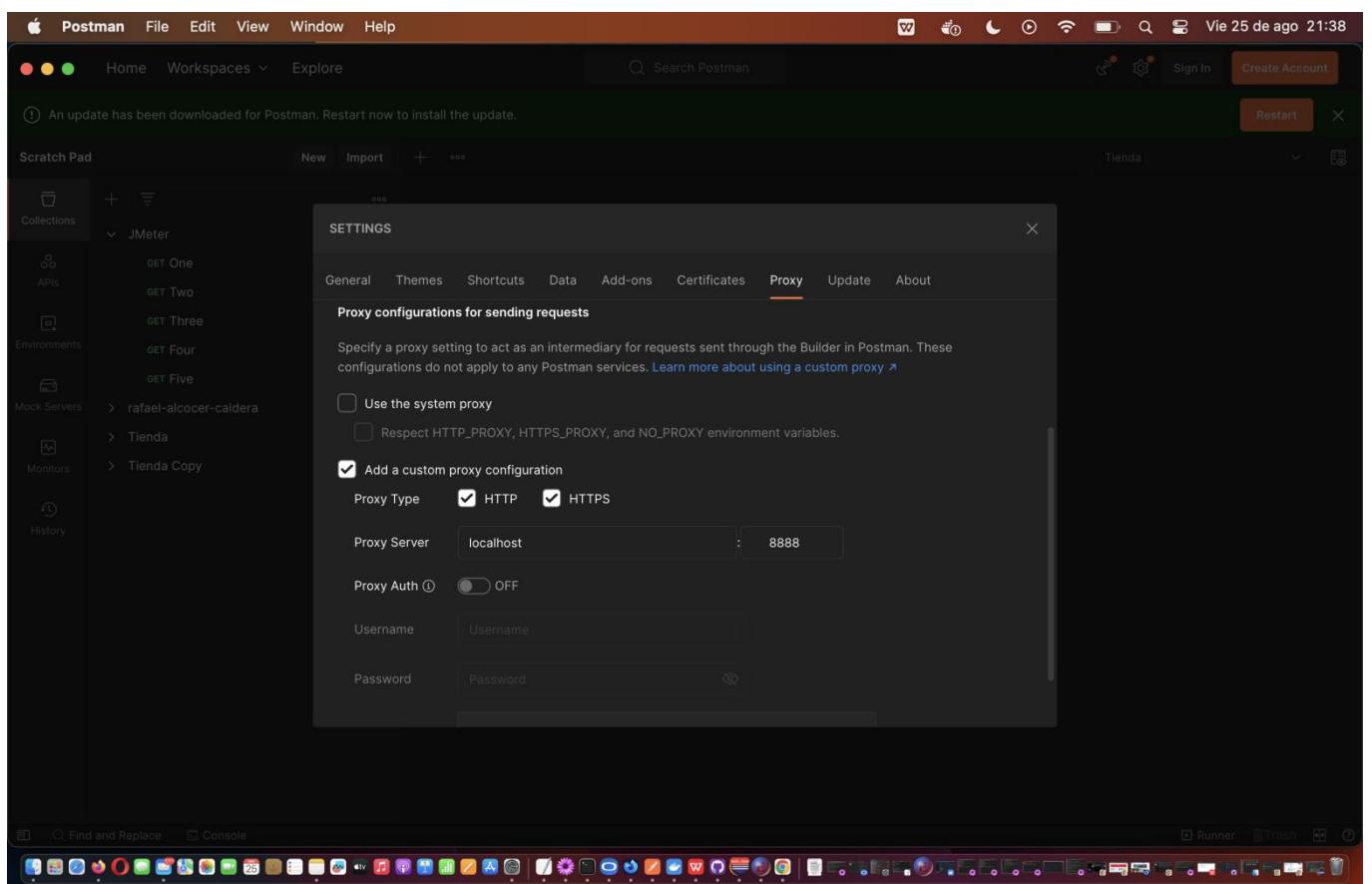
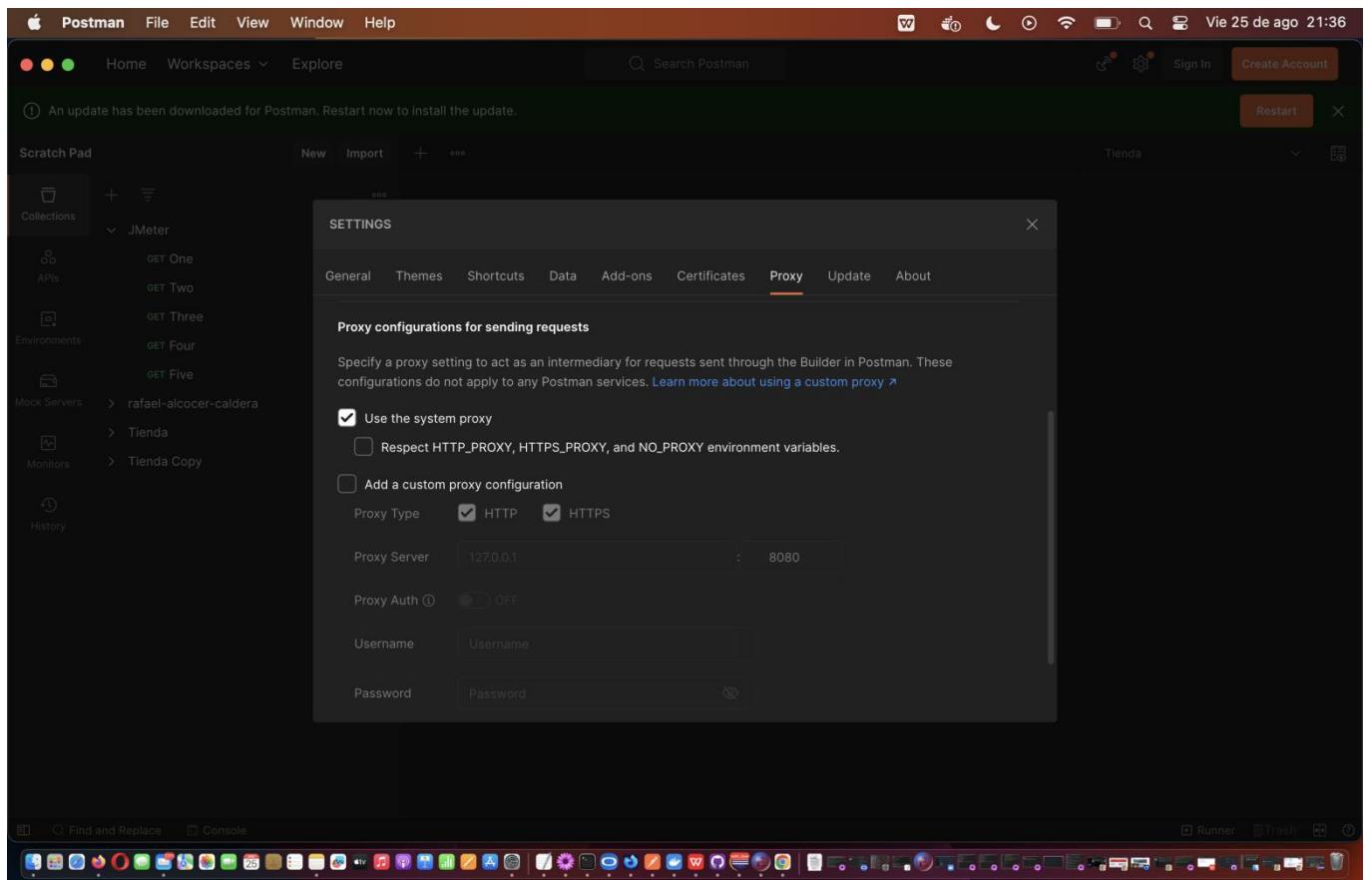
- The following was copied from:
 - English: <https://apiumhub.com/tech-blog-barcelona/automatizing-postman-test-jmeter/>
 - Spanish: <https://apiumhub.com/es/tech-blog-barcelona/automatizando-test-postman-jmeter/>

■ Postman Settings

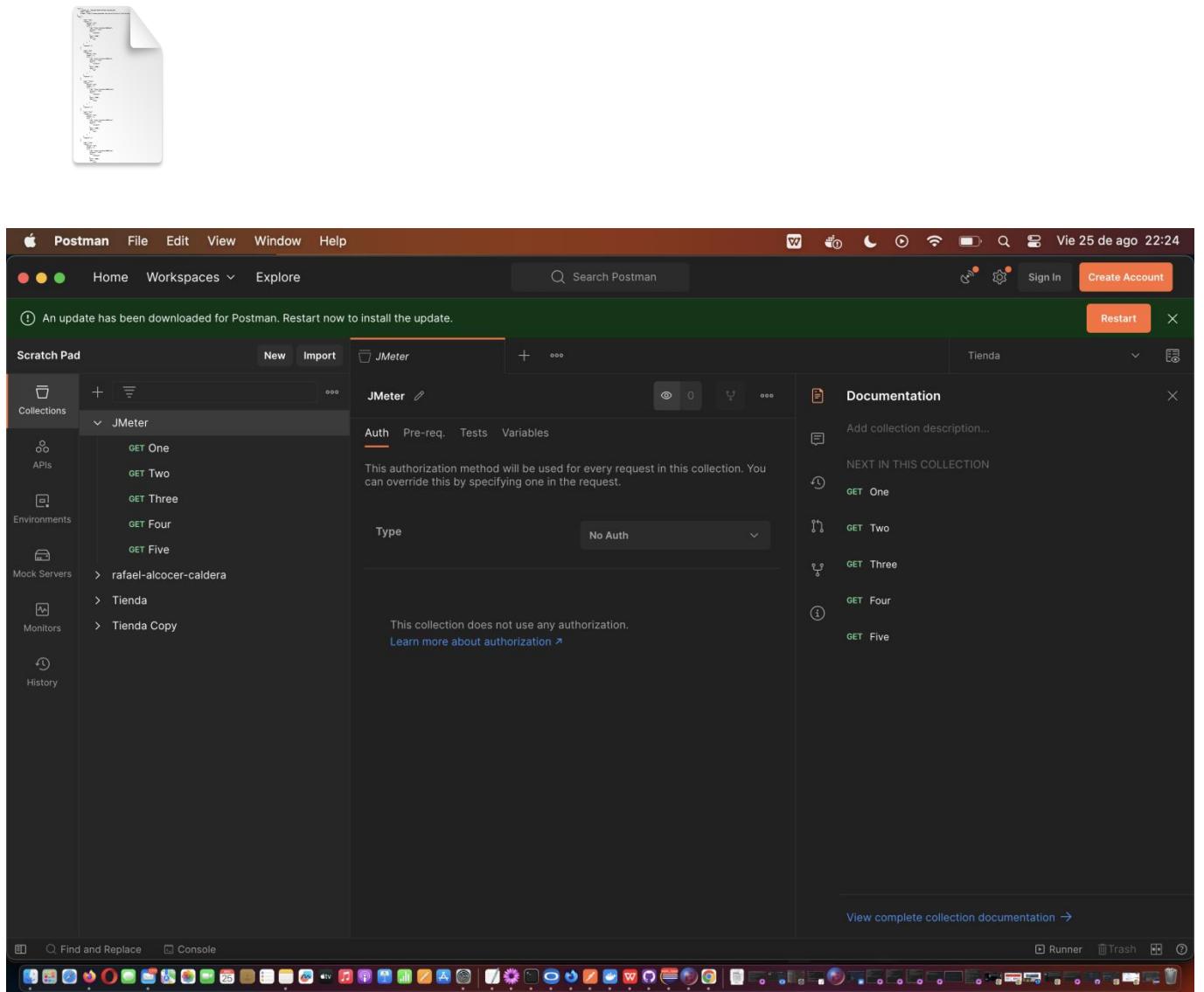
- Postman -> Settings -> Proxy -> Add a custom proxy configuration
 - Unselect -> Use the system proxy
 - Select -> Add a custom proxy configuration
 - Proxy Server: localhost
 - Port: 8888







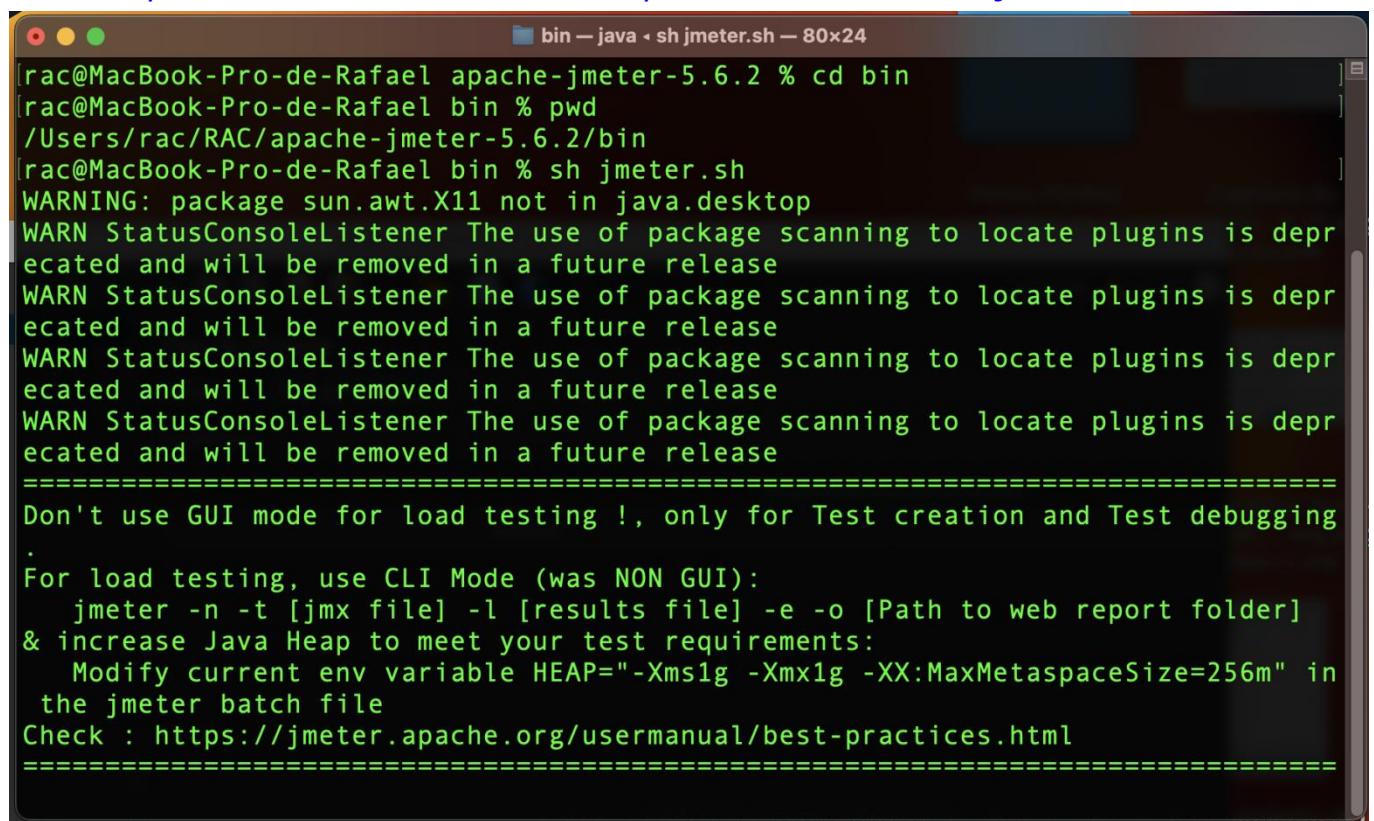
➤ Import the following collection:



■ JMeter Configuration

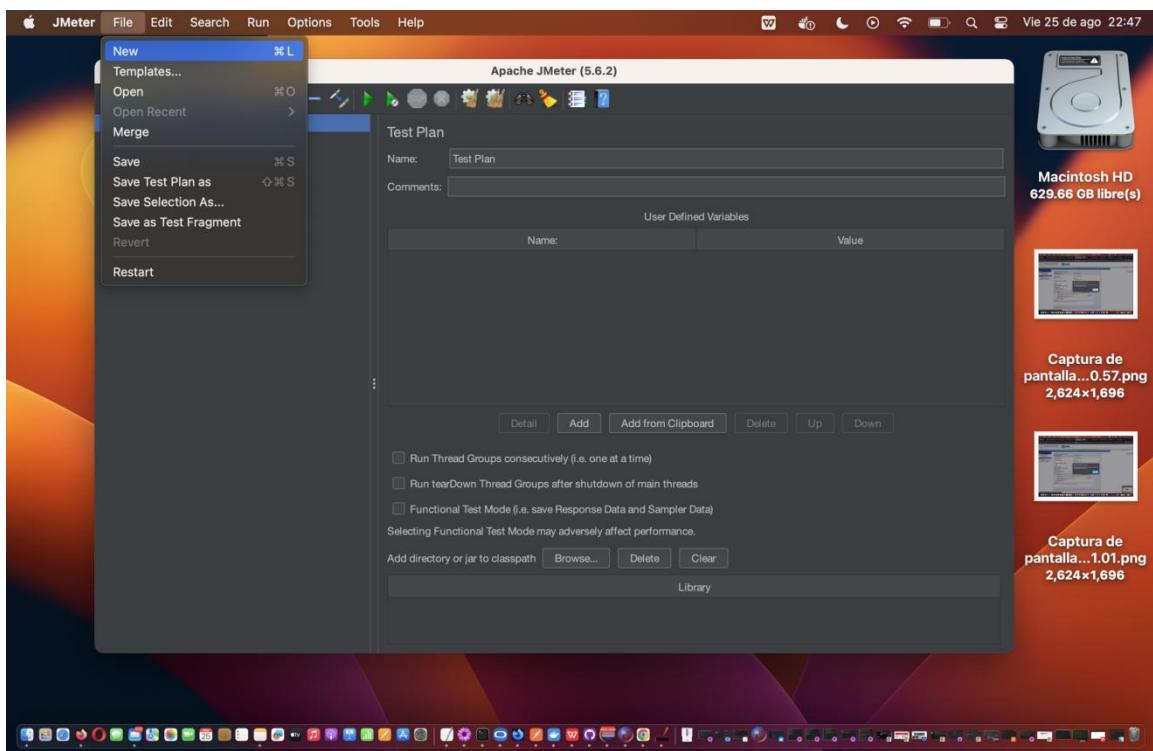
- Download JMeter:
 - https://jmeter.apache.org/download_jmeter.cgi
 - ◆ apache-jmeter-5.6.2.zip

- Run JMeter:
 - Unzip apache-jmeter-5.6.2.zip
 - Go to the path: /apache-jmeter-5.6.2/bin
 - Open a terminal with the above path and execute: sh jmeter.sh

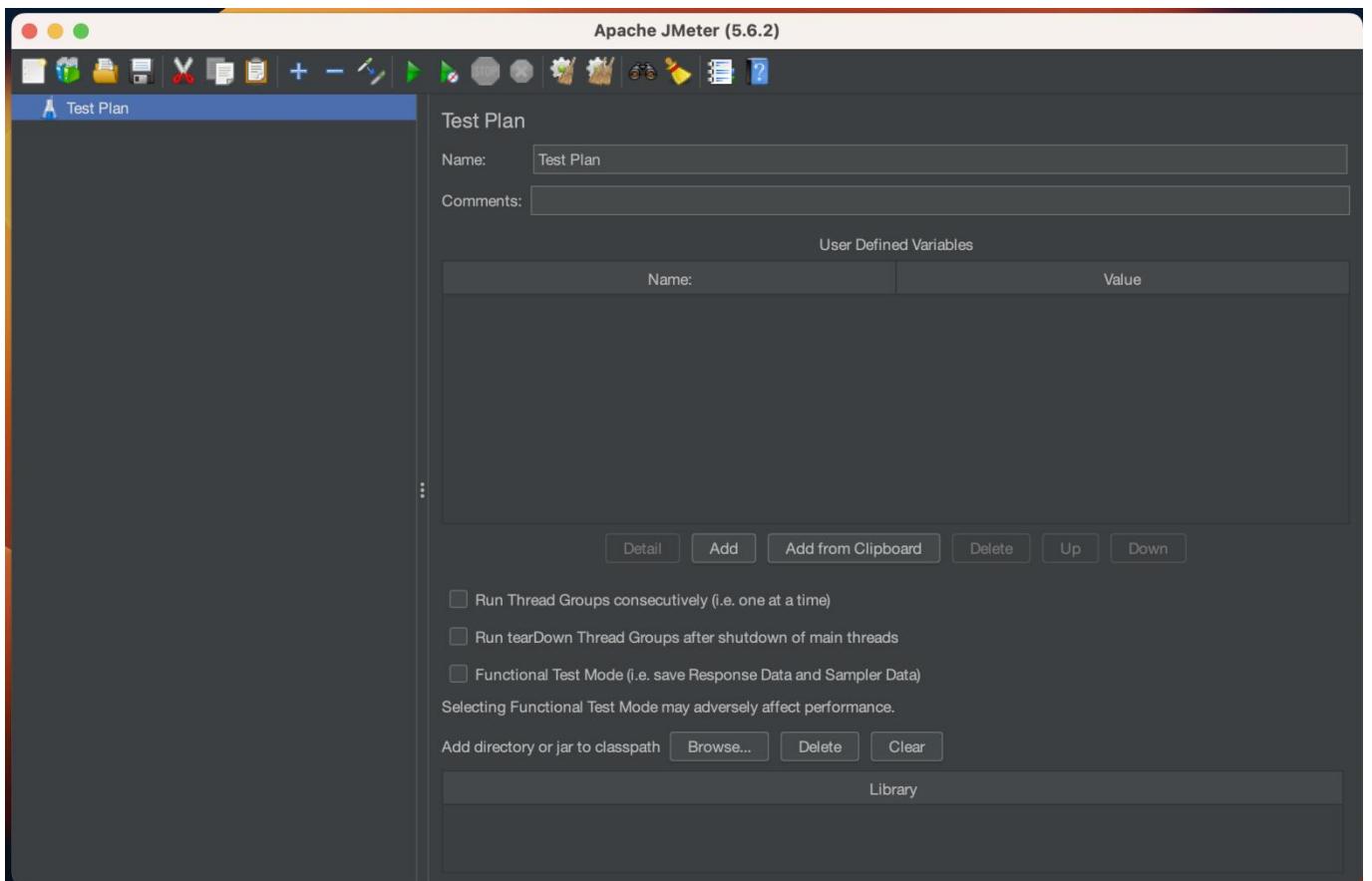


```

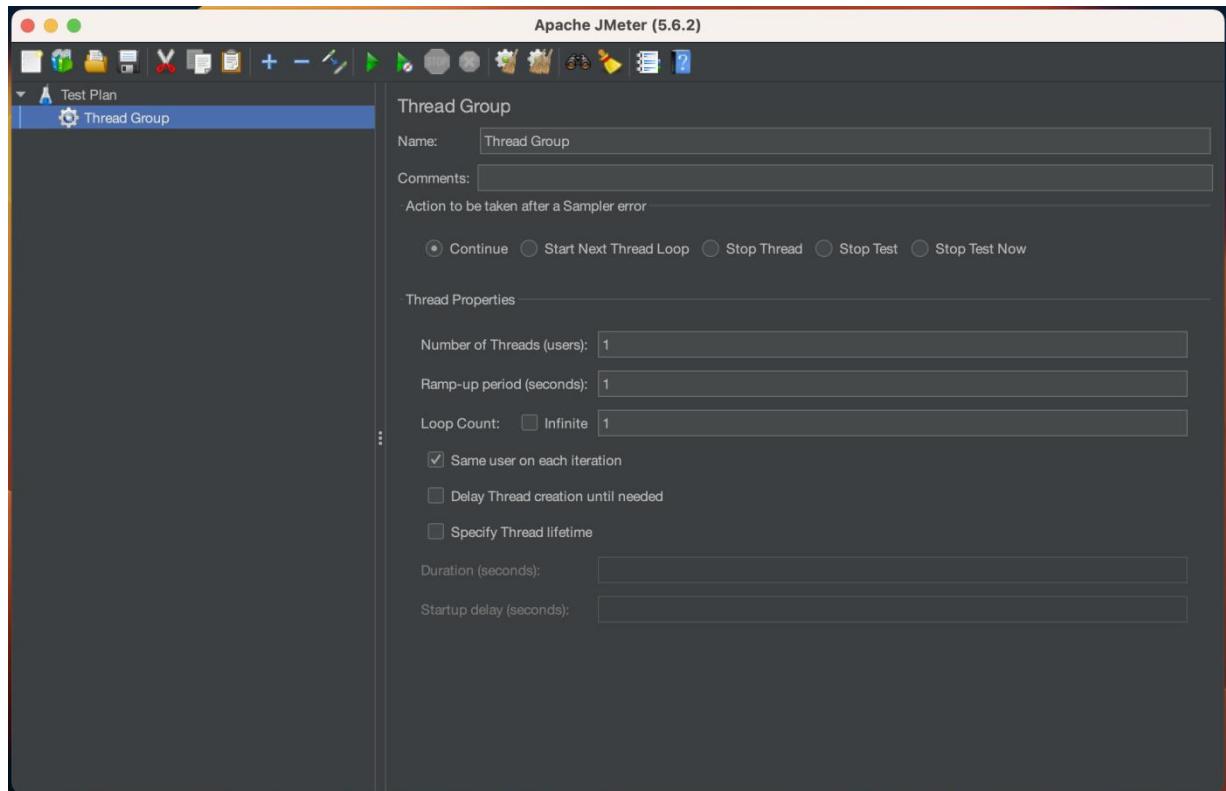
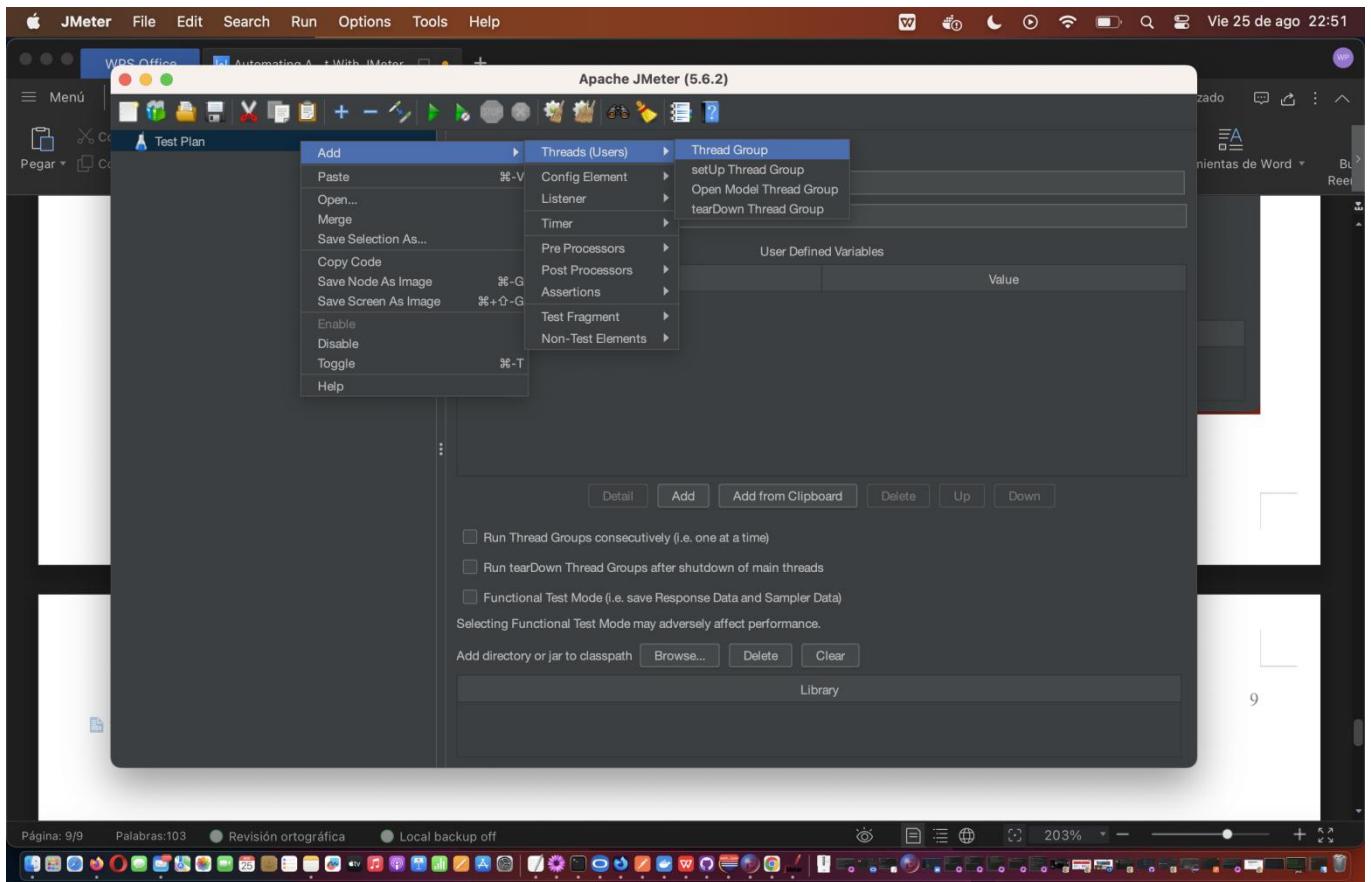
rac@MacBook-Pro-de-Rafael apache-jmeter-5.6.2 % cd bin
rac@MacBook-Pro-de-Rafael bin % pwd
/Users/rac/RAC/apache-jmeter-5.6.2/bin
rac@MacBook-Pro-de-Rafael bin % sh jmeter.sh
WARNING: package sun.awt.X11 not in java.desktop
WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release
WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release
WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release
WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release
=====
Don't use GUI mode for load testing !, only for Test creation and Test debugging
.
For load testing, use CLI Mode (was NON GUI):
  jmeter -n -t [jmx file] -l [results file] -e -o [Path to web report folder]
& increase Java Heap to meet your test requirements:
  Modify current env variable HEAP="-Xms1g -Xmx1g -XX:MaxMetaspaceSize=256m" in
  the jmeter batch file
Check : https://jmeter.apache.org/usermanual/best-practices.html
=====
```



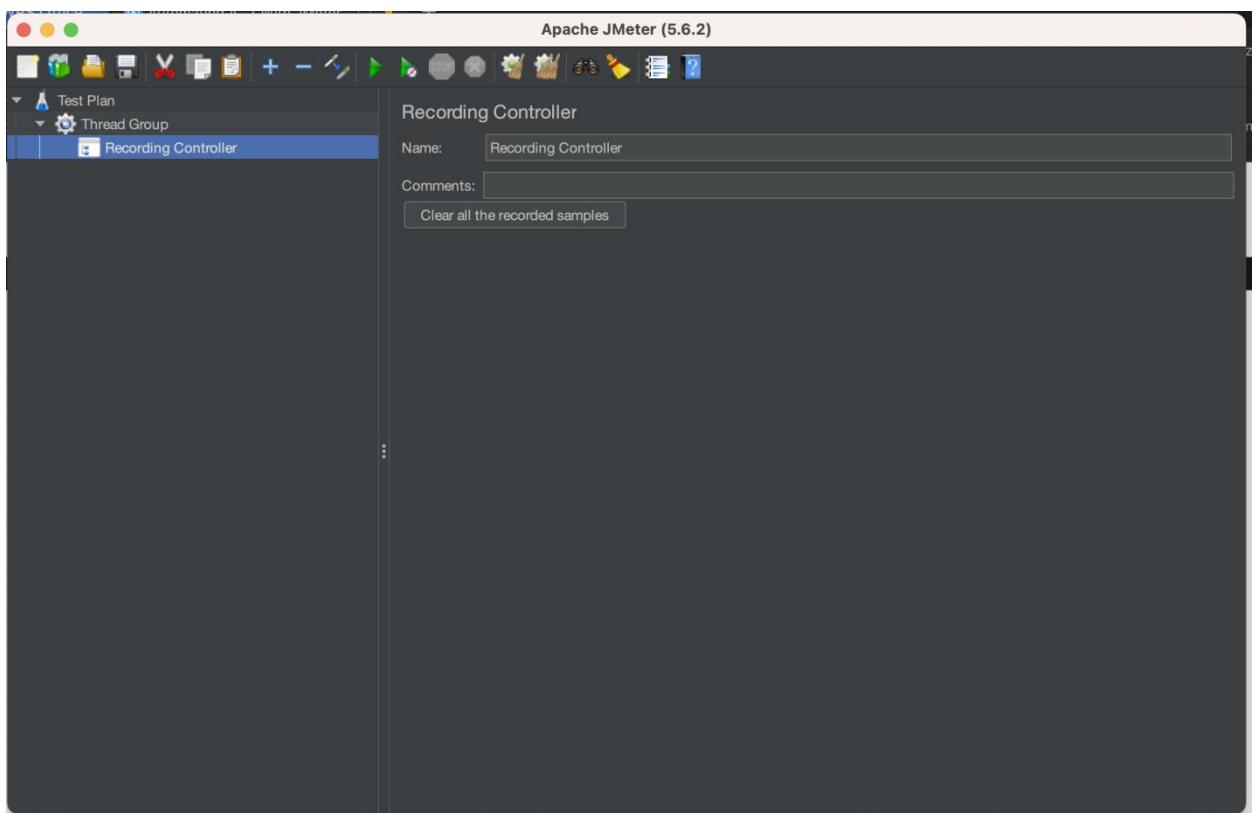
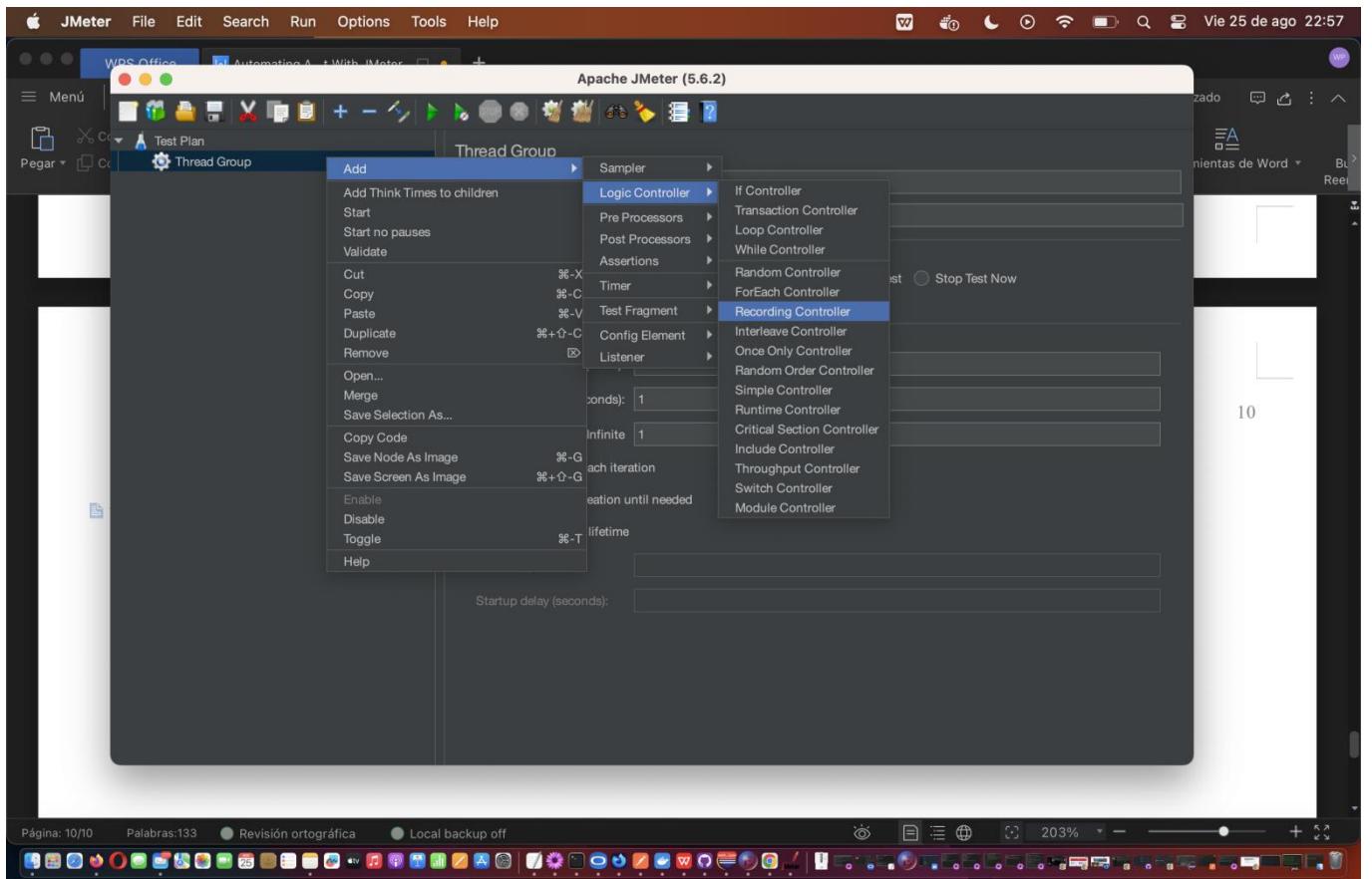
- Create an empty **Test Plan**:
 - File -> New (Archivo -> Nuevo)



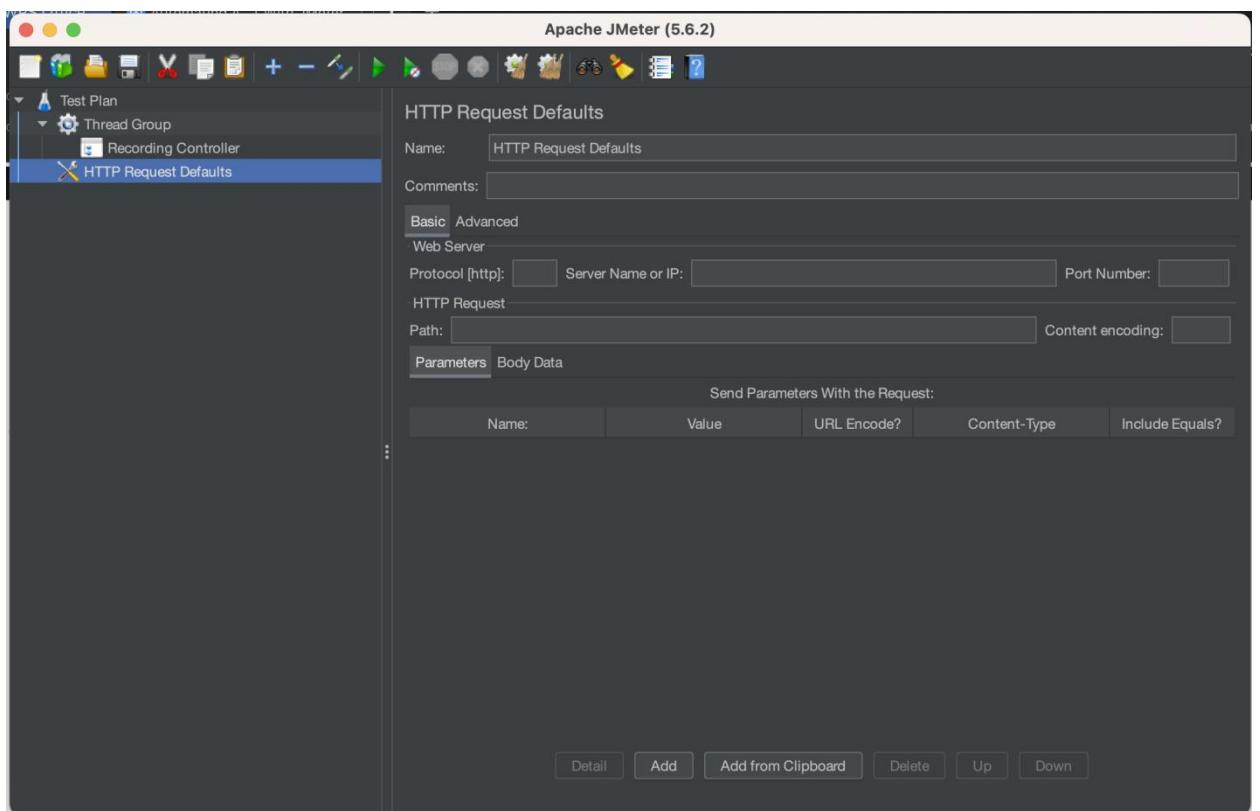
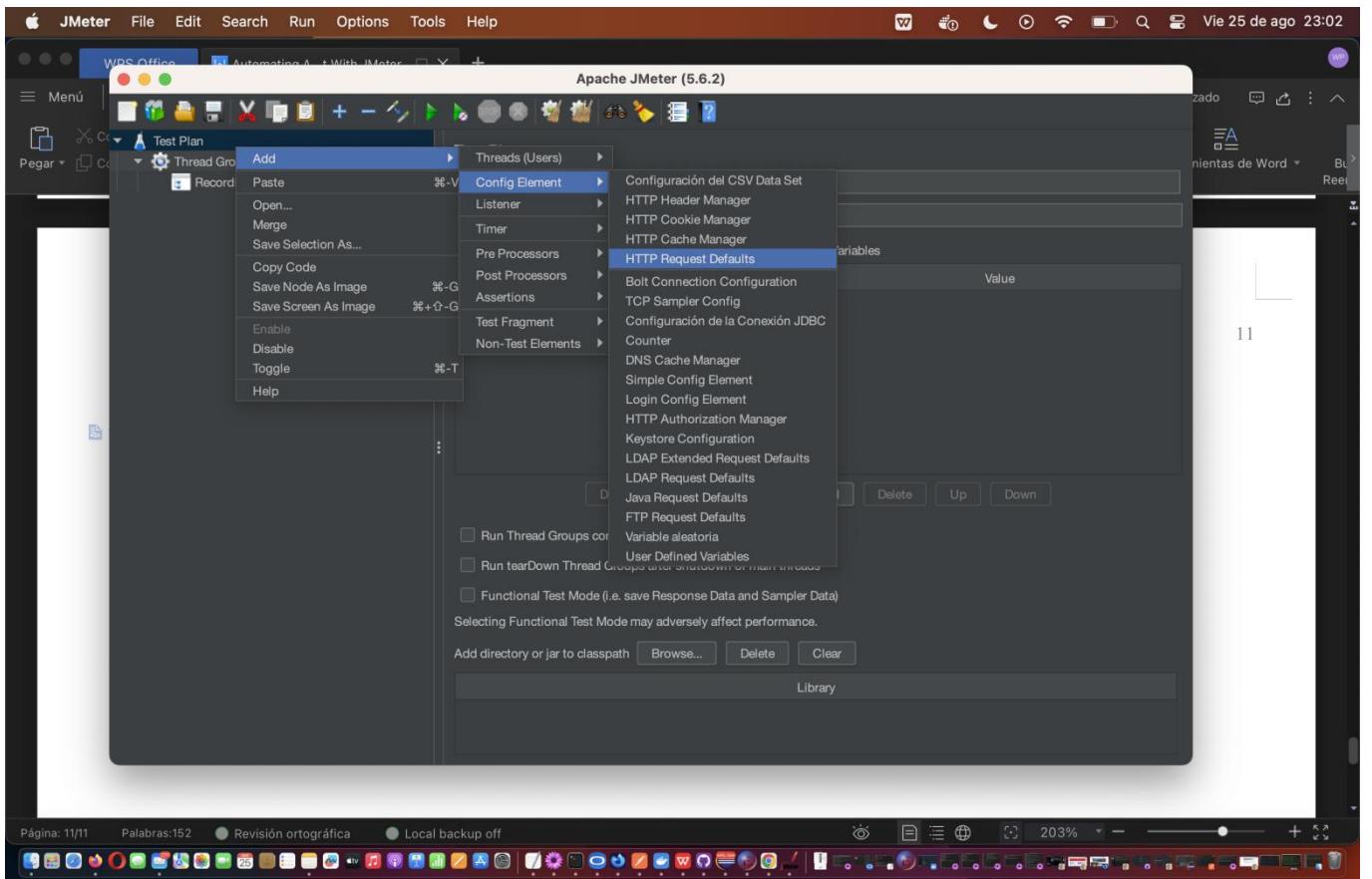
- Add a **Thread Group**, click right button over Test Plan:
 - Add -> Thread (Users) -> Thread Group



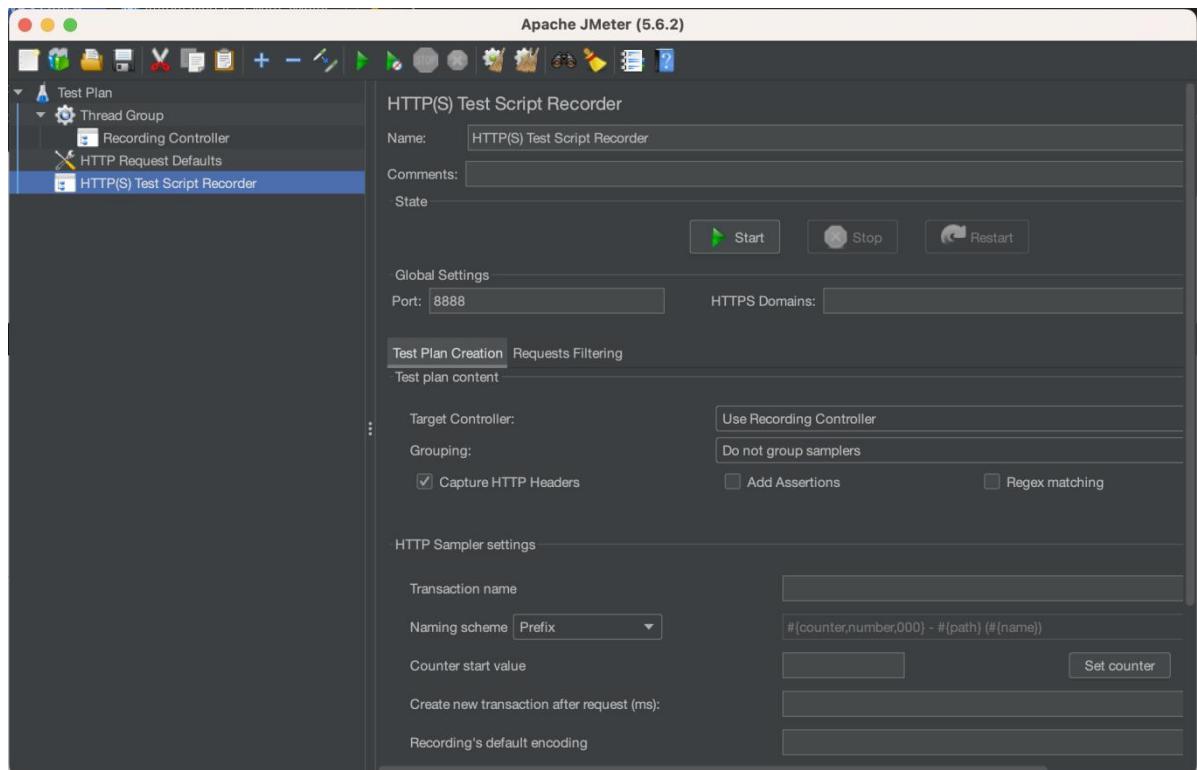
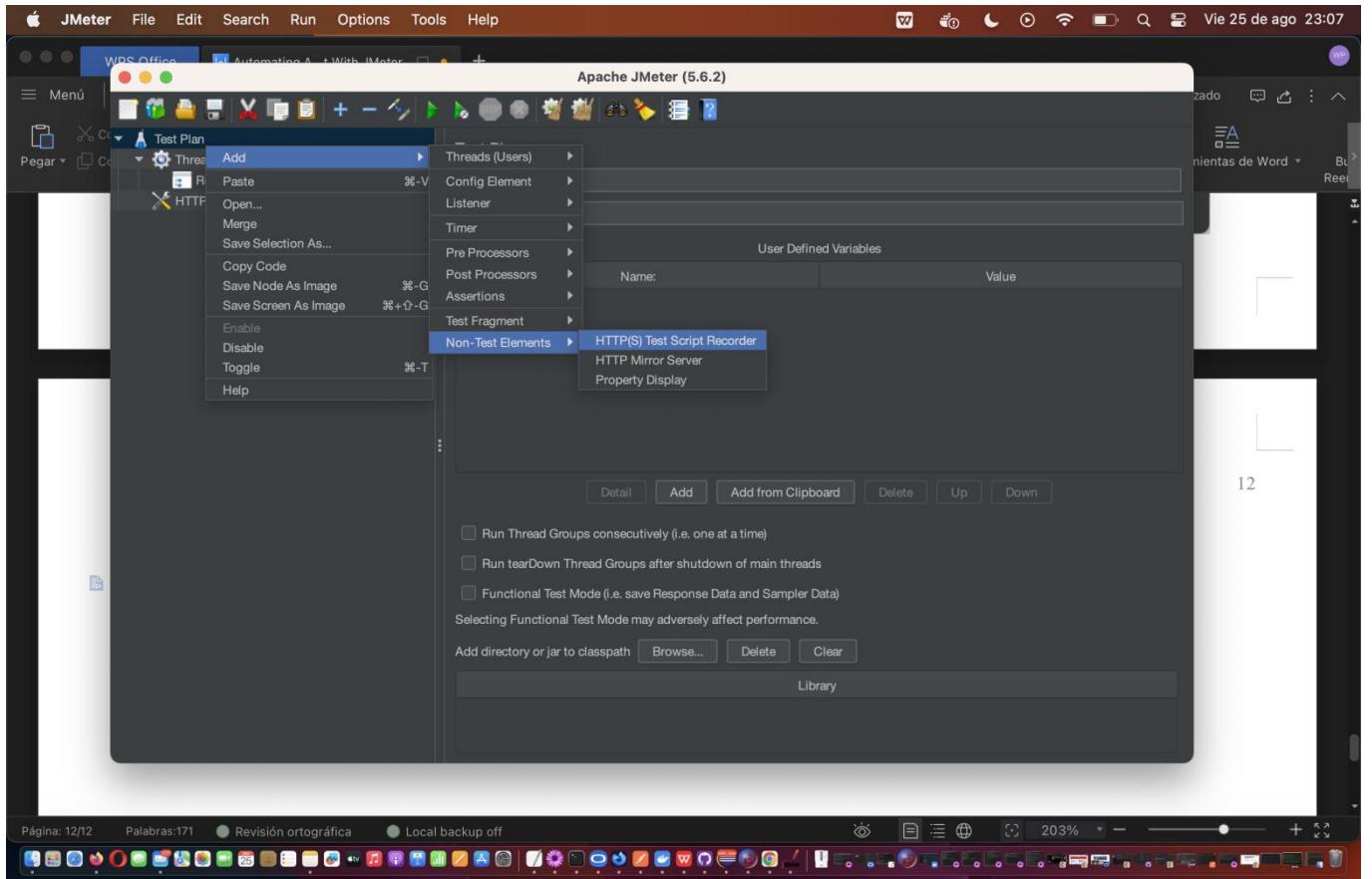
- Add a **Recording Controller**, click right button over Thread Group:
■ Add -> Logic Controller -> Recording Controller



- Add an **HTTP Request Defaults**, click right button over Test Plan:
 ■ Add -> Config Element -> HTTP Request Defaults

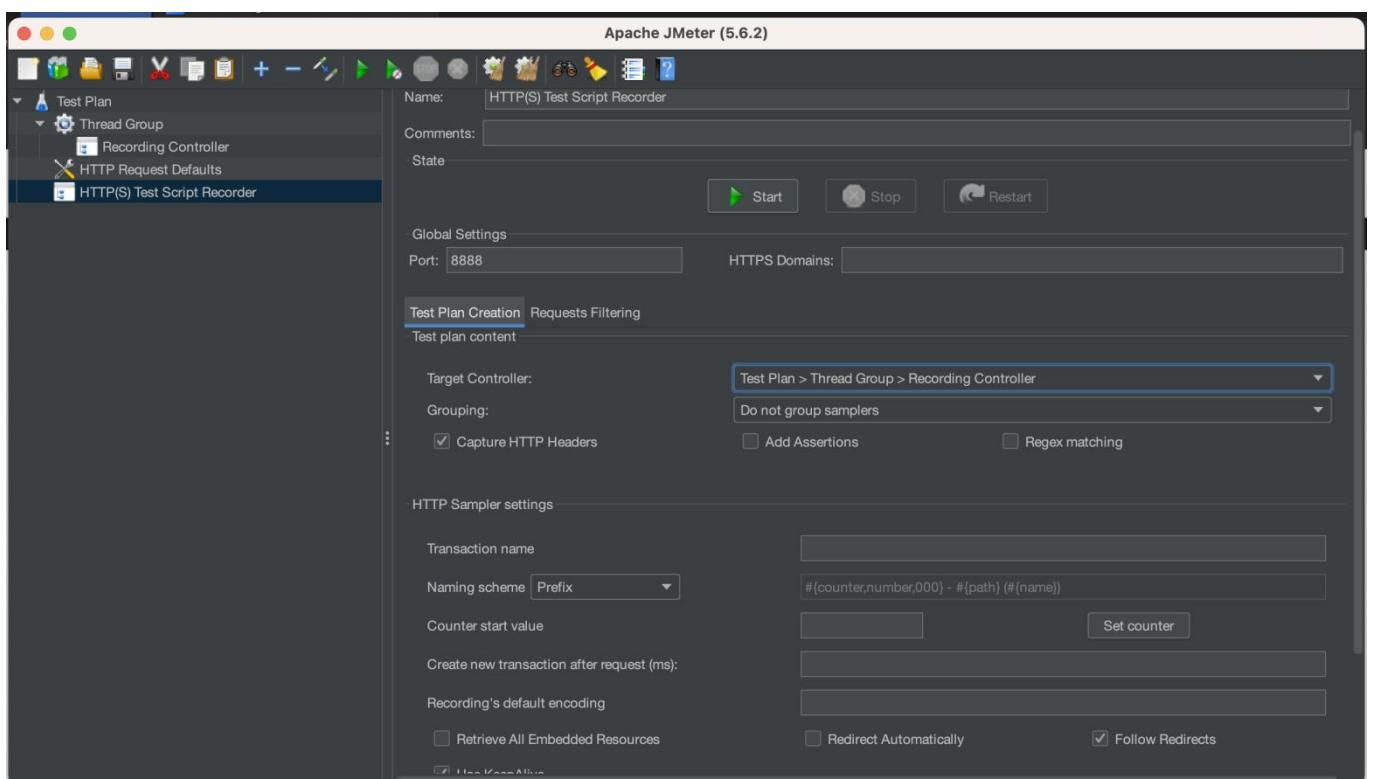
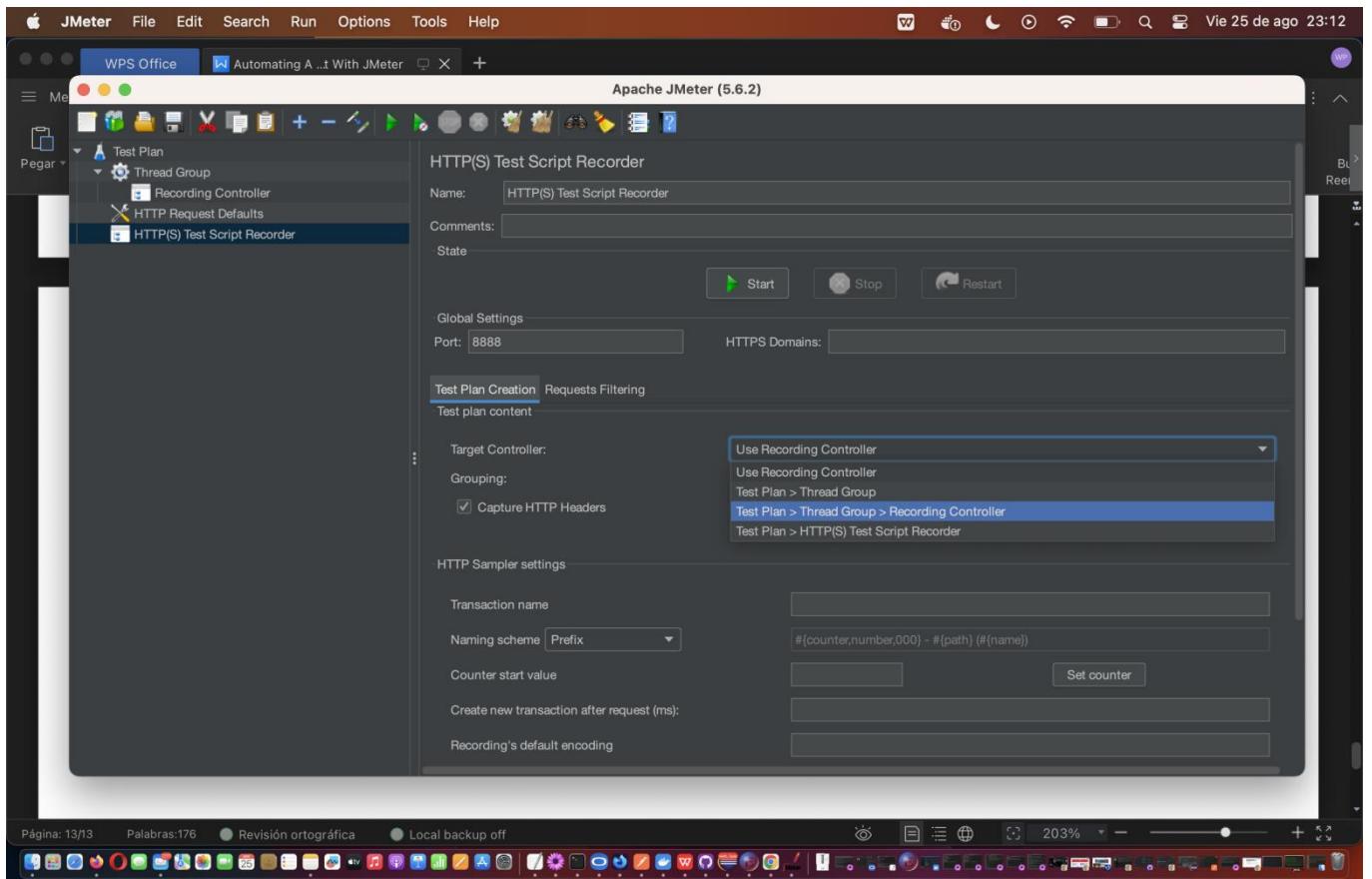


- Add an **HTTP(S) TestScript Recorder**, click right button over Test Plan:
 - Add -> Non-Test Elements -> HTTP(S) TestScript Recorder

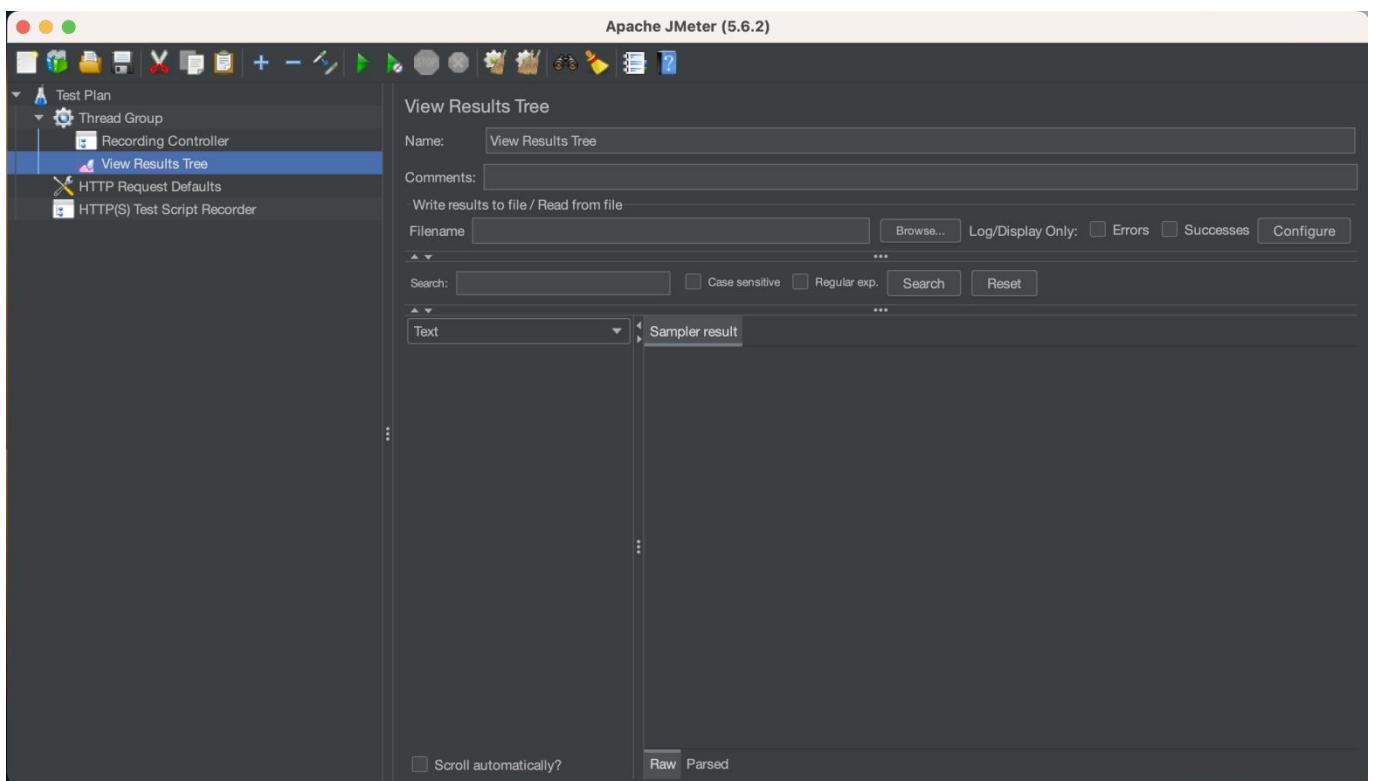
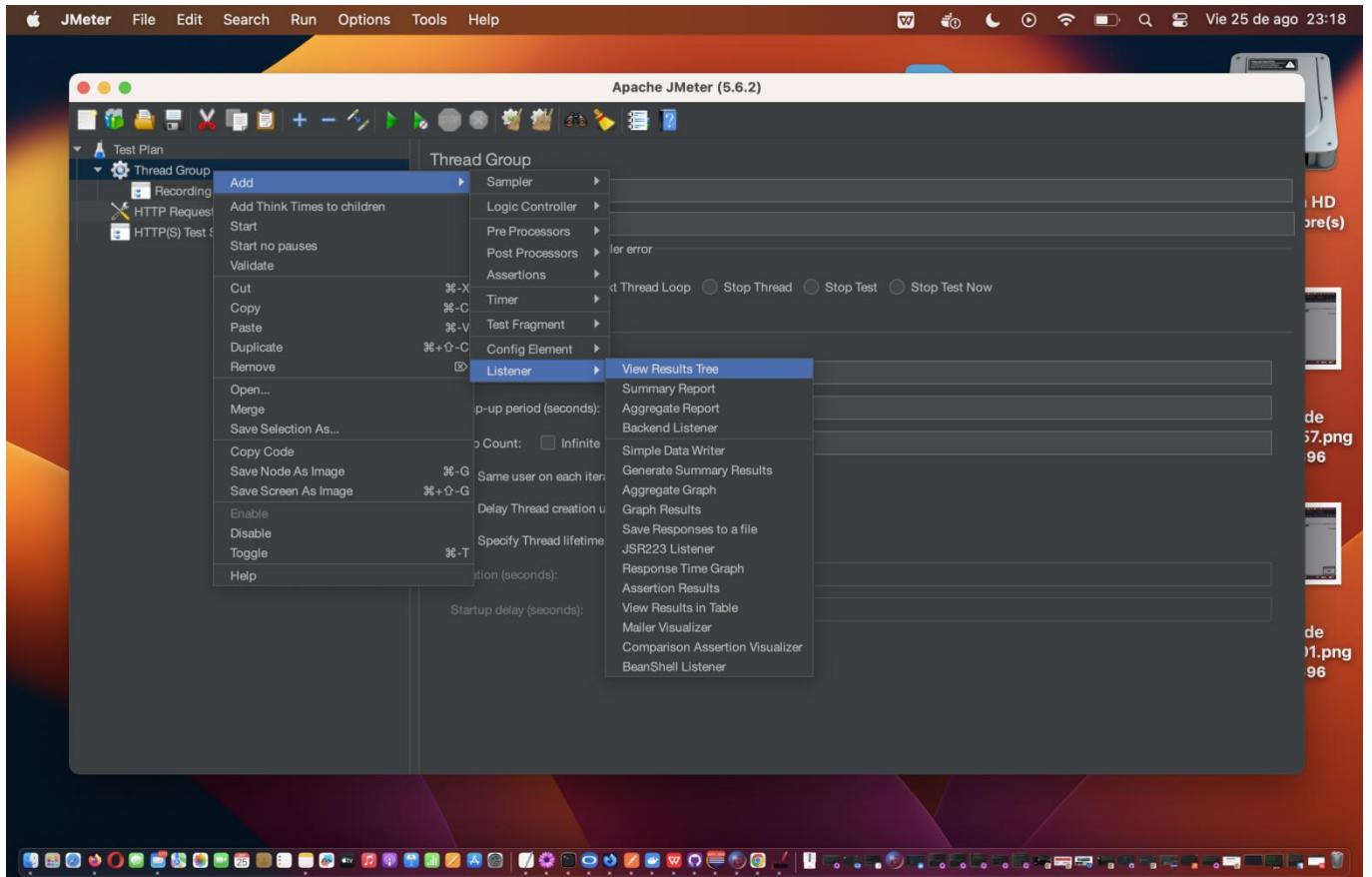


- Port: 8888 -> Like Postman

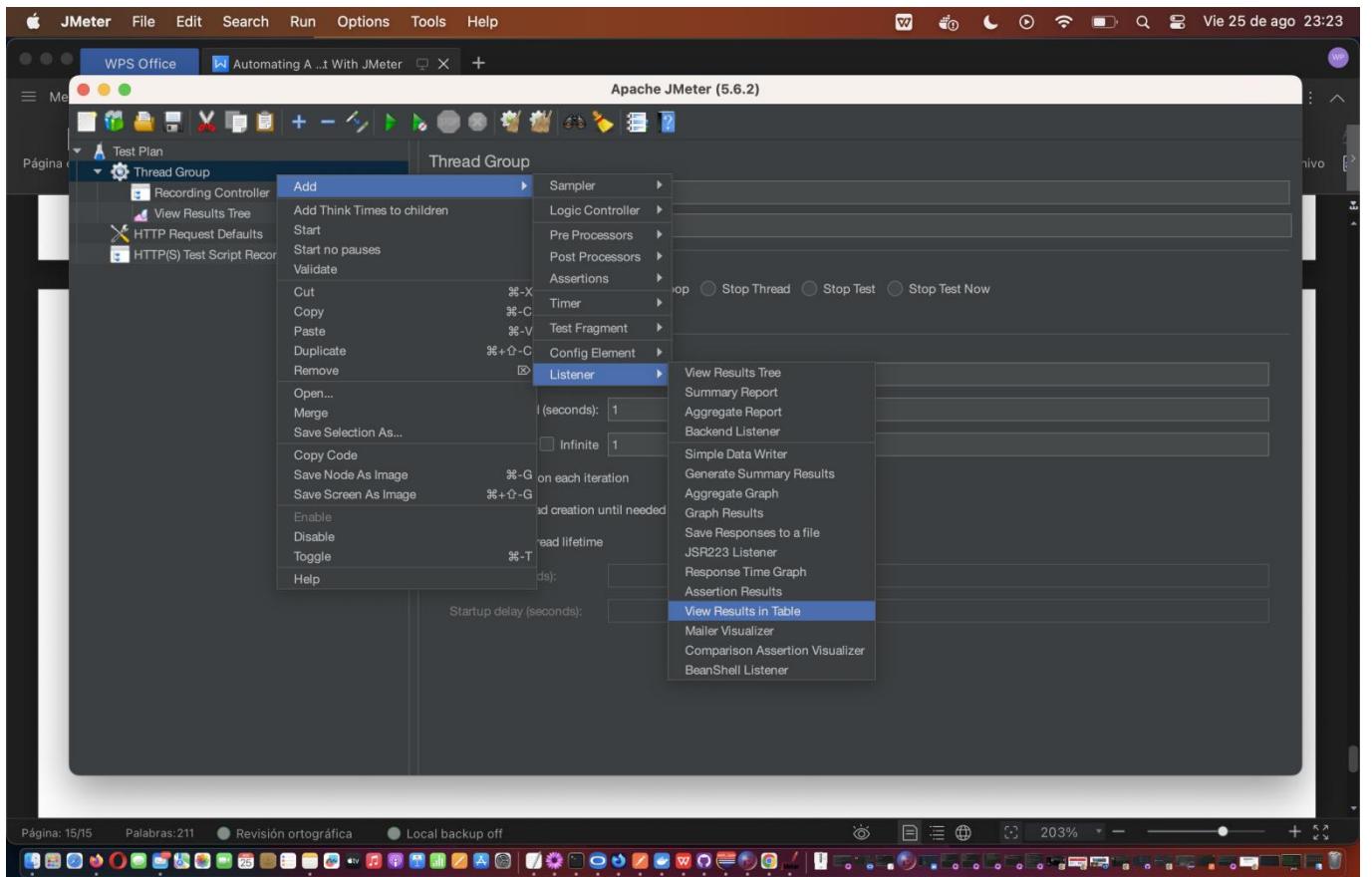
- In Target Controller, choose Test Plan -> Thread Group -> Recording Controller



- To Automate the Saved Tests
- Add a **View Results Tree**, click right button over Thread Group:
 - Add -> Listener -> View Results Tree



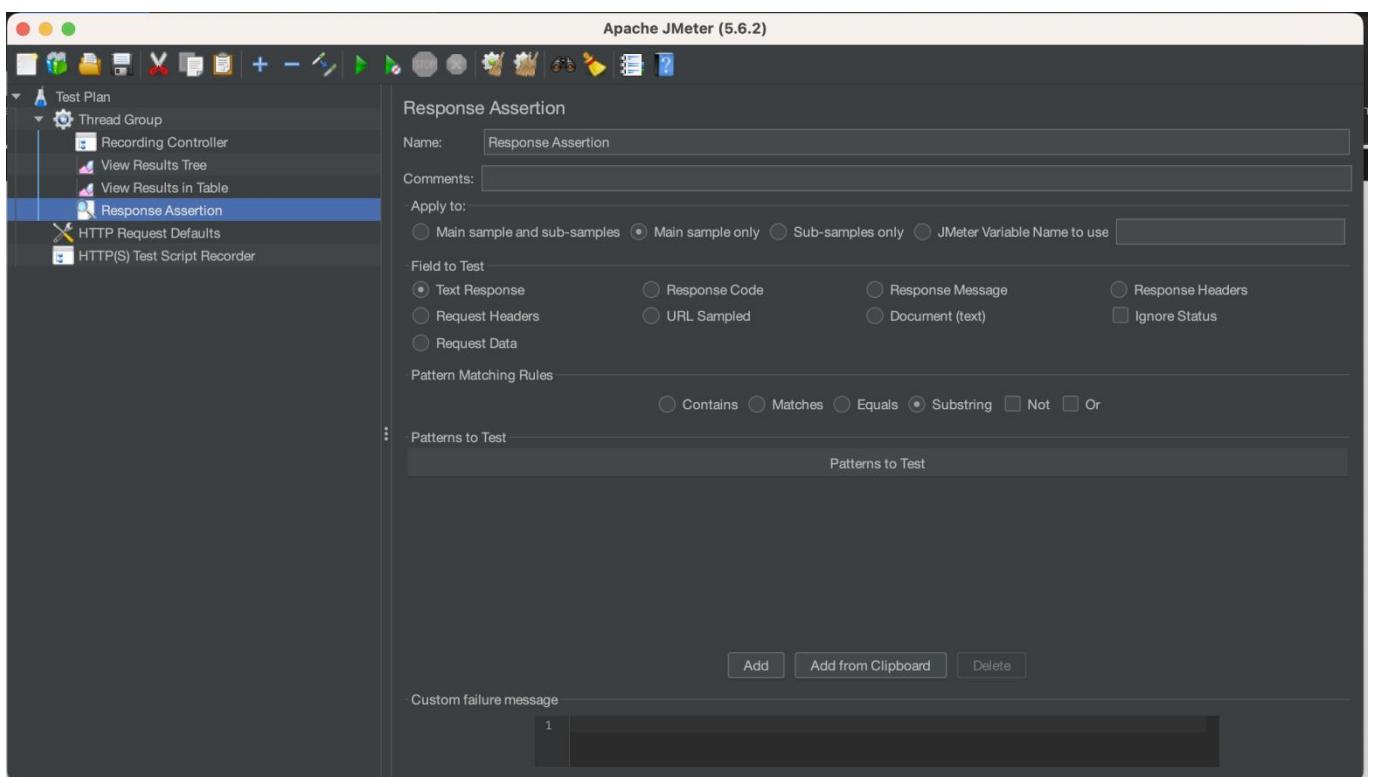
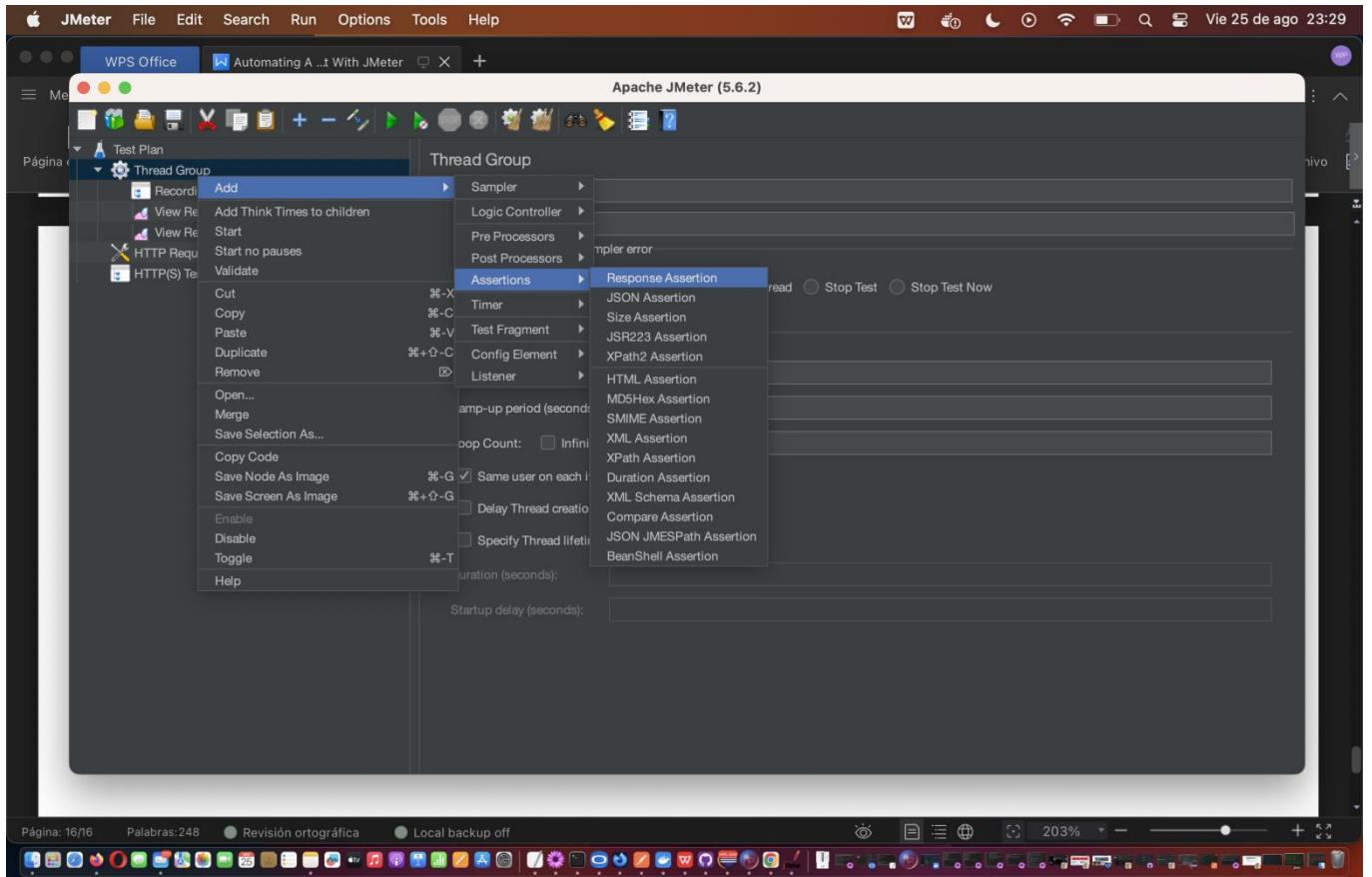
- Add a **View Results Table**, click right button over Thread Group:
- Add -> Listener -> View Results Table



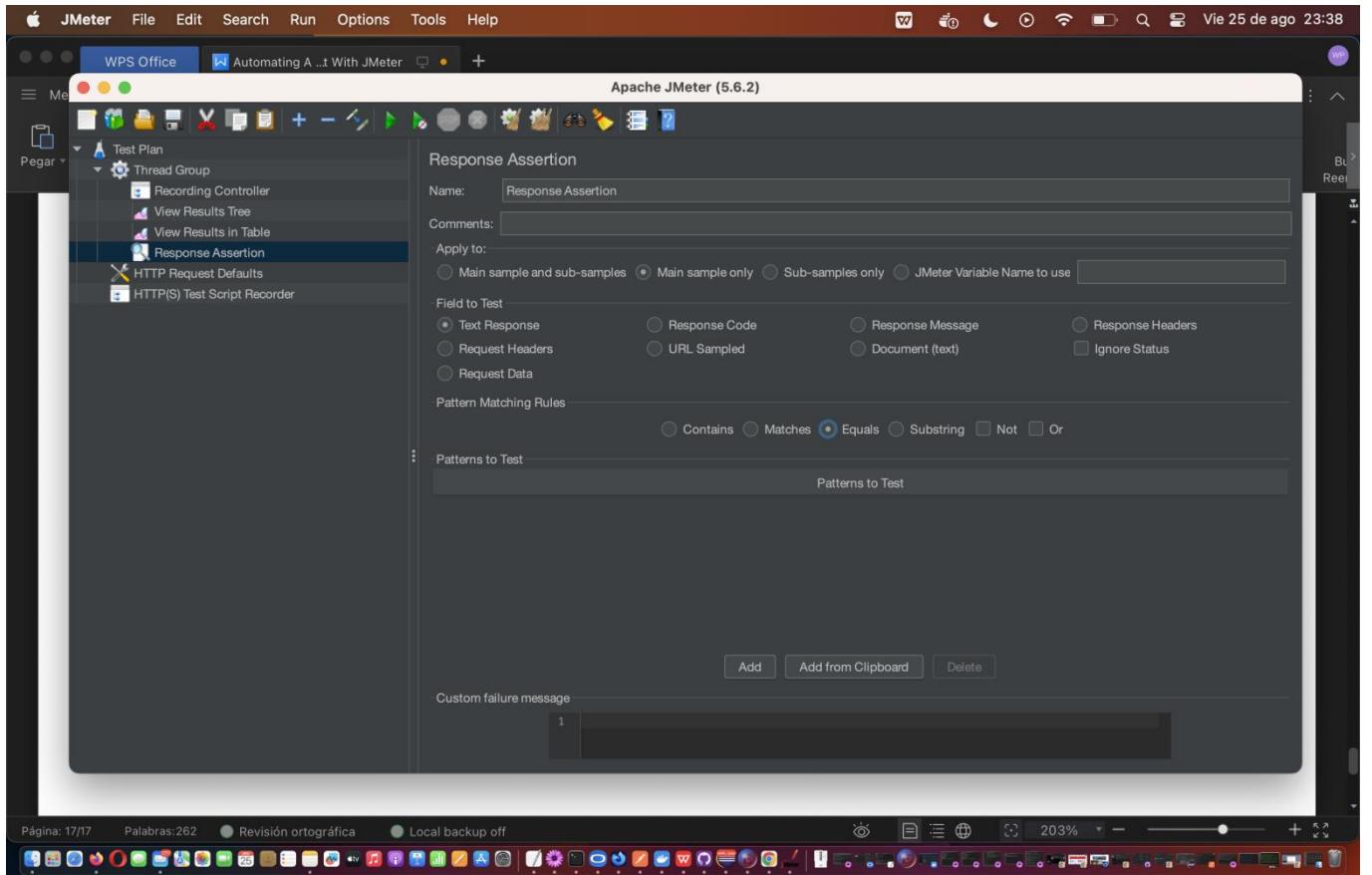
The screenshot shows the Apache JMeter interface with the 'View Results in Table' configuration dialog open. The 'View Results in Table' item is selected in the Test Plan tree. The dialog contains fields for 'Name' (set to 'View Results in Table'), 'Comments', 'Write results to file / Read from file', 'Filename' (empty), and options for 'Log/Display Only', 'Errors', 'Successes', and 'Configure'. A preview table header is shown at the bottom.

Sample #	Start Time	Thread Name	Label	Sample Tim...	Status	Bytes	Sent Bytes	Latency	Connect Ti...
----------	------------	-------------	-------	---------------	--------	-------	------------	---------	---------------

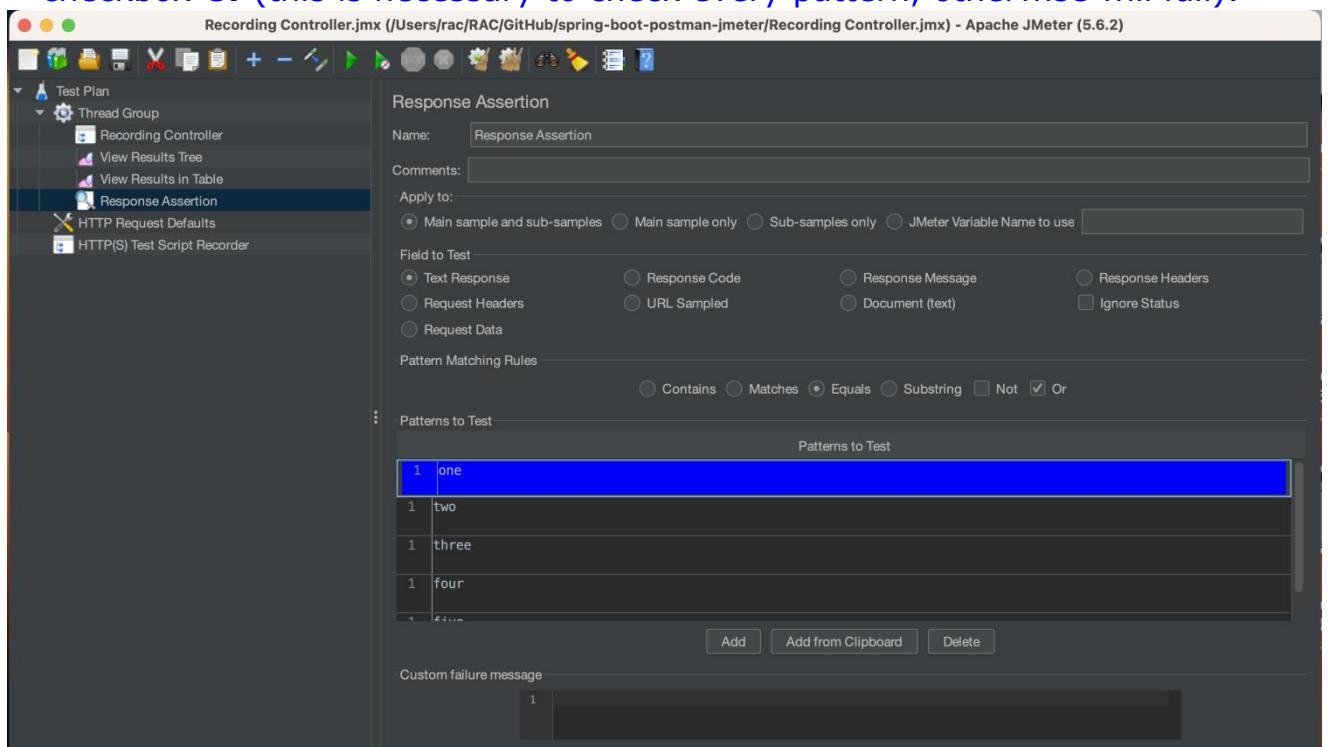
- To Check the Results
- Add a **Response Assertions**, click right button over Thread Group:
 - Add -> Assertions-> Response Assertions



- Add the responses you're waiting for in Postman.
 - Select equals in **Pattern Matching Rules**

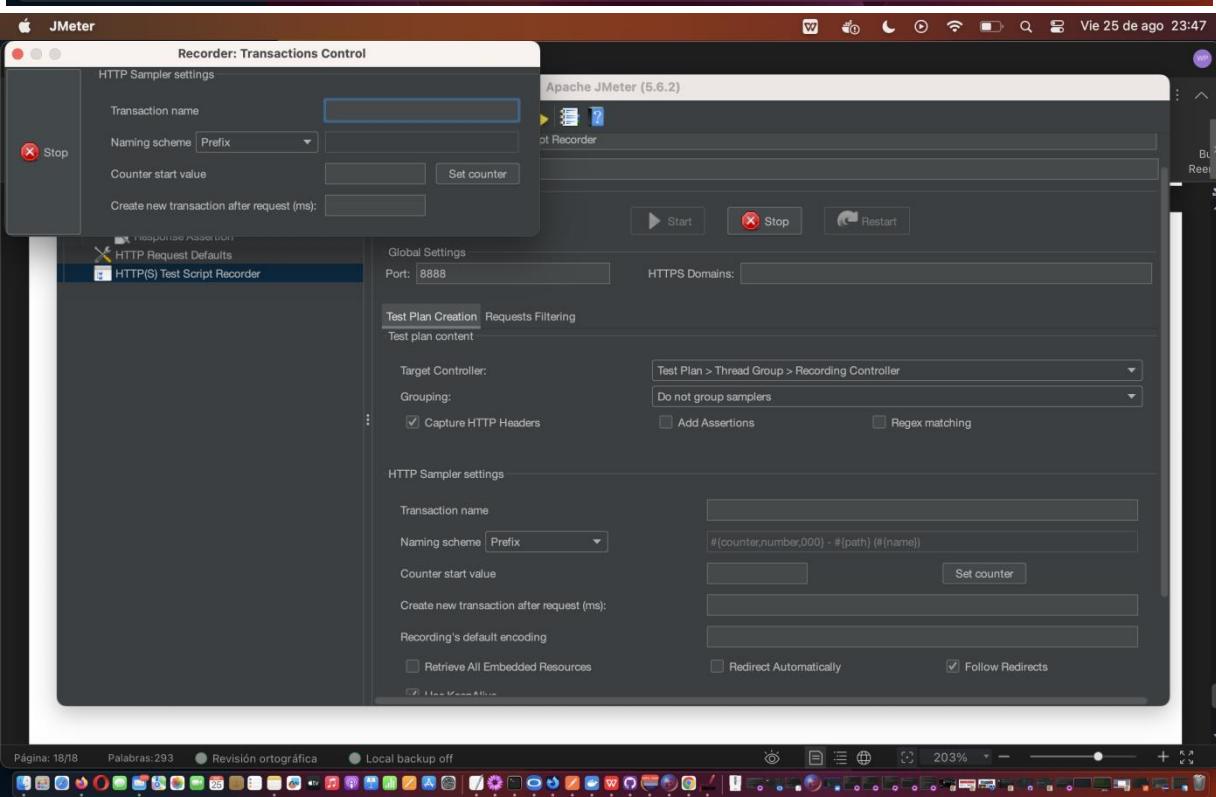
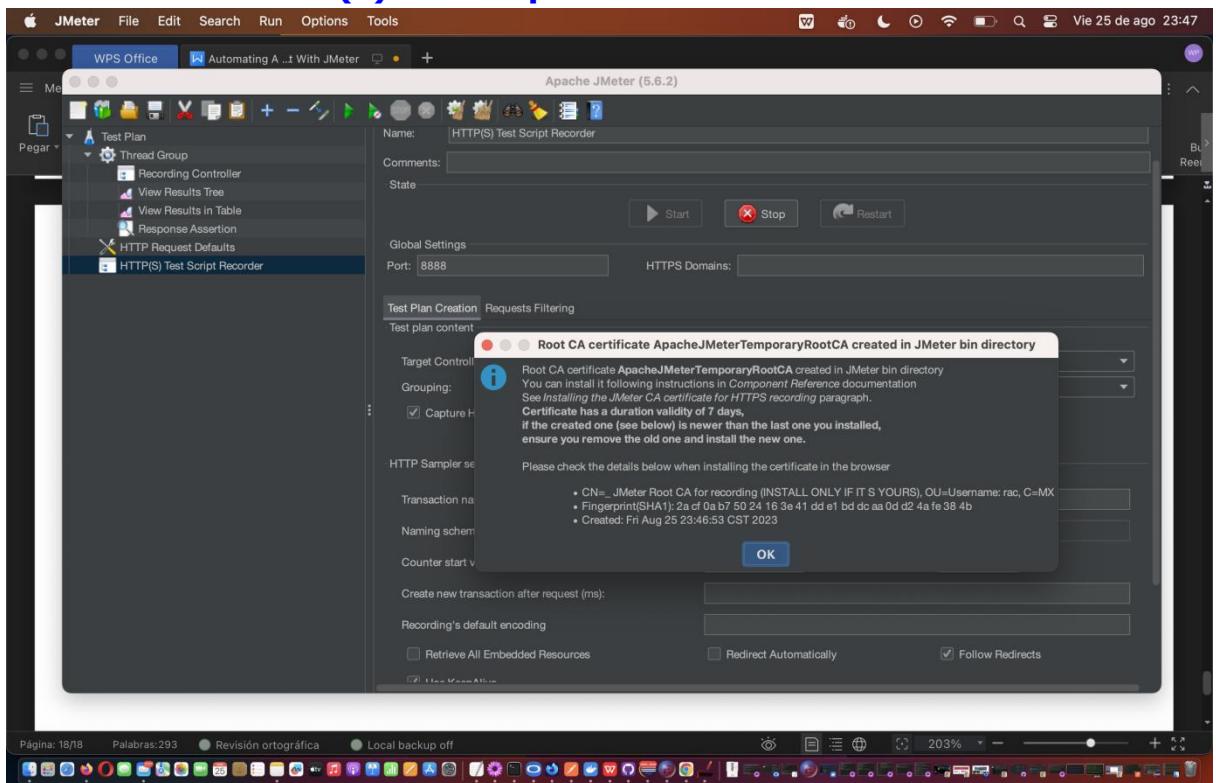


- Add results you get from Postman in Patterns to Test clicking Add, also select the checkbox **Or** (this is necessary to check every pattern, otherwise will fail):

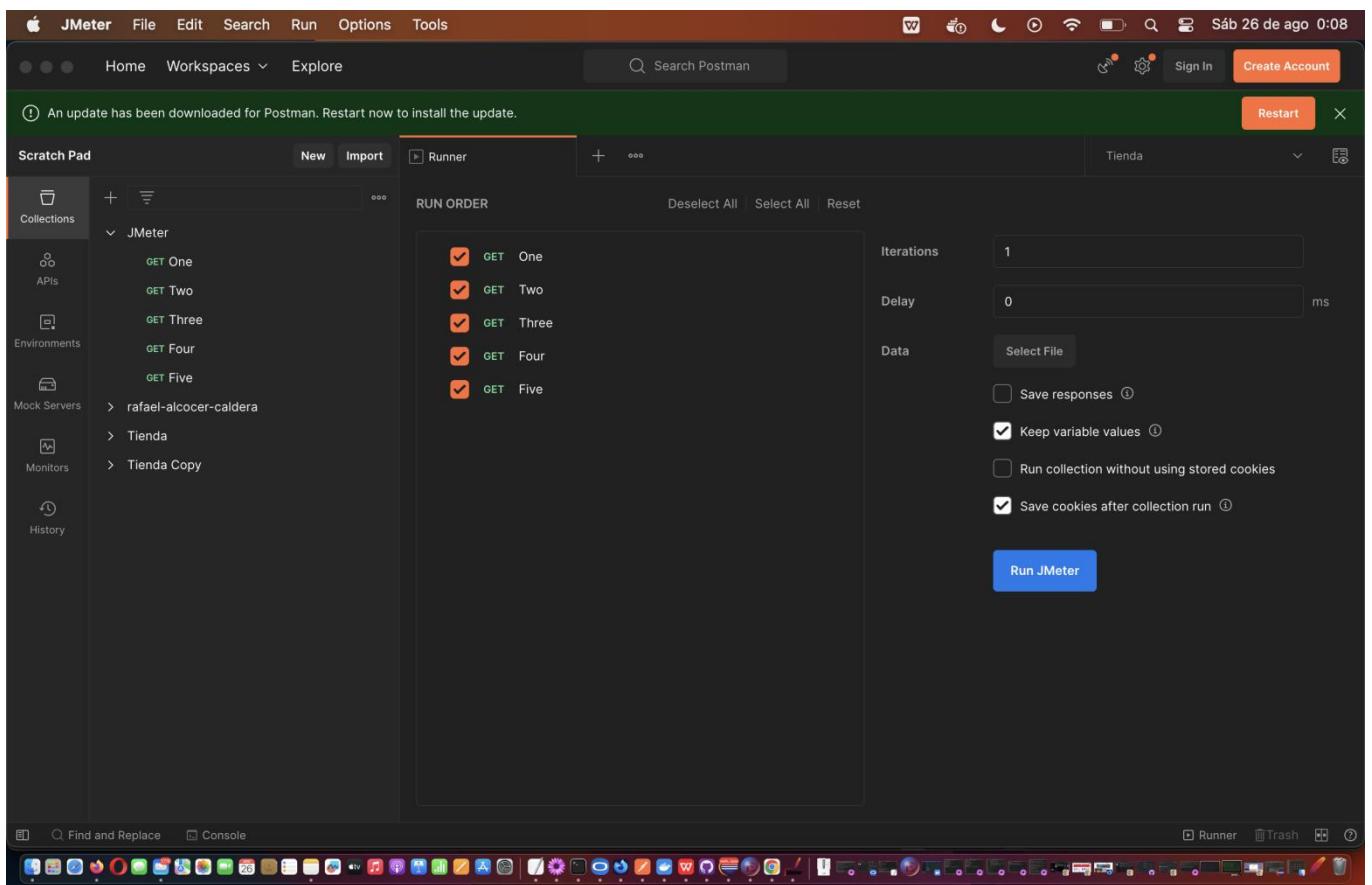
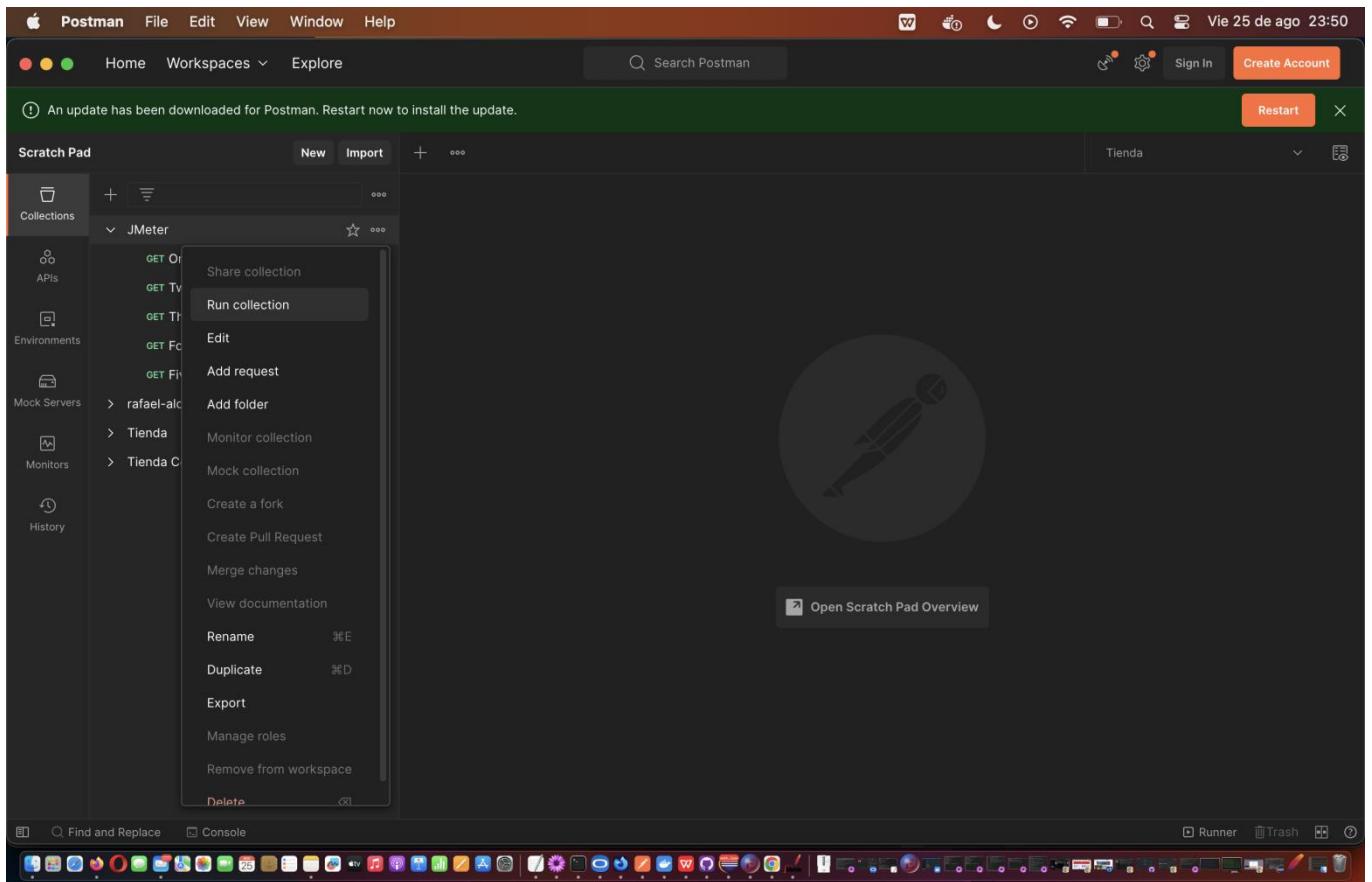


■ Running the Tests

- In JMeter select **HTTP(S) TestScript Recorder** and click on Start:

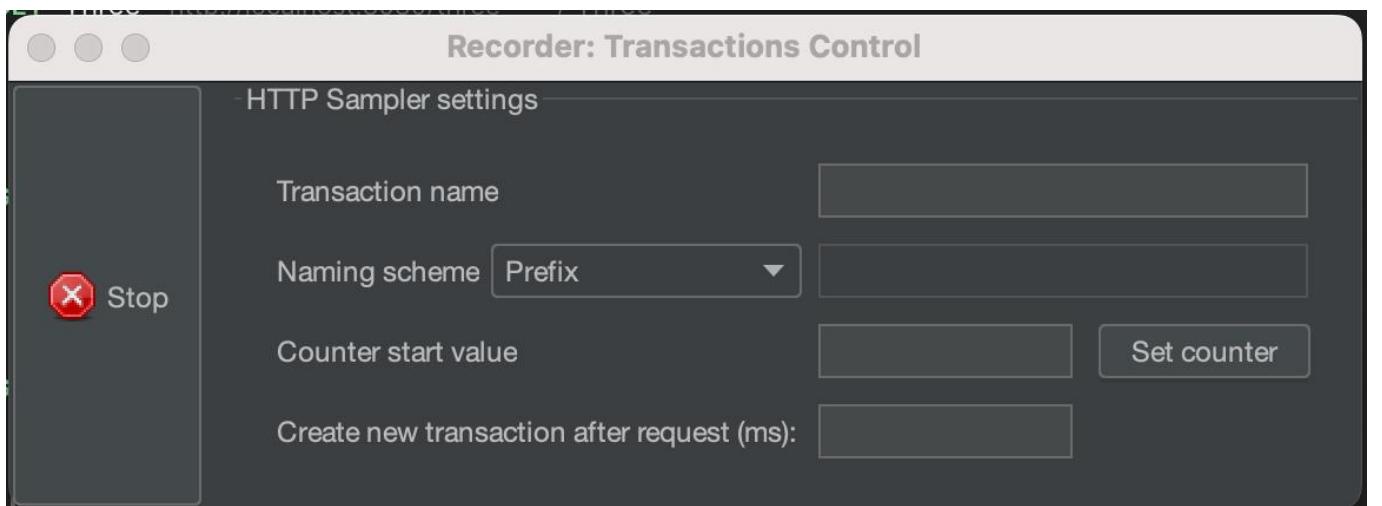


➤ In Postman run the collection:

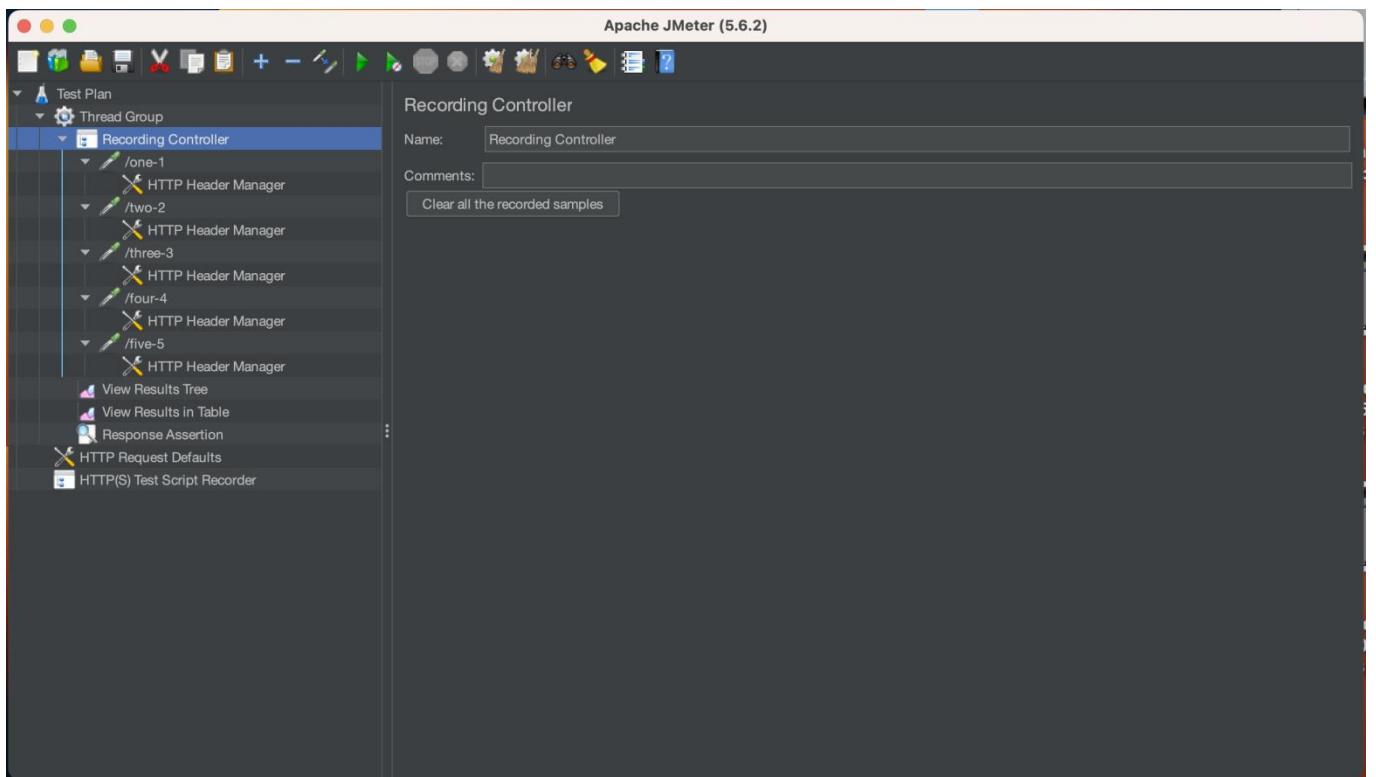


The screenshot shows the Postman application running on a Mac OS X desktop. The main window title is "JMeter". The top menu bar includes "File", "Edit", "Search", "Run", "Options", and "Tools". A system status bar at the top right shows the date and time as "Sáb 26 de ago 0:12". The left sidebar contains sections for "Scratch Pad", "Collections" (with one item), "APIs", "Environments", "Mock Servers" (with one item), "Monitors", and "History". The main workspace displays a collection named "JMeter" under the "Tienda" environment. The collection contains five test cases: "GET One", "GET Two", "GET Three", "GET Four", and "GET Five". Each test case is a row in a table with columns for "Method", "URL", "Label", "Status", "Time", and "Size". All requests are successful (200 OK) with very low response times (e.g., 8 ms) and sizes (e.g., 138 B). Below the table, a note says "This request does not have any tests." for each row. At the bottom of the workspace, there are buttons for "View Summary", "Run Again", "New", and "Export Results". The bottom of the screen shows the Mac OS X dock with various application icons.

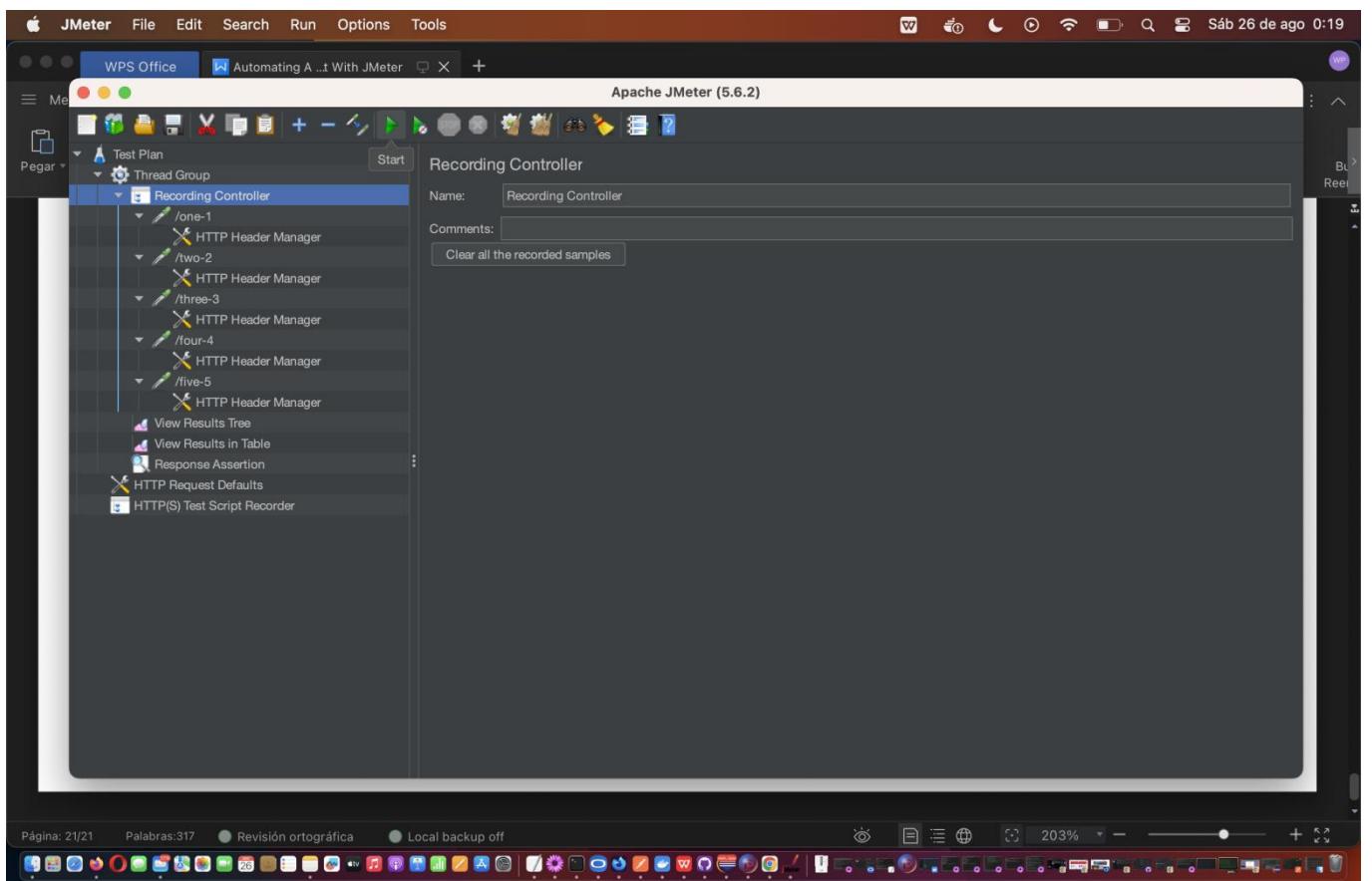
➤ In JMeter Stop the Recorder:



- In JMeter you can see in the Recording Controller, the Postman requests were recorded:



- Click on Start button (the green one in the upper menu):



➤ View results:

Recording Controller.jmx (/Users/rac/RAC/GitHub/spring-boot-postman-jmeter/Recording Controller.jmx) - Apache JMeter (5.6.2)

View Results Tree

Name: View Results Tree
Comments:
Write results to file / Read from file
Filename Browse... Log/Display Only: Errors Successes Configure
Search: Case sensitive Regular exp.

Text

- ✓ /one-36
- ✓ /two-37
- ✓ /three-38
- ✓ /four-39
- ✓ /five-40

Thread Name: Thread Group 1-1
Sample Start: 2023-08-26 00:42:39 CST
Load time: 35
Connect Time: 32
Latency: 35
Size in bytes: 164
Sent bytes: 204
Headers size in bytes: 161
Body size in bytes: 3
Sample Count: 1
Error Count: 0
Data type ("text"|"bin"|""): text
Response code: 200
Response message:

HTTPSampleResult fields:
Content-Type: text/plain; charset=UTF-8
DataEncoding: UTF-8

Scroll automatically?

Recording Controller.jmx (/Users/rac/RAC/GitHub/spring-boot-postman-jmeter/Recording Controller.jmx) - Apache JMeter (5.6.2)

View Results in Table

Name: View Results in Table
Comments:
Write results to file / Read from file
Filename Browse... Log/Display Only: Errors Successes Configure

Sample #	Start Time	Thread Name	Label	Sample Tim...	Status	Bytes	Sent Bytes	Latency	Connect Ti...
1	00:42:39.227	Thread Gro...	/one-36	35	✓	164	204	35	32
2	00:42:39.262	Thread Gro...	/two-37	1	✓	164	204	1	0
3	00:42:39.263	Thread Gro...	/three-38	1	✓	166	206	1	0
4	00:42:39.264	Thread Gro...	/four-39	1	✓	165	205	1	0
5	00:42:39.265	Thread Gro...	/five-40	1	✓	165	205	1	0