



**POLITÉCNICO COLOMBIANO
JAIME ISAZA CADAVID**

TESTING FILE

Integrantes:
Claudia Gil Sánchez.
Sebastián Zapata Castaño.
Mateo Arcila Toro.

Docente: David Córdoba Pimienta.
Materia: Pruebas y gestión de la configuración.

Politécnico Colombiano Jaime Isaza Cadavid
2022-2

1. OBJETIVO

Identificar con el desarrollo de cada una de las pruebas, las cuales se encuentran orientadas a demostrar el control de calidad que se tiene para cada software, además se busca identificar las funcionalidades existentes que estén siendo impactadas por el desarrollo de alguna forma, considerando todos los componentes afectados en todas las capas de la arquitectura de software.

Para cumplir este objetivo se deben definir los criterios de aceptación o rechazo junto con los requisitos no funcionales, es necesario definir el nivel de tolerancia a fallos de calidad, las condiciones que deben cumplirse para dar inicio o reanudar las pruebas; Las condiciones van a depender de los acuerdos de nivel de servicio (SLAs) internos de la organización y también de los acuerdos establecidos en cada proyecto individual.

2. DESCRIPCIÓN DE LOS SISTEMAS

En esta ocasión se van a probar diferentes programas, la página SPACE & BEYOND, <http://demo.testim.io/>, la cual tiene como objetivo registrar las fechas de un destino, cantidad de pasajeros que van a realizar el viaje, selecciona cualquier destino al que desee ir; Esta página nos guía a un formulario donde se van a llenar los datos y subir el archivo de comprobante y finaliza la compra, el fin de la página web es demostrar un procedimiento de selección de datos y completar formularios y selección de campos e imágenes que van a permitir que el usuario se mantenga activo mientras realiza las búsquedas.

Las pruebas van a estar guiadas Junto con la página Timeanddate API Services, <https://dev.timeanddate.com>, donde se selecciona el microservicio llamado Astronomy API, y esta nos ofrece tres servicios diferentes: Astro Position, Astro Event y Tide Service. Todas las ubicaciones de estos servicios se pueden consultar por ID, coordenadas o códigos de aeropuerto IATA; el microservicio va a permitir comprender al usuario como es el funcionamiento y respuestas de la API de acuerdo a cada solicitud enviada.

3. MODULOS DE LOS SISTEMAS A PROBAR

SPACE & BEYOND: para este software se tienen planteados diferentes módulos que competan el servicio de la página web, como inicio se prueba el módulo de ingreso del usuario registrándose con usuario y contraseña y previo a este el registro de fechas y pasajeros, después de completar los campos nos guía a probar el módulo de seleccionar el destino o buscarlo por nombre o color y por último se debe de probar el módulo del registro de datos del usuario final y finalización de la compra.

ASTRONOMY API: Aun que el microservicio tiene diferentes APIS, solo se probara el API de astronimi api, donde se probaran los módulos de astro Position, de este se suministrara información variada de la posición astronómica de diferentes objetos como la luna y el sol.

4. FORMULARIOS DEL APlicativo A PROBAR

SPACE & BEYOND:

- El registro y login del usuario que va a realizar el proceso de compra.
- Registro de fechas del viaje ida y regreso, cantidad de pasajeros adultos y menores de edad.
- El formulario del destino donde se busca por nombre o Color.
- Esta página cuenta con un formulario donde se van a completar nombres, correo, teléfono y numero de seguridad social, además viene como agregado adjuntar un documento y registrar si cuenta con código de promoción.

ASTRONOMY API:

- Este micro servicio no contaba con formularios para probar.

5. METODOLOGIA PARA LA APlicACIÓN DE LAS PRUEBAS

SPACE & BEYOND: La metodología para realizar las pruebas al servicio web, fue usando pruebas E2E con WebDriverIO y el patrón de diseño POM en las cuales se prueban diferentes escenarios que pasen o fallen de acuerdo a la necesidad de la página. Se busca con esta metodología que compruebe el funcionamiento de toda una aplicación, de principio a fin, por ello, se le conoce como pruebas de extremo a extremo o pruebas end-to-end (E2E).

Con estas pruebas E2E vamos a garantizar que el flujo de la aplicación funcione como se esperaría ante la interacción de un usuario real. El foco de detectar posibles problemas que van a encontrar nuestros usuarios en el flujo general del programa. Una vez completado cada test buscamos mostrar los resultados y reportes de la ejecución con Allure, un Framework que se convierte en una herramienta de informe de prueba flexible y liviana en varios idiomas que no solo muestra una representación muy concisa de lo que se ha probado en un formulario de informe web ordenado.

Para cada escenario de prueba se va a crear clases y sus métodos específicos que describen el funcionamiento de cada escenario, se van a extraer los xpath o localizadores desde el inspeccionador de Google para cada campo que se va a mapear.

Los requisitos funcionales del sistema propuesto son los siguientes

- Visualizar información
- Insertar información
- Modificar información
- Eliminar información

Los requisitos no funcionales para el sistema son los siguientes

- Requisitos de eficiencia
- Requisitos de seguridad lógica
- Requisitos de usabilidad

El error permitido para el paso a producción del software: 10%.

Cantidad de ejecuciones: a criterio del usuario.

ASTRONOMY API: Para realizar este microservicio se realizaron las pruebas de aceptación y de performance, utilizando el framework de Karate y Jmeter.

Como estrategia del desarrollo de las pruebas de aceptación, se coloca una rampa o espacio de las transacciones para las pruebas.

Transacciones máximas van a tener entre 50 y 70 por minuto

Las transacciones promedio van a ser entre 20 y 30 por minuto

Cada ejecución va a tener una duración de un minuto como máximo y duración mínima un segundo.

El porcentaje de error aceptado para las pruebas es del 10%

Con la ejecución se van a hacer pruebas enviando 400.00 usuarios el cual va a tener una operación de las 23 horas del día, durante 30 días al mes.

En estas pruebas se tiene un escenario de carga de la cual, se envían 50 transacciones cada mil segundos y estas al final envían 1000.

Tiene como línea base 60 seg pero la ejecución de la carga duro 3min con 42 seg

El resultado de errores se tuvo con un error del 0,7.

Como punto inicial las transacciones máximas fueron de 64000 y el promedio de transacciones fue de 1155.

Para la prueba con el escenario de cargas de estrés se ejecutó al 125% se envían 70 transacciones cada mil segundos y estas al final envían 600.

Tiene como línea base 5 seg pero la ejecución de la carga duro 5 min con 15 seg

El resultado de errores se tuvo con un error del 0,67%

Como punto inicial las transacciones máximas fueron de 292965 y el promedio de transacciones fue de 1167.

6. RESULTADOS POSIBLES DE LAS PRUEBAS

Nombre del resultado de la prueba	Descripción	Causante de error
Pruebas de performance	<ul style="list-style-type: none"> • El sistema funciona de acuerdo a lo solicitado en los requerimientos iniciales. • Permite suministrar diferente información de prueba y responde con exactitud. • Cada prueba responde de acuerdo a la solicitud enviada desde la consola 	Bajo
Pruebas de aceptación	<ul style="list-style-type: none"> • Requerimientos nunca solicitados al proveedor, que afectan el funcionamiento del sistema. • No permitir la creación de más datos a los que se encuentran registrados en el software. • Debe de estar previamente autenticado por una clave generada. • El software diferencia de versiones por lo que debe de estar actualizando para cada prueba. 	Bajo
Pruebas E2E	<ul style="list-style-type: none"> • El sistema presenta inconsistencias en el diseño del formulario y/o informes. • El sistema no funciona como se especificó en los requerimientos iniciales del sistema. • El software no tiene completa cada solicitud realizada. • Falla de acuerdo a cada escenario ya que es un aplicativo que presenta bugs muy seguidos. 	Alto
Prueba de Smoke test	<ul style="list-style-type: none"> • El sistema funciona de acuerdo a lo solicitado en los requerimientos iniciales. • Se ejecuta un escenario desde la página https://loadfocus.com/, mostrando un resultado del 100%. • No presenta errores superficiales de diseño la web probada. 	Bajo

7. RESPONSABLES DE LAS PRUEBAS

Los responsables de las diferentes pruebas son las siguientes personas:

Ing. Mateo Arcila Toro.

Ing. Sebastián Zapata Castaño.

Ing. Claudia Gil Sánchez.

Las cuales en conjunto realizaron y supervisaron el procedimiento de desarrollo y los resultados de cada una de las pruebas implementadas.

Estuvieron al igual a cargo de la documentación de las pruebas anexas.

8. RIESGOS

Ver anexo Matriz de riesgos generada, en la cual se evidencia los posibles riesgos y el porcentaje de impacto que tendrían.

9. OPORTUNIDADES DE MEJORAS

SPACE & BEYOND:

- La página tiene diseñado un login o registro de usuario pero siempre que se registra el usuario, independientemente del nombre siempre queda en la página el nombre JHON.
- Tiene el formulario para realizar el pago, una vez realizado el registro de los datos y haber completado todos los campos no se habilita el botón de pagar o completar pago.

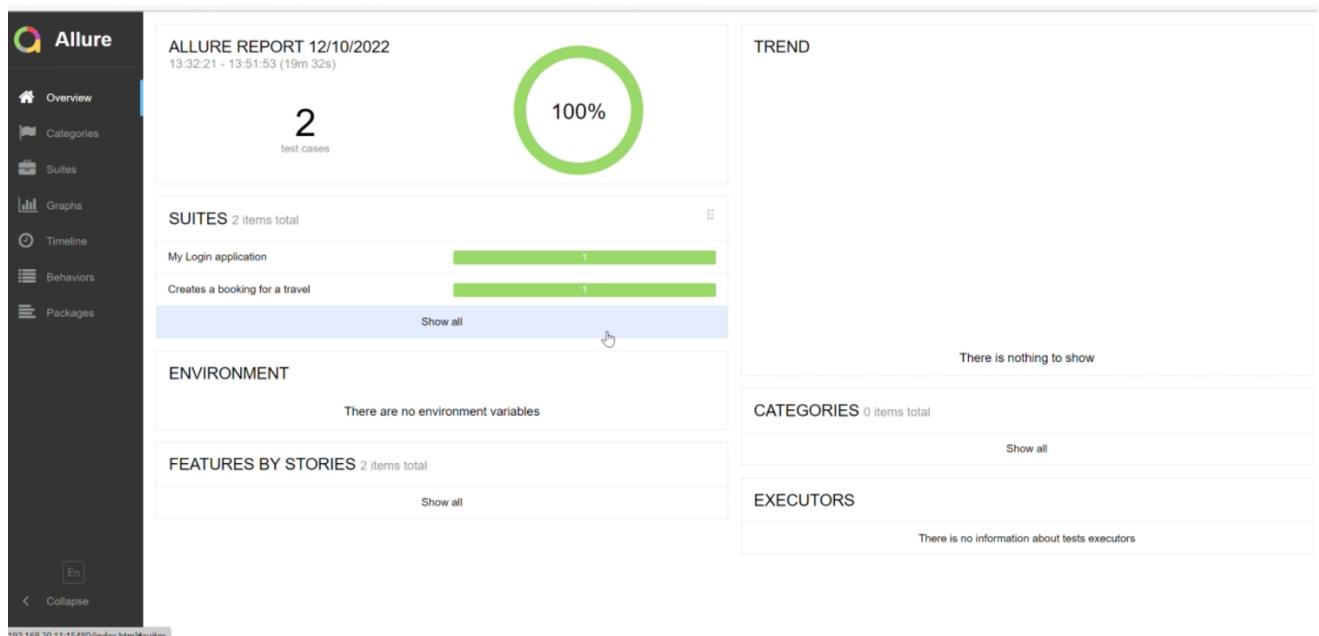
ASTRONOMY API:

- Mejorar el tiempo de expiración de la fecha de consulta del microservicio.
- Implementar más objetos astronómicos para realizar las consultas.
- El porcentaje de error esta alto con respecto a lo esperado, por lo que se recomienda usar la configuración transaccional automática con pods.

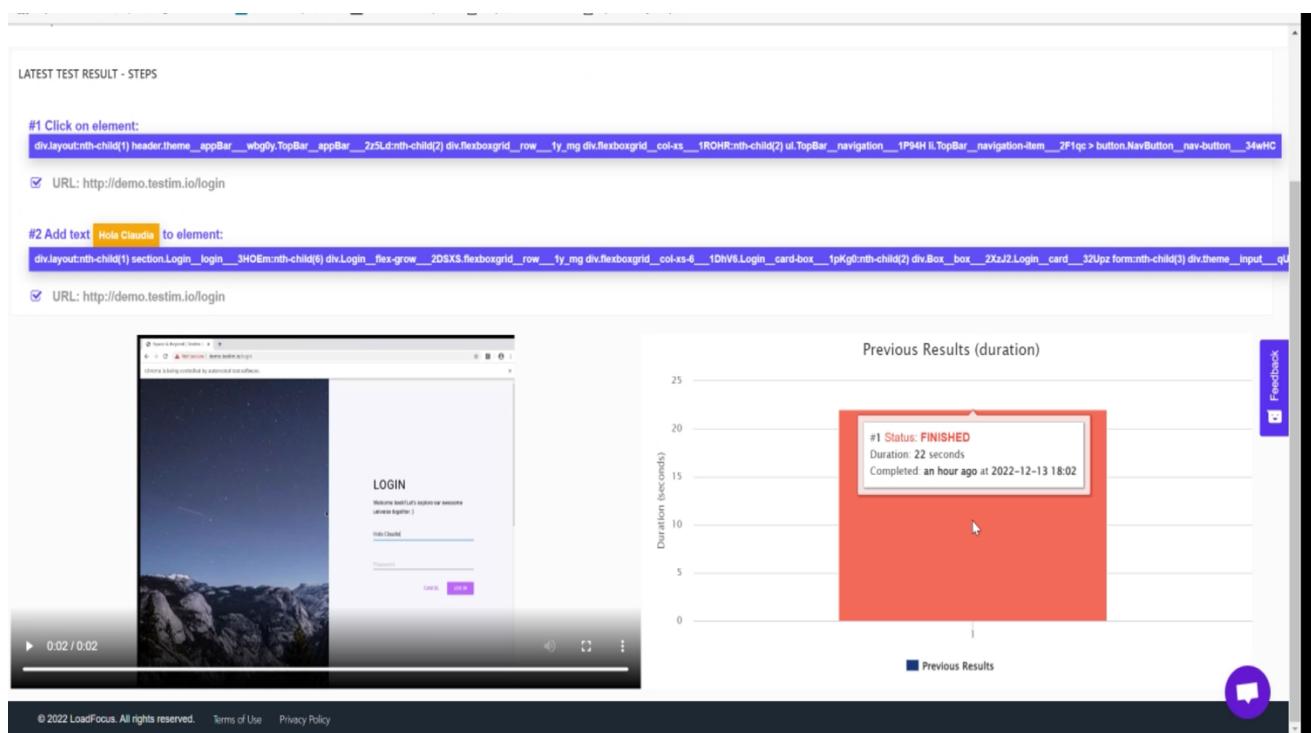
10. EVIDENCIAS OBTENIDAS

SPACE & BEYOND:

- Evidencias E2E



- Evidencias generadas con la página <https://loadfocus.com/>.



ASTRONOMY API:

- Evidencias pruebas de Performance

The screenshot displays two windows of the Apache JMeter application, version 5.5, used for performance testing.

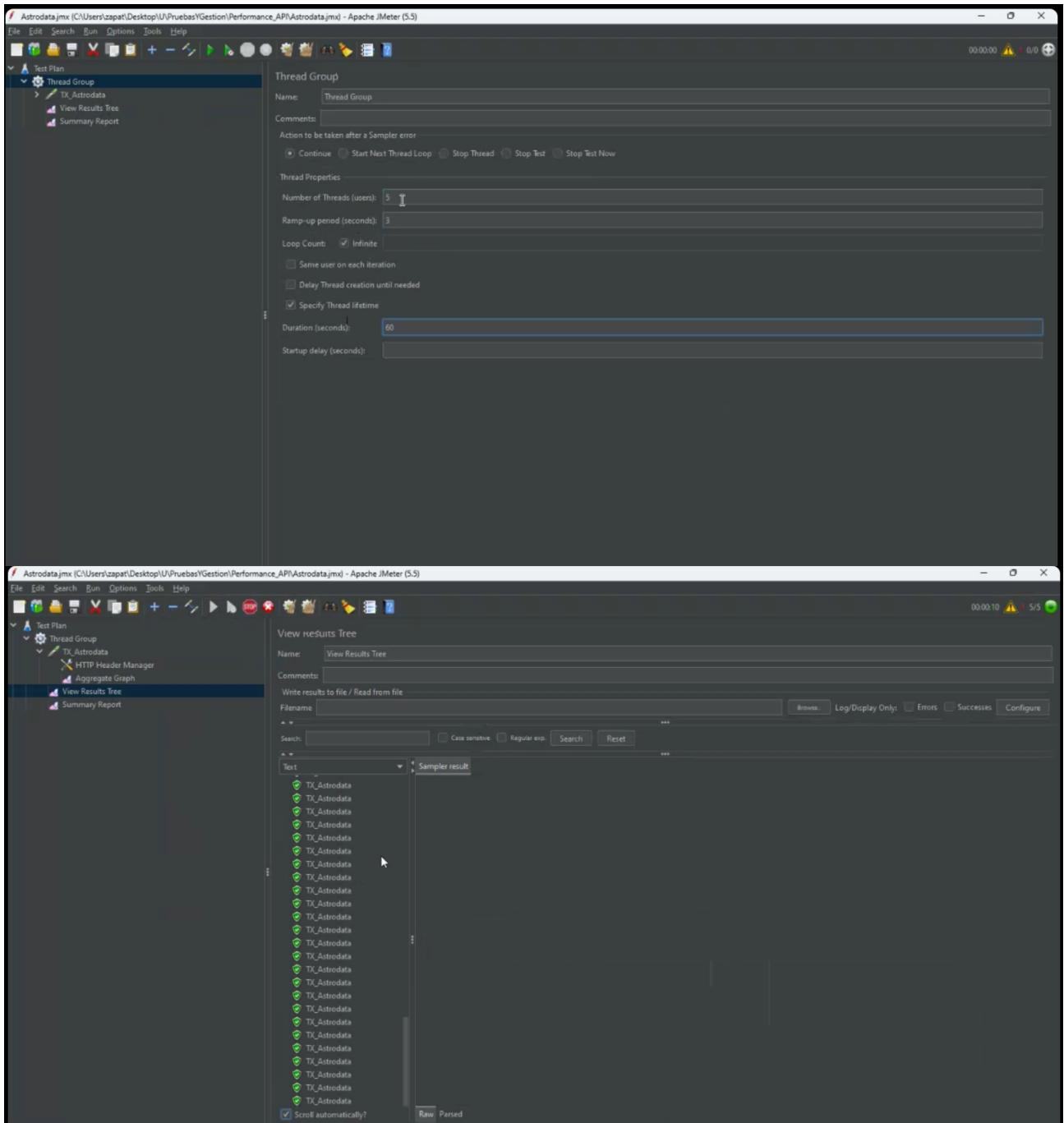
Top Window (HTTP Request Configuration):

- Test Plan:** Contains a **Thread Group** named **TX_Astrodata**.
- HTTP Request:**
 - Name:** TX_Astrodata
 - Web Server:** Protocol (Http), https
 - Server Name or IP:** apixatime.com
 - Port Number:** (empty)
 - HTTP Request Method:** GET
 - Path:** /astrodte
 - Advanced Options:** Redirect Automatically, Follow Redirects, Use KeepAlive, Use multipart/form-data, Browser-compatible headers.
 - Parameters:** A table showing parameters sent with the request:

Name	Value	URL Encode?	Content-Type	Include Equals?
accesskey	dGR65GR5fr	✓		✓
expires	2022-12-10T22%3A44%3A48%2B00%3A00	✓		✓
signature	Fyy4ly%2BhFGvatfnODCbtsGrFxE%3D	✓		✓
version	3	✓		✓
prettyprint	1	✓		✓
object	sun	✓		✓
placed	noway%2Feslo	✓		✓
interval	2022-12-11T01%3A44%3A48	✓		✓

Bottom Window (Thread Group Settings):

- Test Plan:** Contains a **Thread Group** named **Thread Group**.
- Thread Group Properties:**
 - Name:** Thread Group
 - Comments:** Action to be taken after a Sampler error: Continue (radio button selected).
 - Action:** Continue, Start Next Thread Loop, Stop Thread, Stop Test, Stop Test Now.
 - Thread Properties:**
 - Number of Threads (users):** 10
 - Ramp-up period (seconds):** 1
 - Loop Count:** Infinite
 - Same user on each iteration
 - Delay Thread creation until needed
 - Specify Thread lifetime
 - Duration (seconds):** 10
 - Startup delay (seconds):** (empty)



Astrodata.jmx (C:\Users\zapat\Desktop\U\Pruebas\Gestion\Performance_API\Astrodata.jmx) - Apache JMeter (5.5)

File Edit Search Run Options Tools Help

Test Plan

- Thread Group
 - TX_Astrodata
 - HTTP Header Manager
 - Aggregate Graph
 - View Results Tree
 - Summary Report

Summary Report

Name: Summary Report

Comments:

Write results to file / Read from file

Filename:

Browse... Log/Display Only: Errors Successes Configure

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
TX_Astrodata	997	287	232	3488	200.14	0.00%	16.7/sec	4.13	5.58	253.7
TOTAL	997	287	232	3488	200.14	0.00%	16.7/sec	4.13	5.58	253.7

Include group name in label? Save Table Data Save Table Header

Astrodata.jmx (C:\Users\zapat\Desktop\U\Pruebas\Gestion\Performance_API\Astrodata.jmx) - Apache JMeter (5.5)

File Edit Search Run Options Tools Help

Test Plan

- Thread Group
 - TX_Astrodata
 - HTTP Header Manager
 - Aggregate Graph
 - View Results Tree
 - Summary Report

Aggregate Graph

Name: Aggregate Graph

Comments:

Write results to file / Read from file

Filename:

Browse... Log/Display Only: Errors Successes Configure

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %	Throughput	Received KB/sec	Sent KB/sec
TX_Astrodata	997	287	286	289	300	1283	232	3488	0.00%	16.7/sec	4.13	5.58
TOTAL	997	287	286	289	300	1283	232	3488	0.00%	16.7/sec	4.13	5.58

Settings: Graph

Display Graph

Save Graph Save Table Data Save Table Header

Column settings

Columns to display: Average Median 90% Line 95% Line 99% Line Min Max Foreground color

Value font: Sans Serif Size: 10 Style: Normal Draw outlines bar? Show number grouping? Value labels vertical?

Column label selection:

Title: Synchronize with name

Font: Sans Serif Size: 16 Style: Bold

Graph size

Dynamic graph size Width: Height:

X Axis Y Axis (milli-seconds)

Max length of x-axis label: Scale maximum value:

Legend

Placement: Bottom Font: Sans Serif Size: 10 Style: Normal

Astrodata.jmx (C:\Users\zapat\Desktop\U\Pruebas\Gestion\Performance_API\Astrodata.jmx) - Apache JMeter (5.5)

File Edit Search Run Options Tools Help

Test Plan

- Thread Group
 - TX_Astrodata
 - HTTP Header Manager
 - Aggregate Graph
 - View Results Tree
 - Summary Report

Aggregate Graph

Name: Aggregate Graph

Comments:

Write results to file / Read from file

Filename:

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %	Throughput	Received KB/sec	Sent KB/sec
TX_Astrodata	997	287	266	289	300	1283	232	3488	0.00%	16.7/sec	4.13	5.58
TOTAL	997	287	266	289	300	1283	232	3488	0.00%	16.7/sec	4.13	5.58

Settings Graph

Aggregate Graph

Milliseconds

TX_Astrodata

Average Median Min Max

0 287 266 232 3488

TX_Astrodata.jmx (C:\Users\zapat\Desktop\U\Pruebas\Gestion\Performance_API\Astrodata.jmx) - Apache JMeter (5.5)

File Edit Search Run Options Tools Help

Test Plan

- Thread Group
 - TX_Astrodata
 - HTTP Header Manager
 - Aggregate Graph
 - View Results Tree
 - Summary Report

Thread Group

Name: Thread Group

Comments:

Action to be taken after a Sampler error:

Continue Start Next Thread Loop Stop Thread Stop Test Stop Test Now

Thread Properties

Number of Threads (users): 50

Ramp-up period (seconds): 10

Loop Count: Infinite 1000

Same user on each iteration

Delay Thread creation until needed

Specify Thread lifetime

Duration (seconds):

Startup delay (seconds):

Astrodata.jmx (C:\Users\zapat\Desktop\U\Pruebas\Gestion\Performance_API\Astrodata.jmx) - Apache JMeter (5.5)

File Edit Search Run Options Tools Help

Test Plan

Thread Group

TX_Astrodata_Carga

HTTP Header Manager

Aggregate Graph

View Results Tree

Summary Report

View Results Tree

Name: View Results Tree

Comments:

Write results to file / Read from file

Filename:

Search: Case sensitive Regular expression Search Reset

Text Sampler result Request Response data

Thread Name: Thread Group 1-14
Sample Start: 2022-12-12 09:22:27 COT
Load time: 1208
Connect Time: 1115
Latency: 1206
Size in bytes: 253
Sent bytes: 343
Headers size in bytes: 200
Body size in bytes: 53
Data type: "text/html"; charset=UTF-8
Error Count: 0
Data type ("text/html"); charset=UTF-8
Response code: 200
Response message: ✓

HTTPSampleResult fields:
Content-Type: application/json; charset=UTF-8
DataEncoding: UTF-8

Raw Parsed

Scroll automatically

Astrodata.jmx (C:\Users\zapat\Desktop\U\Pruebas\Gestion\Performance_API\Astrodata.jmx) - Apache JMeter (5.5)

File Edit Search Run Options Tools Help

Test Plan

Thread Group

TX_Astrodata_Carga

HTTP Header Manager

Aggregate Graph

View Results Tree

Summary Report

Summary Report

Name: Summary Report

Comments:

Write results to file / Read from file

Filename:

Browse Log/Display Only Errors Successes Configure

Label	# Samples	Average	Min	Max	Std. Dev	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
TX_Astrodata_Ca...	6467	1653	13	64132	2097.64	0.76%	29.1/sec	7.72	9.68	271.6
TOTAL	6467	1653	13	64132	2097.64	0.76%	29.1/sec	7.72	9.68	271.6

Include group name in label? Save Table Data Save Table Header

Astrodata.jmx (C:\Users\zapat\Desktop\U\Pruebas\Gestion\Performance_API\Astrodata.jmx) - Apache JMeter (5.5)

File Edit Search Run Options Tools Help 00:03:42 0 0/50 +

Test Plan

- Thread Group
 - TX_Astrodata_Carga
 - HTTP Header Manager
 - Aggregate Graph
- View Results Tree
- Summary Report

Aggregate Graph

Name: Aggregate Graph
Comments:
Write results to file / Read from file
Filename:

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %	Throughput	Received KB/s...	Sent KB/sec
TX_Astrodata...	6467	1653	1155	3419	4484	8527	13	64132	0.76%	29.1/sec	7.72	9.68
TOTAL	6467	1653	1155	3419	4484	8527	13	64132	0.76%	29.1/sec	7.72	9.68

Settings Graph

Aggregate Graph

Milliseconds

Average Median Min Max

Astrodata.jmx (C:\Users\zapat\Desktop\U\Pruebas\Gestion\Performance_API\Astrodata.jmx) - Apache JMeter (5.5)

File Edit Search Run Options Tools Help 00:03:42 0 0/50 +

Test Plan

- Thread Group
 - TX_Astrodata_Estres
 - HTTP Header Manager
 - Aggregate Graph
- View Results Tree
- Summary Report

Thread Group

Name: Thread Group
Comments:
Action to be taken after a Sampler error
 Continue
 Start Next Thread Loop
 Stop Thread
 Stop Test
 Stop Test Now

Thread Properties

Number of Threads (users): 63
Ramp-up period (seconds): 20
Loop Count: Infinite 100
 Same user on each iteration
 Delay Thread creation until needed
 Specify Thread lifetime
 Duration (seconds):
 Startup delay (seconds):

Summary Report

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
TX_Astrodata_Estres	9422	2011	100	292965	4759.10	0.67%	30.0/sec	7.93	9.97	270.9
TOTAL	9422	2011	100	292965	4759.10	0.67%	30.0/sec	7.93	9.97	270.9

Aggregate Graph

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %	Throughput	Received KB/sec	Sent KB/sec
TX_Astrodata_Estres	9422	2011	1167	3978	4825	12397	100	292965	0.67%	30.0/sec	7.93	9.97
TOTAL	9422	2011	1167	3978	4825	12397	100	292965	0.67%	30.0/sec	7.93	9.97

- Evidencias pruebas de Aceptación

Summary Tags Feature: src/test/java/Api/tests/users.feature I as a user i need validate astrodta Api service	
Scenarios	
2022-12-12 07:13:01 p. m.	210
[1:3] Get astrology data	1
[2:17] Post new objet	9
[3:24] validate version	1
Scenarios: [1:3] Get astrology data	ms: 2341
4 Given url "https://api.xmltime.com"	1
5 And path "/astrodta"	1
6 And path expires = "2022-12-11T04%3A02%3A19%2800%3A00"	1
7 And param accesskey = "dGR6SGRSfr"	1
8 And param signature = "%4lAhgGMvDu4dad5l@muchN2BaWk%3D"	1
9 And param version = "3"	1
10 And param prettyprint = "1"	1
11 And param object = "sun"	0
12 And param placeid = "norway%2Foslo"	0
13 And param interval = "2022-12-11T02%3A02%3A19"	0
14 When method get	2116
15 Then status 200	0
Scenarios: [2:17] Post new objet	ms: 928
18 Given url "https://api.xmltime.com"	1
19 And path "/astrodta"	0
20 And request {"object": "moon"}	107
21 When method post	820
22 Then status 401	0
Scenarios: [3:24] validate version	ms: 459
25 Given url "https://api.xmltime.com"	0
26 And path "/astrodta?expires=2022-12-	4
27 When method get	322
28 Then status 200	0
29 And match \$version == 4	133
30 And match \$version contains '#notnull'	0