2025

Pruebas de Carga

WICHAT

GRUPO ES 4C

ASW – INGENIERÍA INFÓRMATICA DE SOFTWARE

Contenido

Initial Situation

The application is deployed on an Azure virtual machine with the following specifications:

- Linux as the operating system
- Ubuntu 24.04 LTS image
- 30 GB of storage
- Standard D2s v3 size
- 8 GB of RAM
- 2 vCPUs
- x64 architecture

The version of the deployed application used for testing is as follows.

Used Tool

To carry out the load testing, the free tool Gatling—recommended by the teaching staff—was used. Specifically, version 3.13 of Gatling Bundle was employed. Gatling is a load testing tool based on Scala, designed to simulate a large number of concurrent users on web applications and APIs, with a focus on performance and automation.

This version stands out for its automation and simplicity, allowing the use of both the recorder and testing features from a Maven project, with scripts written in Scala (though Java is also supported). Gatling generates reports in HTML format, which will be presented below.

Tests

Previous Configuration

Before running the tests, the Recorder tool is used to capture the requests we want to test and convert them into code. In our case, we chose to use Java. Each simulated user performs the following actions:

- Access the registration window
- Add a new user
- Log in
- Play the game (answering 10 questions)
- Make 2 queries to the LLM
- View the ranking
- View the user profile
- Modify settings and save
- Automatic requests: load multimedia resources, update statistics, fetch in-game settings, etc.

After generating the Java class containing the code to perform these requests, it is slightly modified to generate random users and simulate simultaneous user insertions. The class used can be accessed through the following link.

Finally, the load tests are executed, varying the number of users:

Test 1: 120 users

First, 120 users are tested (2 users per second for 60 seconds), yielding the following results:

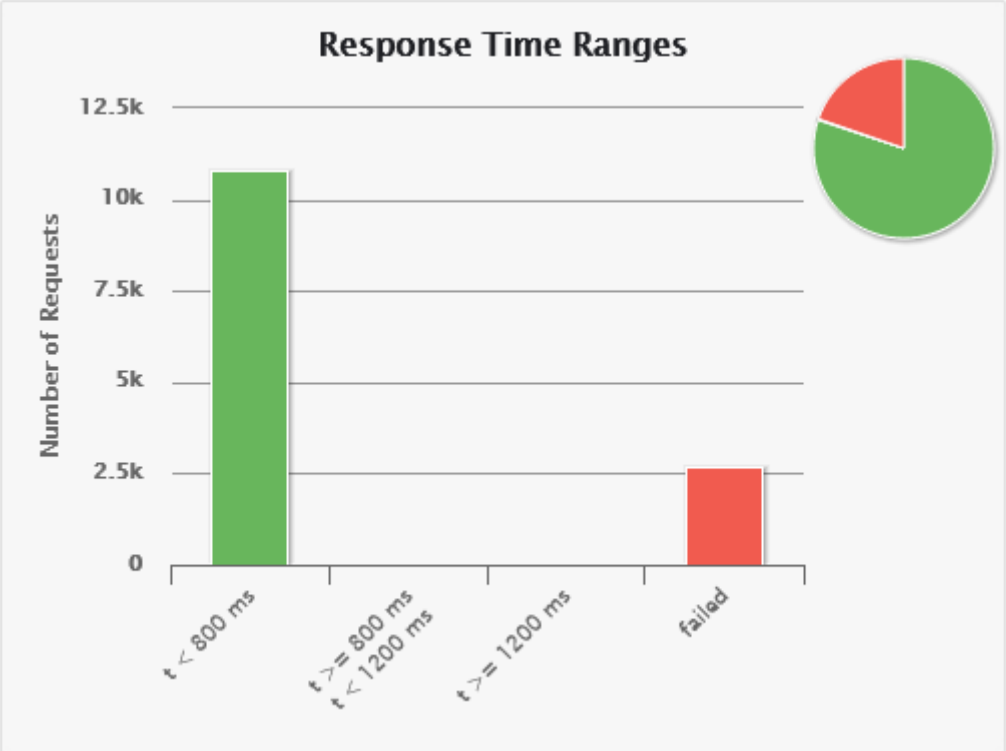
Requests	Executions			
	Total	OK	KO	% KO
All Requests	5,410	4,366	1,044	NaN
Inicio	120	120	0	0.00
Ir A Registro	120	120	0	0.00
Cargar recurso multimedia	1,440	1,440	0	NaN
Cargar recurso multimedia Red...	1,090	255	835	NaN
Añadir Usuario	120	120	0	0.00
Iniciar Sesión	120	120	0	0.00
Obtener Pregunta	1,080	1,080	0	NaN
Obtener Ajustes	240	240	0	0.00
Preguntar al LLM	240	31	209	87.08
Aumentar Partidas Jugadas	120	120	0	0.00
Actualizar Estadísticas	120	120	0	0.00
Consultar Ranking	240	240	0	0.00
Consultar Perfil	120	120	0	0.00
Obtener Amigos	120	120	0	0.00
Guardar Ajustes	120	120	0	0.00

With 120 users, we observe that 80.7% of the requests are successfully completed and on time. However, errors are concentrated solely in the redirections for loading multimedia resources (429) and the queries to the LLM (likely because Gemini has a request limit within a specific time frame, which seems to be 31).

The full report can be accessed through the following link.

Test 2: 300 users

Next, 300 users are tested (5 users per second for 60 seconds), yielding the following results:



Requests	Executions			
	Total	OK	KO	% KO
All Requests	13,510	10,846	2,664	NaN
Inicio	300	300	0	0.00
Ir A Registro	300	300	0	0.00
Cargar recurso multimedia	3,600	3,600	0	NaN
Cargar recurso multimedia Red...	2,710	615	2,095	NaN
Añadir Usuario	300	300	0	0.00
Iniciar Sesión	300	300	0	0.00
Obtener Pregunta	2,700	2,700	0	NaN
Obtener Ajustes	600	600	0	0.00
Preguntar al LLM	600	31	569	94.83
Aumentar Partidas Jugadas	300	300	0	0.00
Actualizar Estadísticas	300	300	0	0.00
Consultar Ranking	600	600	0	0.00
Consultar Perfil	300	300	0	0.00
Obtener Amigos	300	300	0	0.00
Guardar Ajustes	300	300	0	0.00

The results are almost identical, with 80.1% of requests completed on time. As for the errors, once again, they are solely attributed to the redirections for multimedia resource loading and the queries to the LLM.

While the redirections for multimedia resource loading fail at a similar rate, the number of completed LLM requests matches exactly with the previous test (31), which reinforces the idea that the Gemini request limit is the cause, and explains the slight 0.6% difference in completed requests.

The full report can be accessed through the following link.

Test 3: 600 users

In the third iteration, 600 users are tested (10 users per second for 60 seconds), yielding the following results:

Requests	Executions			
	Total	OK	KO	% KO
All Requests	27,010	21,645	5,365	NaN
Inicio	600	600	0	0.00
Ir A Registro	600	600	0	0.00
Cargar recurso multimedia	7,200	7,200	0	NaN
Cargar recurso multimedia Red...	5,410	1,215	4,195	NaN
Añadir Usuario	600	599	1	0.17
Iniciar Sesión	600	600	0	0.00
Obtener Pregunta	5,400	5,400	0	NaN
Obtener Ajustes	1,200	1,200	0	NaN
Preguntar al LLM	1,200	31	1,169	NaN
Aumentar Partidas Jugadas	600	600	0	0.00
Actualizar Estadísticas	600	600	0	0.00
Consultar Ranking	1,200	1,200	0	NaN
Consultar Perfil	600	600	0	0.00
Obtener Amigos	600	600	0	0.00
Guardar Ajustes	600	600	0	0.00

For 600 users, we begin to observe the first issues: although the application continues to respond correctly to 80% of the requests, it is noted that 10.2% had response times exceeding 800 ms.

As for the errors, they are still concentrated solely on the redirections for multimedia content loading and the queries to the LLM (the 31 request limit remains in place). A single error (400) was observed when adding a user, which is likely due to a collision in the random user generation.

The full report can be accessed through the following link.

Test 4: 1200 users

In the final iteration, 1200 users are tested (20 users per second for 60 seconds), yielding the following results:

Requests	Executions			
	Total	OK	KO	% KO
All Requests	52,676	39,459	13,217	NaN
Inicio	1,200	1,200	0	NaN
Ir A Registro	1,200	1,200	0	NaN
Cargar recurso multimedia	13,118	13,118	0	NaN
Cargar recurso multimedia Red...	10,810	2,415	8,395	NaN
Añadir Usuario	1,200	520	680	NaN
Iniciar Sesión	1,200	559	641	NaN
Obtener Pregunta	10,800	10,800	0	NaN
Obtener Ajustes	2,400	1,779	621	NaN
Preguntar al LLM	2,400	61	2,339	NaN
Aumentar Partidas Jugadas	1,200	959	241	NaN
Actualizar Estadísticas	1,200	1,039	161	NaN
Consultar Ranking	2,400	2,341	59	NaN
Consultar Perfil	1,200	1,148	52	NaN
Obtener Amigos	1,148	1,133	15	NaN
Guardar Ajustes	1,200	1,187	13	NaN

For 1200 users, a decrease in completed requests can already be observed, with less than 75% being completed. Additionally, 10% of requests had a response time of over 800 ms.

This time, the errors are more evenly distributed:

- Resource loading redirection (error 429): Again, these errors account for the majority, but they maintain a similar proportion to the previous tests.
- Add user (Request timeout): It can be observed that for just over half of the "add user" requests, the server is unable to keep up.
- Log in (error 401): Authorization error, most likely caused by trying to log in with users that could not be registered due to the previous errors.
- Get settings (mostly error 404, Request timeout): The 404 error is likely due to trying to load settings for a user who isn’t registered, due to the previous "add user" errors.

The rest of the requests behave similarly, with errors being split between Request timeout and error 404 (likely due to the failures in adding users).

An interesting point is that, this time, the questions to the LLM resulted in 61 completed answers (almost double the 31 from the previous tests), which may suggest that the request limit to Gemini was refreshed, allowing for 31 more (30 in this case) requests.

The full report can be accessed through the following link.

Conclusions

We can conclude that the application generally performs well, even under high user demand, maintaining completed response rates above 75%.

It is observed that with 600 users, the application begins to experience issues with response times, and with 1200 users, some requests fail to complete (although they are few). It is also noted that errors in the "add user" requests affected some subsequent requests, as they tried to load resources for non-existent users.

Finally, after reviewing the data and conducting a brief investigation, it seems that the Gemini API does not allow more than 31 requests within a short time period, although we cannot confirm this information 100%.