

Informe sobre la API GameStatistics

Introducción

Este informe tiene como objetivo analizar el rendimiento de la API GameStatistics, que ofrece una serie de endpoints para gestionar los datos de los juegos. La API se ha sometido a una serie de pruebas de carga para medir su capacidad de respuesta y su resistencia ante un gran número de solicitudes. En este informe se presentan los resultados de las pruebas, se identifican los límites de la API para cada endpoint, y se explican las decisiones de desarrollo que se han tomado para mejorar la API.

Metodología

Para realizar las pruebas de carga, se ha utilizado peticiones asíncronas que van aumentando cada segundo, y se ha medido el tiempo de respuesta y el código de estado de cada solicitud. Se ha repetido el proceso varias veces, aumentando el número de solicitudes por segundo, hasta que la API dejó de responder correctamente o se produjo algún error.

Resultados

En esta sección se presentan los resultados de las pruebas de carga para cada endpoint de la API. Se muestran las métricas de tiempo de respuesta, código de estado y número de solicitudes por segundo. También se identifican los límites de la API para cada endpoint, es decir, el número máximo de solicitudes por intervalos de tiempo que la API puede manejar sin dejar de responder correctamente o producir algún error.

1. Endpoint GET/gamedata

Este endpoint permite obtener todos los datos de los juegos almacenados en la base de datos. Se ha observado que este endpoint tiene un rendimiento bastante bajo, ya que el tiempo de respuesta aumenta considerablemente a medida que se incrementa el número de solicitudes por segundo. Además, el endpoint deja de responder correctamente cuando se supera un cierto umbral de solicitudes en un intervalo de tiempo, y el servidor cierra la conexión o devuelve un código de error.

La siguiente tabla muestra las métricas de tiempo de respuesta y código de estado para diferentes valores de solicitudes por segundo:

Solicitudes	Tiempo de respuesta medio (s)	Tiempo de respuesta máximo (ms)	Código de estado más frecuente
210	8	12	200
500	15	22	200
700-800	24-38	44	200
900	45	60	500

Se puede apreciar que el tiempo de respuesta medio crece de forma exponencial a medida que se aumenta el número de solicitudes, lo que indica que el endpoint tiene una baja escalabilidad. Además, se observa que cuando se alcanzan las 900 solicitudes en 45 segundos (Esto varía acorde a los datos que tenga la tabla en la BD, mi caso es así dado que metí muchos registros), el tiempo de respuesta medio se dispara y el código de estado más frecuente pasa a ser 500, lo que significa que el servidor ha encontrado un error interno y no puede procesar las solicitudes.

Por lo tanto, se puede concluir que el límite de la API para este endpoint es de 900 solicitudes en 45 segundos, ya que a partir de ese valor el tiempo de respuesta se vuelve inaceptable y el servidor empieza a fallar. Dependiendo de la cantidad de registro que haya en la DB.

2. Endpoint POST/gamedata

Este endpoint permite insertar un nuevo dato de juego en la base de datos. Se ha observado que este endpoint tiene un rendimiento moderado, ya que el tiempo de respuesta se mantiene relativamente estable a medida que se incrementa el número de solicitudes por segundo. Sin embargo, el endpoint también deja de responder correctamente cuando se supera un cierto umbral de solicitudes en un intervalo de tiempo, y el servidor cierra la conexión o devuelve un código de error. Si aparece una excepción de `Object.Reference` ejecuta el test en modo prueba. Este error suponemos que es debido a los threadings.

La siguiente tabla muestra las métricas de tiempo de respuesta y código de estado para diferentes valores de solicitudes por segundo:

Solicitudes	Tiempo de respuesta medio (s)	Tiempo de respuesta máximo (s)	Código de estado más frecuente
200	1.4	2.4	201
300	2.2	3.8	201
350	3.4	5.8	201
450	5	7	201
500	7.3	8.2	500

Se puede apreciar que el tiempo de respuesta medio crece de forma lineal a medida que se aumenta el número de solicitudes, lo que indica que el endpoint tiene una escalabilidad moderada. Además, se observa que cuando se alcanzan las 500 solicitudes en 7.3, el tiempo de respuesta medio se dispara y el código

de estado más frecuente pasa a ser 500, lo que significa que el servidor ha encontrado un error interno y no puede procesar las solicitudes.

Por lo tanto, se puede concluir que el límite de la API para este endpoint es de 450 solicitudes por segundo, ya que a partir de ese valor el tiempo de respuesta se vuelve inaceptable y el servidor empieza a fallar.

3. Endpoint DELETE/gamedata

Este endpoint permite eliminar un dato de juego de la base de datos. Se ha observado que este endpoint tiene un rendimiento muy alto, ya que el tiempo de respuesta se mantiene muy bajo a medida que se incrementa el número de solicitudes en un intervalo de tiempo. El endpoint se testeó enviando solicitudes con un identificador erróneo, devolviendo un código de estado 404.

La siguiente tabla muestra las métricas de tiempo de respuesta y código de estado para diferentes valores de solicitudes por segundo:

Solicitudes	Tiempo de respuesta medio (s)	Tiempo de respuesta máximo (s)	Código de estado más frecuente
2000	1.8	2.8	404
3000	2.9	3.9	404
3500	3.7	5.2	404
4000	5.9	7.7	404
5000	8.2	12.3	500

Se puede apreciar que el tiempo de respuesta medio crece de forma lineal a medida que se aumenta el número de solicitudes por segundo, lo que indica que el endpoint tiene una escalabilidad alta. Además, se observa que cuando se alcanzan las 5000 solicitudes en 8,3 segundos, el tiempo de respuesta medio se dispara y el código de estado más frecuente pasa a ser 500, lo que significa que el servidor ha encontrado un error interno y no puede procesar las solicitudes.

Por lo tanto, se puede concluir que el límite de la API para este endpoint es de 4000 solicitudes en un intervalo de 7 segundos, ya que a partir de ese valor el tiempo de respuesta se vuelve inaceptable y el servidor empieza a fallar.

4. Endpoint PUT/gamedata

Este endpoint permite actualizar un dato de juego existente en la base de datos. Se ha observado que este endpoint tiene un rendimiento moderado, ya que el tiempo de respuesta se mantiene relativamente estable a medida que se incrementa el número de solicitudes por segundo. Sin embargo, el endpoint también deja de responder correctamente cuando se supera un cierto umbral de solicitudes en un intervalo de tiempo, y el servidor cierra la conexión o devuelve un código de error. Si aparece una excepción de `Object.Reference` ejecuta el test en modo prueba/depuración. Este error suponemos que es debido a los threadings.

La siguiente tabla muestra las métricas de tiempo de respuesta y código de estado para diferentes valores de solicitudes por segundo:

Solicitudes	Tiempo de respuesta medio (s)	Tiempo de respuesta máximo (s)	Código de estado más frecuente
360	2	3.6	204
400	3.1	4.7	204
500	4.4	6.2	500

Se puede apreciar que el tiempo de respuesta medio crece de forma lineal a medida que se aumenta el número de solicitudes por segundo, lo que indica que el endpoint tiene una escalabilidad moderada. Además, se observa que cuando se alcanzan las 400 solicitudes en 3.5 segundos, el tiempo de respuesta medio se dispara y el código de estado más frecuente pasa a ser 500, lo que significa que el servidor ha encontrado un error interno y no puede procesar las solicitudes.

Por lo tanto, se puede concluir que el límite de la API para este endpoint es de 400 solicitudes en 3.7 segundos, ya que a partir de ese valor el tiempo de respuesta se vuelve inaceptable y el servidor empieza a fallar.

5. Endpoint GET/gamedata/id

Este endpoint permite obtener un dato de juego específico de la base de datos, indicando su identificador. Se ha observado que este endpoint tiene un rendimiento alto, ya que el tiempo de respuesta se mantiene bajo a medida que se incrementa el número de solicitudes en un intervalo de tiempo. El endpoint se probó cuando se envían solicitudes con un identificador erróneo, devolviendo un código de estado 404.

La siguiente tabla muestra las métricas de tiempo de respuesta y código de estado para diferentes valores de solicitudes por segundo:

Solicitudes	Tiempo de respuesta medio (s)	Tiempo de respuesta máximo (s)	Código de estado más frecuente
200	2	3.5	404
500	6.7	8.1	404
650	8.2	10.8	404
750	12.3	17	500

Se puede apreciar que el tiempo de respuesta medio crece de forma lineal a medida que se aumenta el número de solicitudes, lo que indica que el endpoint tiene una escalabilidad alta. Además, se observa que cuando se alcanzan las 750 solicitudes en 12 segundos, el tiempo de respuesta medio se dispara y el código de estado más frecuente pasa a ser 500, lo que significa que el servidor ha encontrado un error interno y no puede procesar las solicitudes.

Por lo tanto, se puede concluir que el límite de la API para este endpoint es de 650 solicitudes cada 9 segundos, ya que a partir de ese valor el tiempo de respuesta se vuelve inaceptable y el servidor empieza a fallar.

Conclusiones

En este informe se ha analizado el rendimiento de la API GameStatistics, que ofrece una serie de endpoints para gestionar los datos de los juegos. Se ha identificado los límites de la API para cada endpoint, es decir, el número máximo de solicitudes por segundo que la API puede manejar sin dejar de responder correctamente o producir algún error. Los resultados se han presentado en forma de tablas, y se han explicado las decisiones de desarrollo que se han tomado para testear la API. Dependiendo también de las características del sistema operativo.

Los límites de la API para cada endpoint son los siguientes:

Endpoint	Límite (solicitudes por segundo)
GET/gamedata	900
POST/gamedata	350
DELETE/gamedata	4500
PUT/gamedata	400
GET/gamedata/id	650

Se puede observar que los endpoints que tienen un rendimiento más bajo son los que implican una operación de escritura o una consulta de todos los datos en la base de datos. Los endpoints que tienen un rendimiento más alto son los que implican una operación de lectura o una consulta de un dato específico en la base de datos.

Recomendaciones

En esta sección se ofrecen algunas recomendaciones para mejorar el rendimiento de la API GameStatistics, basándose en los resultados de las pruebas de carga y en las mejores prácticas de desarrollo de APIs. Estas recomendaciones son:

- Implementar una paginación para el endpoint GET/gamedata, de forma que no devuelva todos los datos de los juegos en una sola respuesta, sino que los divida en páginas más pequeñas que se puedan solicitar por separado. Esto reduciría el tiempo de respuesta y el consumo de recursos del servidor y del cliente, y permitiría una mayor escalabilidad del endpoint.
- Implementar un mecanismo de caché para el endpoint GET/gamedata/id, de forma que almacene los datos de los juegos más solicitados en una memoria temporal y los devuelva sin consultar la base de datos cada vez. Esto mejoraría el tiempo de respuesta y la disponibilidad del endpoint, y reduciría la carga de la base de datos.
- Implementar un control de concurrencia para los endpoints POST/gamedata y PUT/gamedata, de forma que evite que se produzcan conflictos o inconsistencias cuando se intenta insertar o actualizar un dato de juego al mismo tiempo desde diferentes solicitudes. Esto evitaría los errores de referencia de objeto y garantizaría la integridad de los datos.
- Implementar un control de errores para todos los endpoints, de forma que se capturen y se manejen adecuadamente las excepciones que puedan ocurrir durante el procesamiento de las solicitudes, y se devuelvan mensajes de error claros y útiles al cliente. Esto facilitaría la depuración y la resolución de problemas, y mejoraría la experiencia de usuario.