

## Pruebas de Carga con Apache JMeter

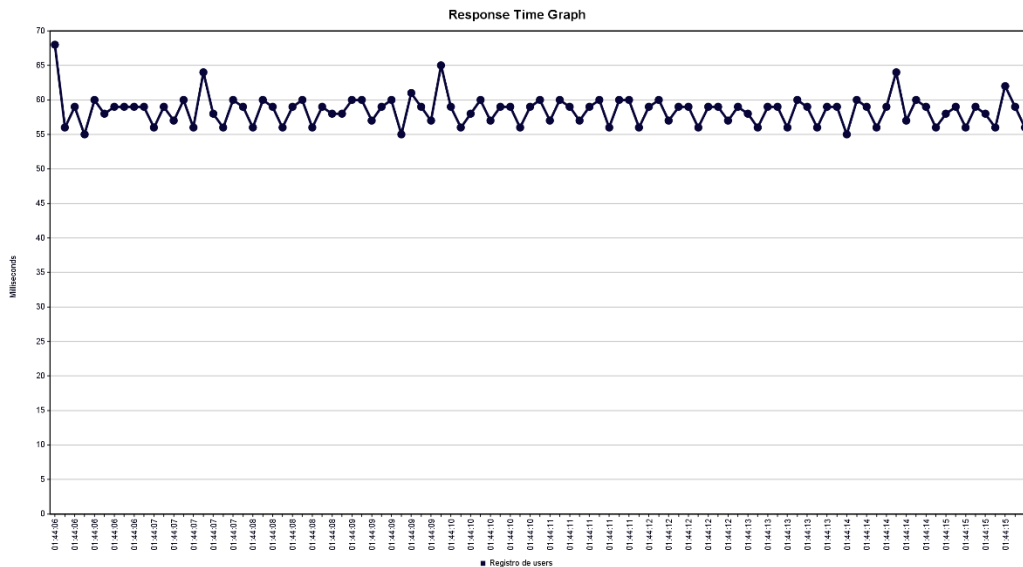
Teniendo en consideración que el software este hecho para una clínica pequeña, no es muy probable que la cantidad de usuarios que ingresen a la aplicación sean mayores a 100. Basándonos en lo anterior, se crearon 3 pruebas de carga con 100 usuarios para 3 distintos end-points, los cuales se presentan a continuación:

### Prueba “users”:

Esta prueba está enfocada al método POST del api “usuarios” que se encarga de registrar a los usuarios dentro del software. La prueba trata de tratar de registrar un nuevo usuario múltiples ocasiones, es decir, simula como 100 usuarios tratan de crear un mismo perfil dentro de la aplicación, por lo que se espera que el software solo permita el registro de un usuario y los demás generen un tipo de error.

La prueba por ejecutar simula 100 usuarios distribuidos uniformemente en 10 segundos (ramp-up = 10s) donde se espera que el tiempo de respuesta sea menor o igual a 1 segundo sea fallo o no.

Al ejecutarse la prueba se obtuvo el siguiente grafico:



Response time graph “users”

Observando la gráfica, se aprecia que el tiempo de respuesta ronda entre los 55 y 70 ms en responder, lo cual es bastante bueno y nos garantiza que el software está preparado para responder en tiempos muy pequeños la posibilidad de sobrecarga en el apartado de registro.

En cuanto a como el software reacciona a los 100 usuarios tratando de registrar el mismo perfil es dejando registrar al primer usuario y los demás lanzándole un mensaje de error y cada respuesta en menos de 1 segundo, lo cual es correcto y era lo esperado.

La posible razón por la cual el software responde correctamente a esta carga se puede responder en dos partes:

El registro del software está preparado para estas situaciones de múltiples usuarios tratando de registrar el mismo perfil, verificando antes de registrar si este perfil se encuentra en la base de datos, lanzando un mensaje de error si es que el perfil existe.

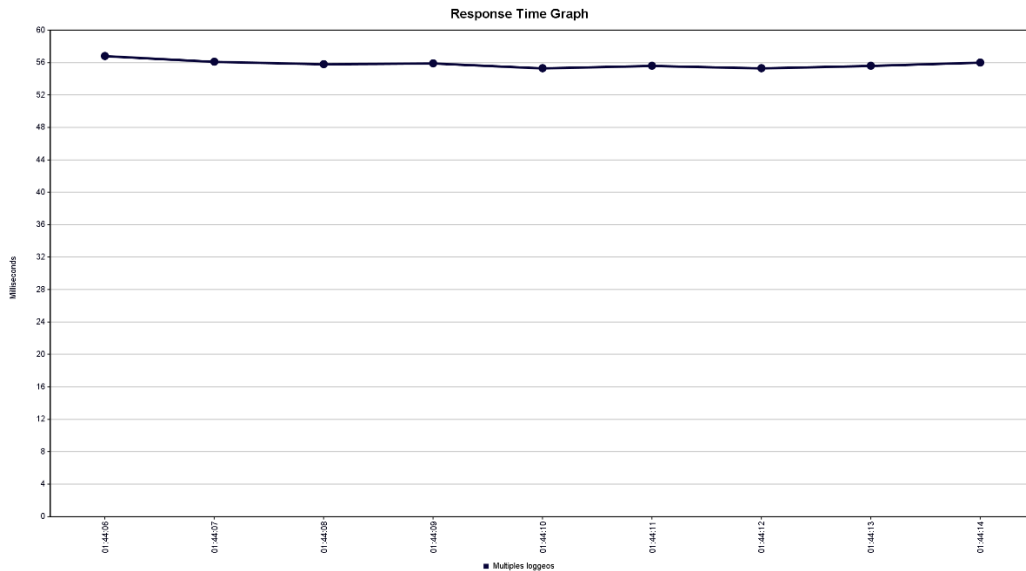
El registro del software, en base lo anterior se puede decir que un nuevo usuario siempre obtendrá un tiempo de respuesta mayor que uno que se encuentra en la base de datos, esto se debe a que el proceso de registro, inicialmente verifica si el usuario esta dentro de la base de datos, posteriormente, si este no se encuentra, deberá ingresar al usuario dentro de la base de datos, generando un proceso extra generando un tiempo de respuesta sutilmente mayor, y en comparación a cuando un usuario ya esta registrado en el software, el registro solo comprueba si esta dentro de la base de datos, una vez encuentra al usuario lanza rápidamente el error. Por esta razón es que en la grafica el mayor tiempo de respuesta alcanzado se encuentra en la primera llamada del registro con un tiempo de 70ms, en cambio los demás intentos de registro fluctúan en tiempos menores a 65ms aproximadamente.

### **Prueba “Login”:**

Esta prueba está enfocada al método POST del api “login” que se encarga de ingresar a los usuarios dentro del software. La prueba trata de tratar de ingresar un usuario múltiples ocasiones, es decir, simula como un usuario trata de ingresar a la plataforma un total de 100 veces, por lo que se espera que el software deje ingresar al usuario todas las ocasiones que quiera.

La prueba por ejecutar simula 100 usuarios distribuidos uniformemente en 10 segundos (ramp-up = 10s) donde se espera que el tiempo de respuesta sea menor o igual a 1 segundo sea fallo o no.

Al ejecutarse la prueba se obtuvo el siguiente grafico:



Response time graph "Login"

Observando la gráfica, se aprecia que el tiempo de respuesta ronda entre los 53 y 58 ms en responder, lo cual es bastante bueno y nos garantiza que el software está preparado para responder en tiempos muy pequeños la posibilidad de sobrecarga en el apartado de Login.

En cuanto a cómo el software reacciona a los 100 usuarios tratando de ingresar a la plataforma es dejando ingresar a cada usuario con las credenciales correctas en menos de 1 segundo, lo cual era lo esperado inicialmente.

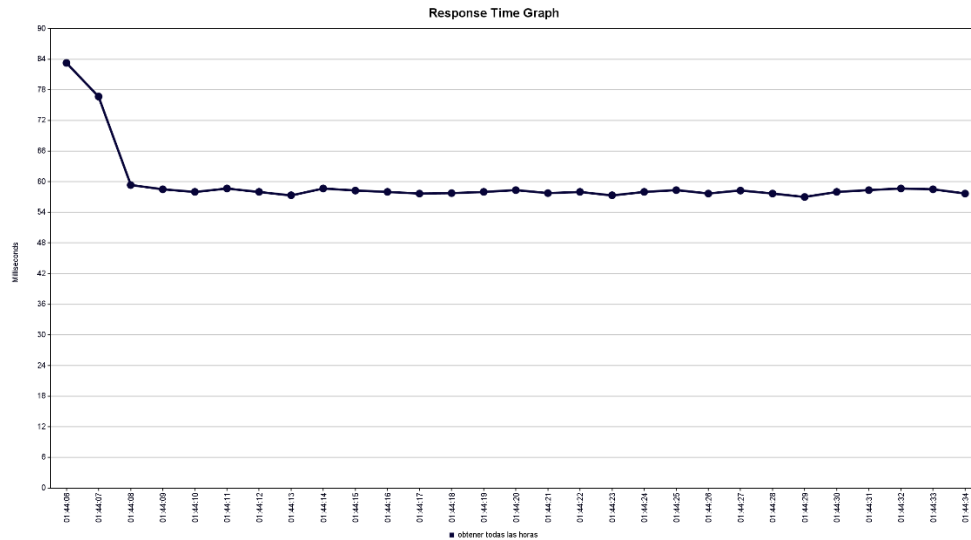
La posible razón por la cual el software responde correctamente a esta carga es porque independiente si las credenciales son correctas o no, el software primero deberá consultar a la base de datos si se encuentra el perfil con las credenciales escritas en la plataforma, si se encuentra el perfil lo deja ingresar y si no le lanza un error. Es por esta razón por la cual el testeo del ingreso de credenciales correctas se puede apreciar que su tiempo es casi lineal y fluctúa siempre de la misma manera entre los 53 y 58 ms, dando a entender que el software puede aguantar grandes cantidades de personas intentando ingresar casi al mismo tiempo en la plataforma.

### Prueba "horas":

Esta prueba está enfocada al método GET del api "events", en específico "getAllevvents" que se encarga de obtener todas las horas que se encuentran dentro de la base de datos. La prueba trata de tratar simular como 100 usuarios piden ver las horas que se encuentran dentro de la base de datos, por lo que se espera que el software cargue o proporcione correctamente las horas todas las veces que se necesite.

La prueba por ejecutar simula 100 usuarios distribuidos uniformemente en 30 segundos (ramp-up = 30s) donde se espera que el tiempo de respuesta sea menor o igual a 1 segundo sea fallo o no.

Al ejecutarse la prueba se obtuvo el siguiente grafico:



*Response time graph "horas"*

Observando la gráfica, se aprecia que el tiempo de respuesta inicial es muy alto y luego baja y ronda entre los 55 y 60 ms en responder, lo cual es bastante bueno y nos garantiza que el software está preparado para responder en tiempos muy pequeños la posibilidad de sobrecarga en el apartado de obtener horas.

En cuanto a cómo el software reacciona a los 100 usuarios tratando de cargar las horas de la plataforma es entregándole a cada usuario que solicita las horas en un tiempo menor de 1 segundo, lo cual era lo esperado inicialmente.

La posible razón por la cual el software responde correctamente a esta carga se puede analizar de la siguiente manera:

El software en las primeras solicitudes obtiene las horas de la base de datos y las debe cargar una por una hasta entregarlas todas, pero posteriormente a la primera solicitud, los servidores guardan esta información y les es más sencillo cargar estas horas, lo que se traduce en un tiempo de respuesta menor.

Ahora bien, tomando en cuenta lo anterior, la razón por la que el software inicialmente tiene un tiempo de respuesta inicial muy alto de casi 84 ms, y luego posteriormente va bajando su tiempo de respuesta hasta llegar a fluctuar de casi manera lineal entre los 55 y 60 ms, dando a entender que el software puede aguantar grandes cantidades de personas intentando cargar las horas del calendario, adaptándose gracias a las características de los servidores.