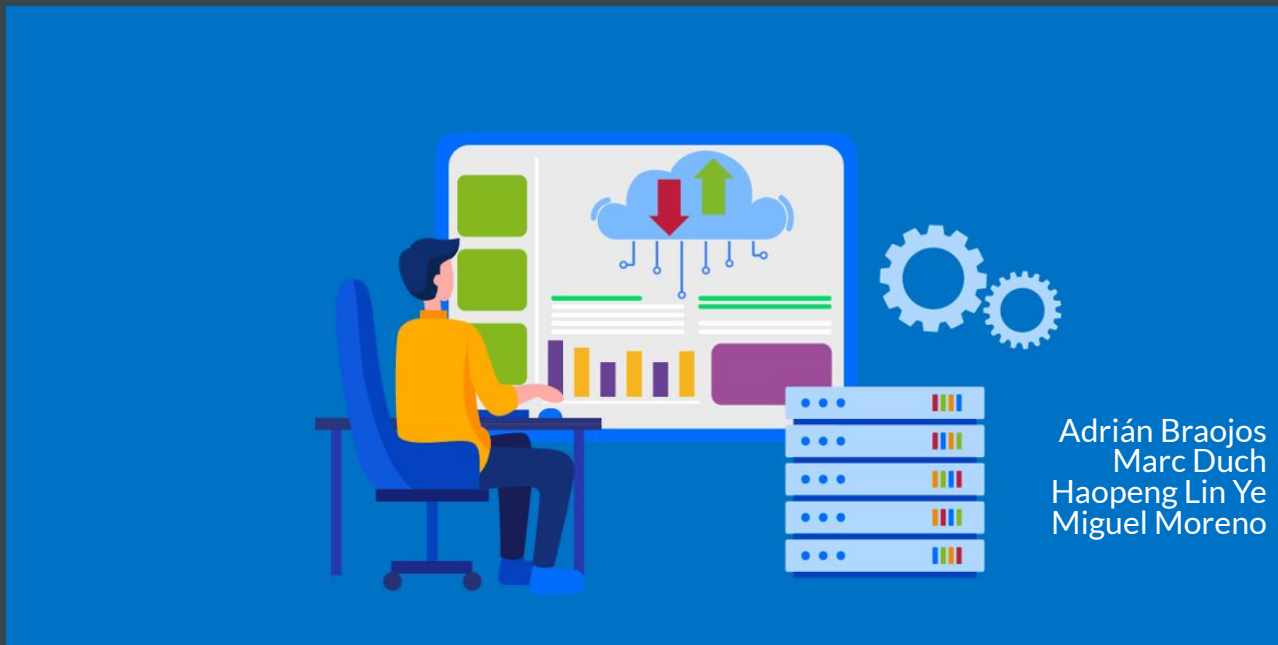


Pruebas de rendimiento para aplicaciones web



Adrián Braojos
Marc Duch
Haopeng Lin Ye
Miguel Moreno

Índice

¿Que es ?

Objetivos

Tipos

¿Por qué lo usamos ?

¿Como se hacen ?

Conclusión

Biografía y reparto de trabajo

¿Qué es ?

Es una técnica de prueba de software no funcional.

Determina cómo el funcionamiento de una aplicación se mantiene bajo una cierta carga de trabajo.

Clave para asegurar calidad de software.

Dependiendo de la metodología:

- Agile realiza las pruebas al principio de cada iteración
- Cascada las realiza al final de cada iteración



Objetivos

- Evaluación de la salida de la aplicación
- La velocidad de procesamiento
- La velocidad de transferencia de datos
- El uso del ancho de banda de la red
- El máximo de usuarios concurrentes
- La utilización de la memoria
- La eficiencia de la carga de trabajo
- Los tiempos de respuesta de los comandos.



Tipos

1. Pruebas de carga

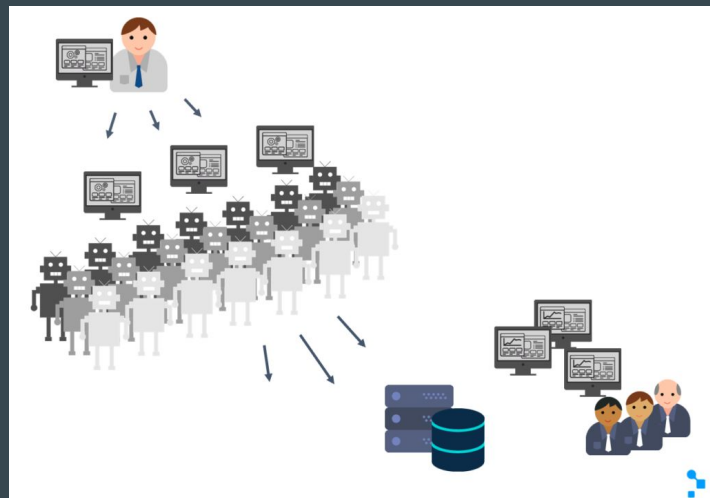
Miden el tiempo de respuesta del software al aumentar la carga de trabajo.

- Identificar cuellos de botella.
- Determinar el número de usuarios concurrentes que puede manejar.

2. Pruebas de resistencia / Soak testing

Miden el tiempo máximo que el software puede soportar una carga de trabajo constante, para determinar cómo de sostenible es a largo plazo.

- Estudiar los KPI (indicadores claves de rendimiento)
- Comprobar que el tiempo de respuesta es constante a largo plazo



Tipos

3. Pruebas de estrés

Miden el rendimiento del software en situaciones donde las cargas de tráfico están por encima de lo previsto.

- Determinar la estabilidad del software
- Entender el punto en el que falla y como se recupera
- Entender que provoca el fallo: CPU, memoria, espacio en el disco, etc.

4. Pruebas de picos

Tipo de prueba de estrés. Mide el rendimiento en un pico de carga de trabajo.

- Determinar si el software puede manejar el pico de forma repetida y rápida.



Tipos

5. Pruebas de volumen / Inundación

Comprueba el rendimiento del software al introducir un gran volumen de datos.

- Analiza posibles pérdidas de datos
- Analizar tiempos de respuesta
- Analizar la fiabilidad de almacenamiento

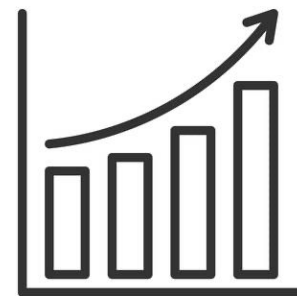


Tipos

6.Pruebas de escalabilidad

Miden el rendimiento del software cuando introduces una cantidad creciente de carga de trabajo.

- Estudiar el comportamiento cuando los datos aumentan o decrecen:
 - Añadir volumen de trabajo, de datos o de usuarios de forma gradual
 - Supervisar el rendimiento durante la prueba




¿Por qué lo usamos?

- Para determinar si la aplicación satisface los requisitos de rendimiento.
- Para localizar los cuellos de botella informáticos dentro de una aplicación.
- Para establecer si los niveles de rendimiento reclamados por un proveedor de software son realmente verdaderos.
- Para comparar dos o más sistemas e identificar el que mejor funciona.
- Para medir la estabilidad en los casos de picos de tráfico.

¿Cómo se hacen?


- 1º Identificar el entorno y las herramientas de prueba
- 2º Definir los criterios de rendimiento aceptables
- 3º Hacer pruebas de planificación y diseño
- 4º Preparar el entorno y las herramientas de prueba
- 5º Ejecutar las pruebas de rendimiento
- 6º Analizar resultados, resolver y volver a probar → 3º

Ejemplo
















 SMARTBEAR
LoadNinja..

DASHBOARDPROJECTSRECORD

Introducing LoadNinja

Upgrade ? 

Recent Load Test Runs


Status	Test run	Date created	Project	Duration (ms)	Errors	Success
✓	10.15.19	10/15/2019, 9:40:49 AM	10.15.19			
✓	Scenario 1	10/14/2019, 7:43:18 PM	Demo			
✓	Scenario 1	10/14/2019, 7:22:40 PM	Demo			
✓	Scenario 1	10/14/2019, 7:12:47 PM	Demo			
✓	10.10.19	10/10/2019, 3:31:10 PM	10.10.19			

Recent Projects

Project	Last Opened
Website Demo	10/15/2019, 3:12:37 PM
10.15.19	10/15/2019, 2:16:58 PM
Demo	10/15/2019, 9:55:28 AM
10.10.19	10/14/2019, 5:04:32 PM
9.17.19	10/10/2019, 9:57:29 AM

Recent Web Tests

Web test	Project	Last opened
PetSearch	Website Demo	10/15/2019, 3:12:20 PM
LoginLogout	Website Demo	10/15/2019, 3:10:55 PM
PetSearch	10.15.19	10/15/2019, 9:30:32 AM
LoginLogout	10.15.19	10/15/2019, 9:25:21 AM
Demo_Script	Demo	10/14/2019, 6:47:29 PM



Ejemplo

The screenshot displays the SmartBear LoadNinja interface. The top navigation bar includes the 'SMARTBEAR LoadNinja' logo, 'DASHBOARD', and 'PROJECTS' tabs. On the right, there are links for 'Upgrade', a help icon, and a user profile icon.

The left sidebar shows a list of test steps for 'webtest1':

- Think Time 711 ms (Current: 711 ms)
- Mouse Move
- Think Time 250 ms (Current: 250 ms)
- Mouse Move
- Think Time 248 ms (Current: 248 ms) - **Selected**
- Mouse Move
- Think Time 146 ms (Current: 146 ms)
- Mouse Move
- Think Time 100 ms (Current: 100 ms)
- Click Event Remove
- 7. Shopping Cart (Total: 1663 ms)
- 8. YAPS PetStore (Total: 0 ms)

The main content area shows the 'ACTIVE TAB' with the URL 'http://petstore.smartload.io/applicationPetstore/shopping/showitem.xhtml'. The application being simulated is 'Yaps Petstore EE 7'. The page displays a 'Shopping Cart' section with the following items:

Item	Quantity	Price	Total	Action
Golden Retriever Tailed	1	\$ 82.0	\$ 82.0	Remove

The total for the cart is \$ 82.0. Below the cart, there is a 'Check Out' button. The page also includes a 'Languages' dropdown and a 'Search' bar.

At the bottom of the page, there is a footer with the text 'Source code on GitHub Antonia Gonçalves' and 'Swagger Contract Conversation [1]'.

Conclusión

Las pruebas de rendimiento son una manera muy buena de determinar si el software cumple con los requisitos de velocidad, y estabilidad bajo diferentes cargas de trabajo.

Pero para obtener los máximos beneficios de ellas se deben seguir los pasos en orden y deben hacerse por un equipo que tenga el conocimiento necesario para entender e identificar las necesidades del cliente así como la funcionalidad de la aplicación.

Si se hacen bien pueden ser muy útiles para desarrollar software estable y útil.

Referencias

- <https://www.microfocus.com/es-es/what-is/performance-testing>
- <https://kinsta.com/es/blog/herramientas-pruebas-rendimiento/>
- <https://www.encora.com/es/blog/pruebas-de-rendimiento-cuando-y-como>
- <https://loadninja.com/>

Reparto de tareas

- Adrián Braojos: Ha hecho parte de la presentación a partir de la información.
- Marc Duch: Ha hecho parte de la presentación a partir de la información.
- Haopeng Lin Ye: Ha buscado la información y referencias.
- Miguel Moreno: Ha hecho parte de la presentación a partir de la información.