

Pruebas no funcionales

Pruebas de carga

Estas pruebas se hacen con el objetivo de determinar y validar la respuesta de la aplicación cuando esta está sometida a una carga de un cierto número de usuarios o de peticiones.

Ejemplo: Verificar si el producto puede soportar la carga de 100 usuarios de forma simultánea. Este resultado se compara con el volumen esperado.

Pruebas de rendimiento

El principal objetivo de este tipo de pruebas no funcionales es calcular la respuesta de la aplicación con diferentes medidas de usuario o peticiones.

Ejemplo: conocer cuál es la respuesta al procesar el ingreso de 10, 100 y 1000 usuarios de forma parametrizada. Este resultado se compara con el resultado esperado.

Pruebas de estrés

Estas pruebas se realizan para encontrar el número de usuarios, peticiones o tiempos que la aplicación puede soportar. Este tipo de pruebas no funcionales son muy semejantes a las pruebas de carga y rendimiento, pero se diferencian en que debemos superar los límites esperados en el ambiente de producción o los límites que fueron determinados en las pruebas.

Ejemplo: encontrar la cantidad de usuarios que soporta de manera simultánea hasta que la aplicación deja de responder (cuelgue o time out), haciéndolo de forma correcta según todas las peticiones.

Petición HTTP

Nombre:

listado

Comentarios

▲ ▼

...

Basic

Advanced

Servidor Web

Protocolo:

http

Nombre de Servidor o IP:

localhost

Puerto:

80

Petición HTTP

POST

▼

Ruta:

http://localhost/html/acceso.html

Codificación del contenido:

☐ Redirigir Automáticamente

☒ Seguir Redirecciones

☒ Utilizar KeepAlive

☐ Usar 'multipart/form-data' para HTTP POST

☐ Cabeceras compatibles con navegadores

Parameters

Body Data

Files Upload

⋮

Enviar Parámetros Con la Petición:

Nombre:	Valor	¿Codificar?	Content-Type	¿Incluir Equals?
---------	-------	-------------	--------------	------------------

Ver Árbol de Resultados

Nombre: Ver Árbol de Resultados

Comentarios

- Escribir todos los datos a Archivo

Nombre de archivo

Navegar...

Log/Most

▲ ▼

Buscar:

☐ Sensible a mayúsculas

☐ Expresión regular

Search

Reset

▲ ▼

Texto ▼

- ✓ listado
- ✓ listado
- ✓ HTTP Request
- ✓ HTTP Request
- ✓ HTTP Request
- ✓ HTTP Request
- ✓ HTTP Request
- ✓ listado
- ✓ listado
- ✓ listado
- ✓ listado
- ✓ listado
- ✓ HTTP Request
- ✓ HTTP Request
- ✓ HTTP Request
- ✓ HTTP Request
- ✓ HTTP Request
- ✓ listado
- ✓ listado

Resultado del Muestreador

Petición

Datos de Respuesta

Request Body

Cabeceras de petición:

```
1 POST http://localhost/html/acceso.html
2
3 POST data:
4
5
6 [no cookies]
7
```

Grupo de Hilos

Nombre: Grupo de Hilos

Comentarios

Acción a tomar después de un error de Muestreador

☒ Continuar ☐ Comenzar siguiente iteración ☐ Parar Hilo ☐ Parar Test ☐ Parar test ahora

Propiedades de Hilo

Número de Hilos 1000

Periodo de Subida (en segundos): 1

Contador del bucle: ☐ Sin fin 1

☒ Same user on each iteration

☐ Retrasar la creación de Hilos hasta que se necesiten

☐ Planificador

Duración (segundos)

Retardo de arranque (segundos)

Gráfico

Nombre:

Comentarios

Escribir todos los datos a Archivo

Nombre de archivo Log/Mostrar sólo: ☐ Escribir en Log ☐ Sólo Errores ☐ Éxitos

Etiqueta	# Muestras	Media	Mediana	90% Line	95% Line	99% Line	Mín	Máx	% Error	Rendimiento	Kb/sec	Sent KB/sec
Estado	2000	346	136	1014	1047	1228	1	1244	0.00%	8.4/sec	29.82	1.44
HTTP Requ...	2000	52	18	155	238	262	0	293	0.00%	8.4/sec	29.79	1.75
Total	4000	199	46	776	1014	1215	0	1244	0.00%	16.7/sec	59.58	3.19

Settings Graph

☐ Guardar la cabecera de la tabla

Column settings

Columns to display: ☒ Media ☐ Mediana ☐ 90% Line ☐ 95% Line ☐ 99% Line ☐ Mín ☐ Máx

Value font: Size: Style: ☒ Draw outlines bar? ☒ Show number grouping? ☒ Value labels vertical?

☐ Column label selection: ☐ Sensible a mayúsculas ☒ Expresión regular

Title

Título del gráfico

Font: Size: Style:

Graph size

☒ Dynamic graph size Anchura Altura

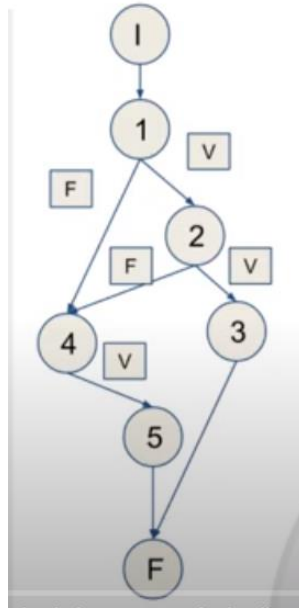
X Axis Y Axis (milli-seconds)

Longitud máxima de la etiqueta del eje X Escala máxima de valores

Pruebas de caja blanca

1. Cobertura de la declaración: Esta técnica garantiza que cada línea del código se pruebe al menos una vez para encontrar más fácilmente el código defectuoso.

- notación de diagramas de flujo: Esta técnica utiliza un grafo dirigido formado por nodos y aristas, donde cada nodo representa un punto de decisión o una secuencia de



afirmaciones.

```

<html>
<head>
</head>
<body>
    <?php
        $mascotisados = mysqli_connect("localhost", "root", "", "mascotisados") or die("Problemas con la conexión");

        $registros = mysqli_query($mascotisados, "select nombre
        from productos where nombre='$_REQUEST[nombre]'") or
        die("Problemas en el select :". mysqli_error($mascotisados));

        if ($reg = mysqli_fetch_array($registros)){ mysqli_query ($mascotisados , "delete from productos where
        nombre='$_REQUEST [nombre]'") or die("Problemas en el select :".mysqli_error ($mascotisados));

        echo "Se efectuó el borrado del producto con el
        nombre digitado.";
        }else
        echo "No existe un producto con ese
        nombre intente nuevamente.";

        mysqli_close ($mascotisados);

    ?>
</body>
</html>
  
```

```

<html>
<head>
</head>
<body>

    <?php
    $mascotisados = mysqli_connect("localhost", "root", "", "mascotisados") or die("Problemas con la conexión");

    $registros = mysqli_query($mascotisados, "select nombre
    from productos where nombre='$_REQUEST[nombre]'" or
    die("Problemas en el select :". mysqli_error($mascotisados));

    if ($reg = mysqli_fetch_array($registros)){ mysqli_query ($mascotisados , "delete from productos where
    nombre='$_REQUEST [nombre]'" or die("Problemas en el select :".mysqli_error ($mascotisados));

    4 echo "Se efectuó el borrado del producto con el
    nombre digitado.";
    }else
    5 echo "No existe un producto con ese
    nombre intente nuevamente.";

    mysqli_close ($mascotisados);

    ?>
</body>
</html>

```

