

Pruebas unitarias - CodeProject.

Introducción, objetivos y justificación.

Introducción.

El documento de pruebas unitarias tiene como finalidad mostrar los procesos de prueba a los que fue expuesto el sistema para el control de inventarios dirigido a la **Empresa Comercializadora de Ropa Infantil Violetas**. Adicional a esto y basándose en la información recolectada, el manual brindará una retroalimentación y permitirá al personal técnico saber los límites del aplicativo.

Objetivo.

Elaborar un documento que especifique dentro de sí las pruebas unitarias elaboradas al sistema para el control de inventarios.

Objetivos específicos.

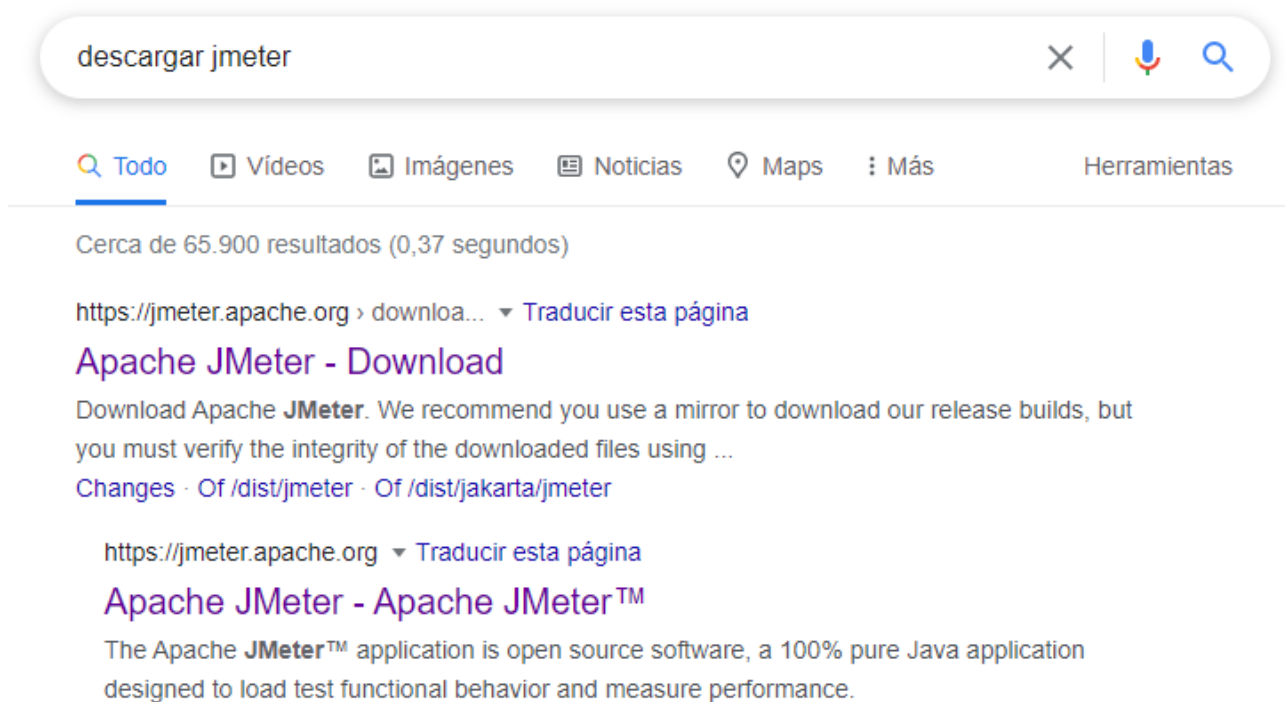
- Elaborar las pruebas necesarias al sistema.
- Documentar, de forma muy detallada y gráfica, la información que arrojaron las distintas pruebas.
- Resumir, a través de una retroalimentación, los resultados de las pruebas y las conclusiones que dejó cada una de ellas.

Justificación.

El documento de pruebas unitarias nos permitirá conocer los límites del sistema. Esto con el fin de saber qué áreas del aplicativo hay que reforzar. Del mismo modo, la función de este documento también es documentar todas y cada una de las pruebas unitarias realizadas al sistema.

Instalación JMeter.

- El primer paso para descargar e instalar **JMeter** consiste en abrir su navegador de preferencia y escribir **Descargar JMeter**.
(https://jmeter.apache.org/download_jmeter.cgi).



Primer paso.

- Dentro del sitio web oficial de **JMeter**, encontraremos una opción que dirá **apache-jmeter-5.4.1.zip**. Este link permitirá que la descarga del archivo con extensión .zip empiece.

Apache JMeter 5.4.1 (Requires Java 8+)

Binaries

[apache-jmeter-5.4.1.tgz sha512 pgp](#)
[apache-jmeter-5.4.1.zip sha512 pgp](#)

Segundo paso.

- Cuando haya descargado nuestro archivo .zip, procederemos a extraer su contenido, que en este caso es una carpeta llamada **apache-jmeter-5.4.1** y pegarla en la raíz de nuestro disco duro.
- Antes de abrir el ejecutable de la aplicación, es necesario instalar el **JDK de Java**, así que debe dirigirse a su navegador, y escribir **Descargar java se jdk 17**. (<https://www.oracle.com/java/technologies/downloads/#jdk17-windows>).
- Ubicado en el sitio oficial de Oracle, debe seleccionar su sistema operativo, y descargar el instalador.

Java SE Development Kit 17 downloads

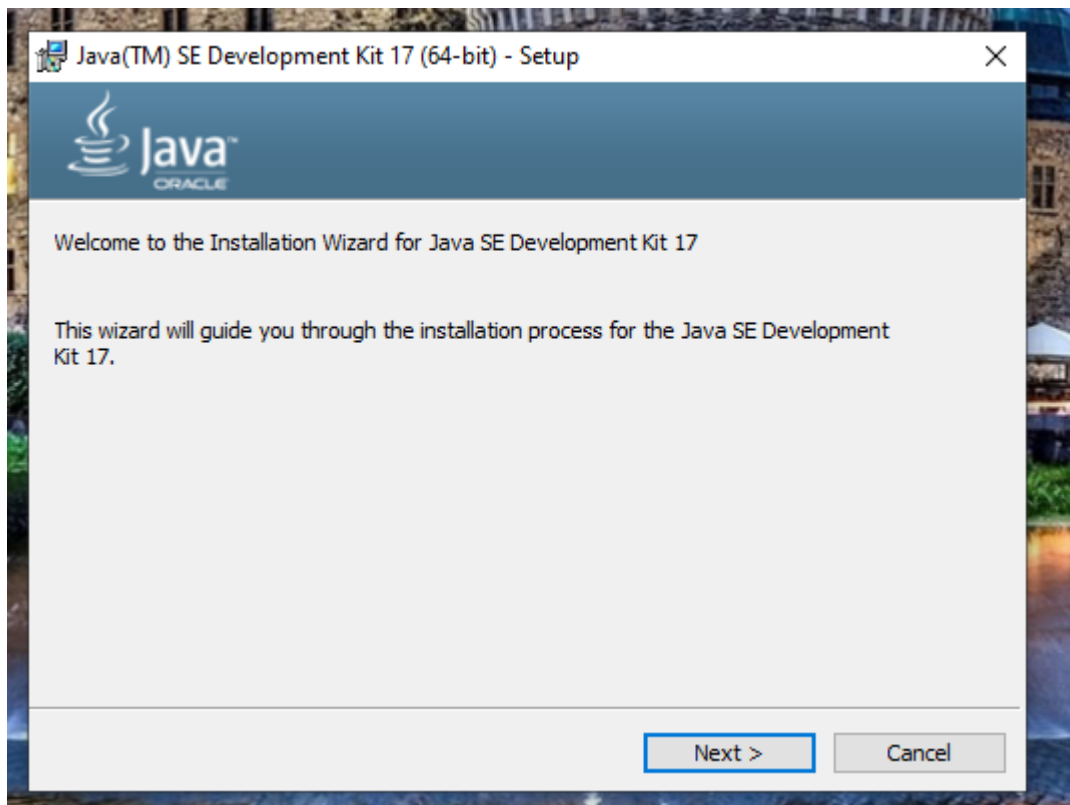
Thank you for downloading this release of the Java™ Platform, Standard Edition Development Kit (JDK™). The JDK is a development environment for building applications and components using the Java programming language.

The JDK includes tools for developing and testing programs written in the Java programming language and running on the Java platform.

Documentation Download		
Linux	macOS	Windows
Product/file description	File size	Download
x64 Compressed Archive	170.64 MB	https://download.oracle.com/java/17/latest/jdk-17_windows-x64_bin.zip (sha256 🔗)
x64 Installer	151.99 MB	https://download.oracle.com/java/17/latest/jdk-17_windows-x64_bin.exe (sha256 🔗)
x64 MSI Installer	150.88 MB	https://download.oracle.com/java/17/latest/jdk-17_windows-x64_bin.msi (sha256 🔗)

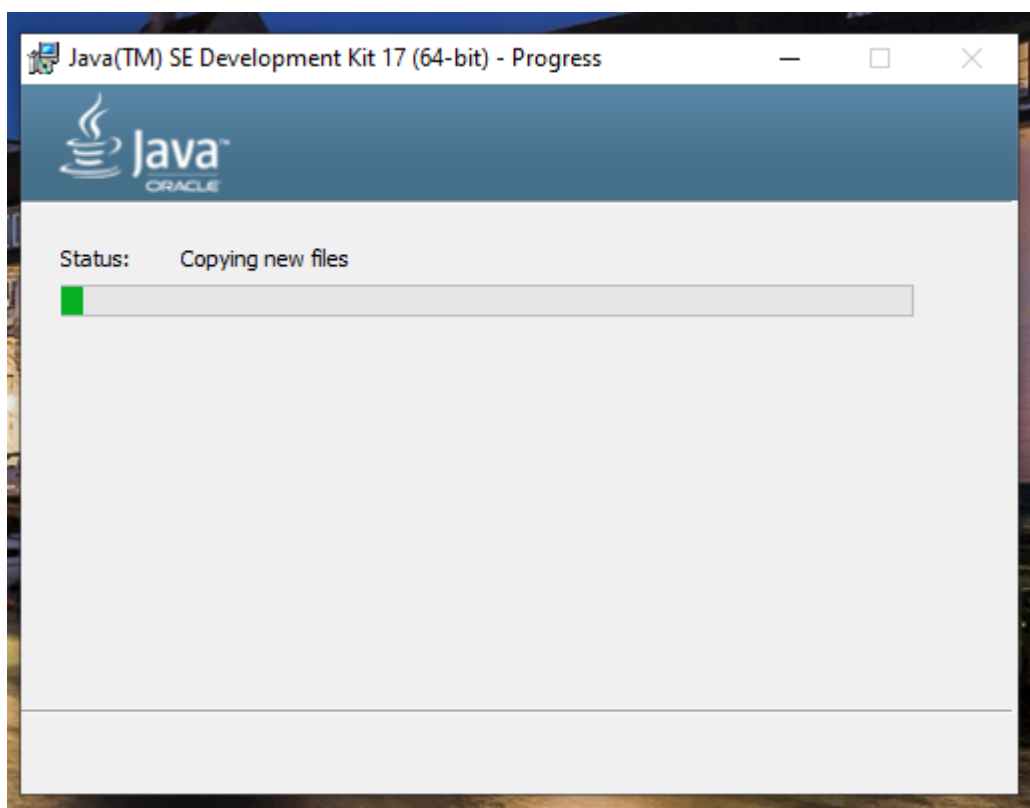
Tercer paso.

- Cuando se haya descargado el instalador, debe proceder a abrirlo. Dar 'clic' **Siguiente** en todo.



Cuarto paso.

- El instalador empezará a instalar todos los componentes en su equipo de computo.



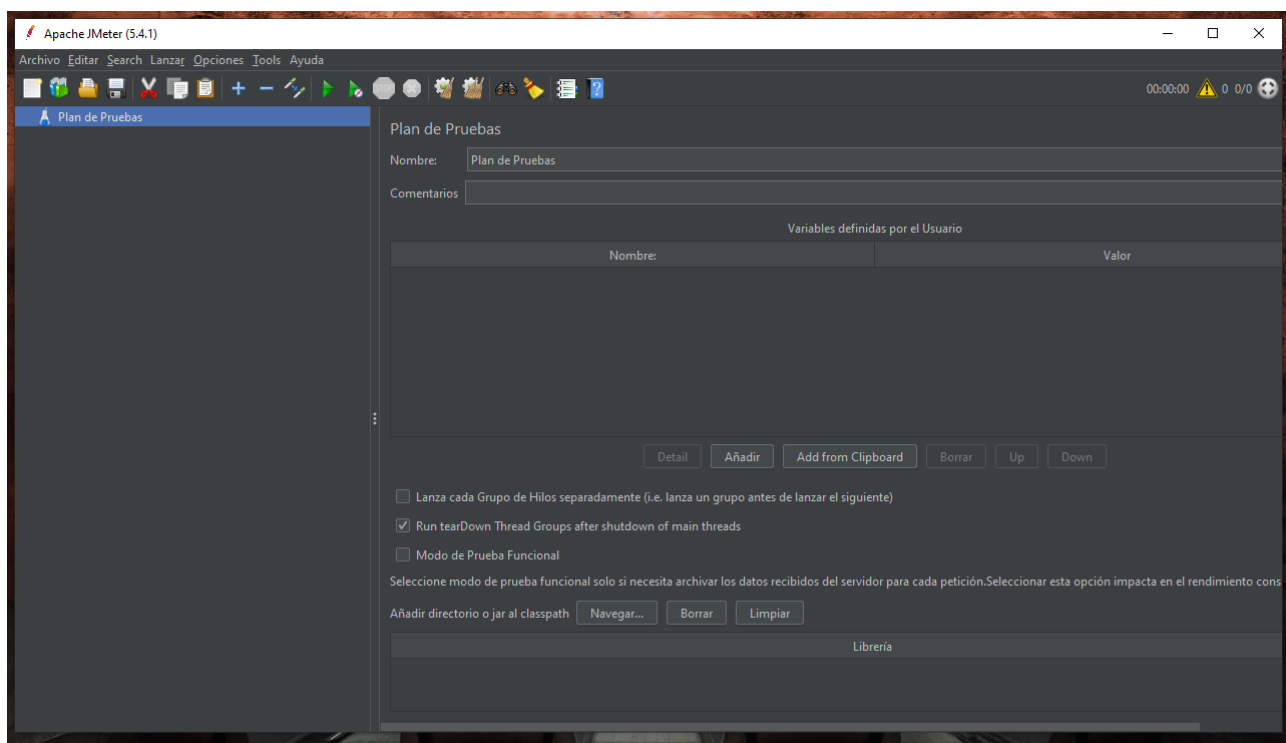
Quinto paso.

- Una vez completada la instalación, deberá mostrarse una interfaz que le notifique que la instalación se hizo satisfactoriamente.



Sexto paso.

- Bien, ahora que ya instaló el **JDK de Java** sólo queda abrir **JMeter** y comprobar que funcione. Diríjase a la carpeta que había descargado previamente llamada ***apache-jmeter-5.4.1***, después ingresar a la carpeta llamada ***bin*** y ejecutar el archivo llamado ***ApacheJMeter.jar***.



Séptimo paso.

- Con esto habrá finalizado la instalación de **JMeter** en su equipo de computo.

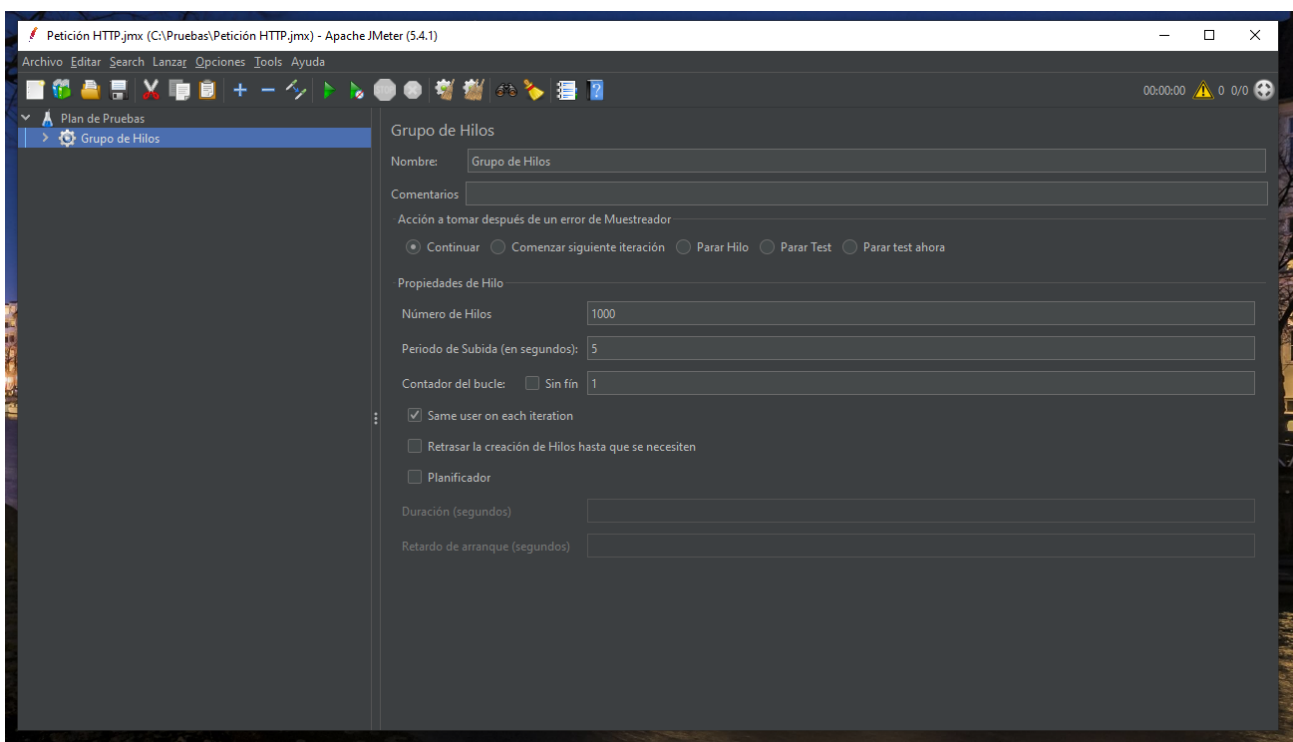
Configuración JMeter.

Para hacer las pruebas de carga y estrés en **JMeter** es importante realizar algunos ajustes.

- La primera configuración que debe tenerse en cuenta es la siguiente. Ubicado en el apartado que dice **Plan de pruebas**, que se encuentra en el panel izquierdo, dar 'clic' derecho > **Añadir > Hilos (Usuarios) > Grupo de Hilos**. Debe quedar una interfaz como la siguiente.



Es importante destacar que en el **Grupo de Hilos** usted debe ajustar el **Número de Hilos (Usuarios)** y el **Periodo de Subida**.



Primer paso.

- El siguiente paso es dar 'clic' derecho sobre **Grupo de Hilos > Añadir > Muestreador > Petición HTTP**. Cuando lo haya creado, debe salir una interfaz como la que se mostrará a continuación.
 - Debe configurar en el **Protocolo** como **http**.

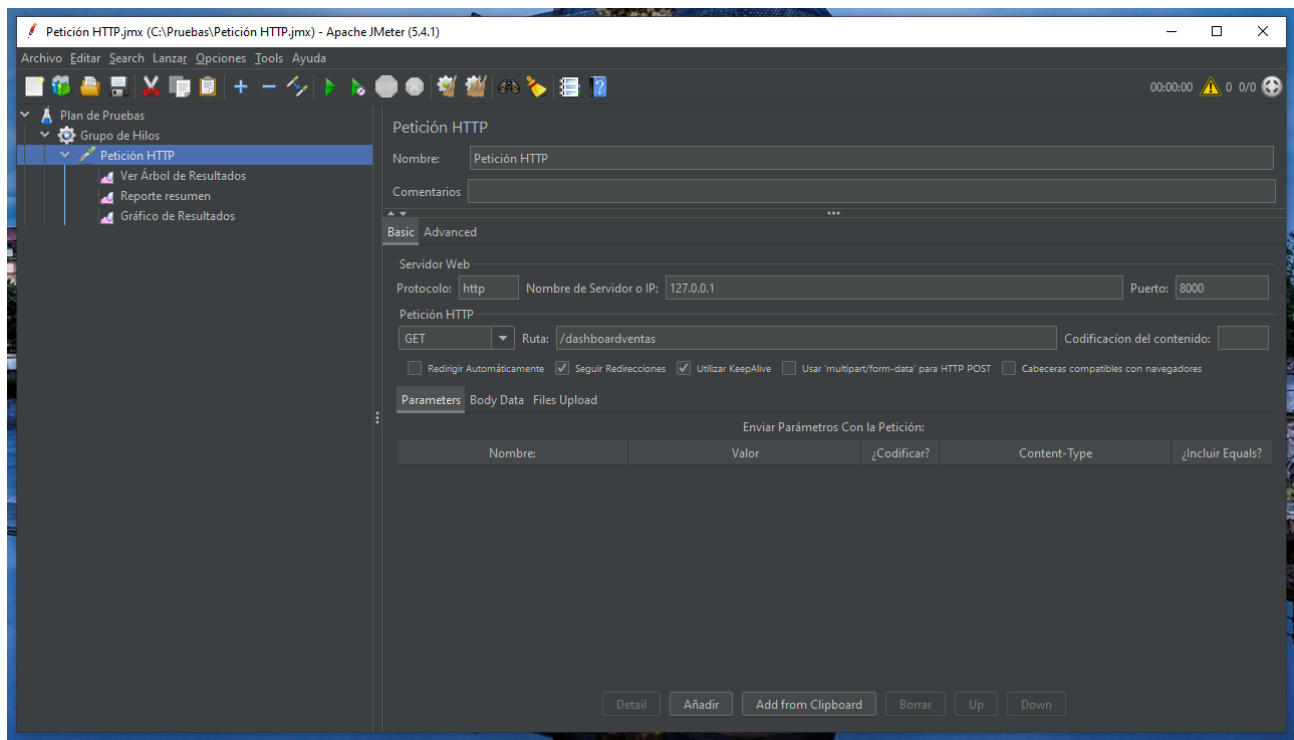
- El **Nombre de Servidor o IP** debe ser igual al localhost de la aplicación, en este caso el localhost es 127.0.0.1.
- Después de esto, debe configurarse el **Puerto**, y éste se ve también en la URL del navegador.
- En la **Petición HTTP** debe seleccionarse el método **GET**.
- La **Ruta** debe ser igual al formulario que deseamos analizar.

The screenshot shows the 'Basic' tab of the JMeter configuration interface. Under 'Servidor Web', the 'Protocolo' is set to 'http', 'Nombre de Servidor o IP' is '127.0.0.1', and 'Puerto' is '8000'. In the 'Petición HTTP' section, the method is 'GET' and the 'Ruta' is '/dashboardventas'. Below this, there are checkboxes for 'Redirigir Automáticamente' (unchecked), 'Seguir Redirecciones' (checked), 'Utilizar KeepAlive' (checked), 'Usar 'multipart/form-data' para HTTP POST' (unchecked), and 'Cabeceras compatibles con navegadores' (unchecked). At the bottom, the 'Parameters' tab is selected, showing a table header for 'Enviar Parámetros Con la Petición:' with columns: 'Nombre:', 'Valor', '¿Codificar?', 'Content-Type', and '¿Incluir Equals?'.

Segundo paso.

- Continuando con la configuración del **JMeter**, usted debe volver al panel izquierdo, dar 'clic' derecho sobre **Petición HTTP > Añadir > Receptor > Ver Árbol de Resultados**.
- De nuevo, usted debe volver al panel izquierdo, dar 'clic' derecho sobre **Petición HTTP > Añadir > Receptor > Reporte Resumen**.
- Y por último, vuelva a dar 'clic' derecho sobre **Petición HTTP > Añadir > Receptor > Gráfico de Resultados**.

Si ha seguido los pasos de esta guía correctamente, usted debería tener una interfaz como esta:



Tercer paso.

- Finalmente, con el **Grupo de Hilos** configurado, la **Petición HTTP** lista y los **receptores** preparados, usted debe dirigirse a la parte superior, en el ícono de color verde que dice **Arrancar**, y ejecutar las pruebas de carga y estrés.
- Con este paso habría finalizado la configuración de **JMeter**.

Pruebas de carga y estrés.

A continuación, se realizarán las pruebas de carga y estrés al aplicativo para el control de inventarios realizado por CodeProject. Estas pruebas intentan medir cuán alto es el porcentaje de error que tiene el aplicativo al momento de recibir cierto número de peticiones en un tiempo de respuesta determinado.

La herramienta que se empleará para realizar las pruebas será **JMeter**.

Primera prueba.

- Tomaremos como formulario de prueba el **dashboard de ventas** del aplicativo. La primera prueba consiste en trabajar con 6 grupos de hilos distintos. Cada uno de ellos tendrá un periodo de subida de 1 segundo.

Número de Hilos	Periodo de Subida (En segundos)	Porcentaje de Error	Rendimiento
1	1	0,00%	7,8/sec
10	1	0,00%	11,0/sec
100	1	0,00%	87,0/sec
1000	1	59,82%	21,5/sec
5000	1	90,90%	858,4/sec
10000	1	96,13%	1291,5/sec

Conclusión primera prueba.

De la primera prueba podemos inferir que a partir del usuario número mil, el aplicativo comienza a tener un elevado porcentaje de error y también el rendimiento comienza a disminuir. Cada vez que aumenta el número de usuarios el aplicativo responde más lento y

eventualmente ninguna petición podrá ser recibida. En base a esto, consideramos oportuno aumentar los segundos de subida a cada grupo de hilos a 3 segundos.

Segunda prueba.

- De nuevo tomaremos el dashboard de ventas como objeto de prueba. En este segundo testeo seguiremos trabajando con los mismos 6 grupos de hilos, pero ahora el periodo de subida será de 3 segundos.

Número de Hilos	Periodo de Subida (En segundos)	Porcentaje de Error	Rendimiento
1	3	0,00%	2,6/sec
10	3	0,00%	3,7/sec
100	3	0,00%	33,6/sec
1000	3	9,00%	201,2/sec
5000	3	73,28%	486,9/sec
10000	3	94,99%	1358,1/sec

Conclusión segunda prueba.

- En esta segunda prueba podemos inferir que el aumentar el periodo de subida a 3 segundos mejoró el rendimiento del aplicativo. El porcentaje de error disminuyó notablemente para el grupo de 1000 usuarios. Pasó de tener en la primera prueba un porcentaje de error del 59% a un casi que inmejorable 9%. Como medida ante esto, consideramos que el aumento de los segundos de subida ayudaría a que el aplicativo pueda recibir cada vez más peticiones.

Tercera prueba.

- Para esta tercera prueba seguiremos haciendo uso del dashboard de ventas. En este tercer ensayo aumentaremos el periodo de subida a 5 segundos.

Número de Hilos	Periodo de Subida (En segundos)	Porcentaje de Error	Rendimiento
1	5	0,00%	200,0/sec (?)
10	5	0,00%	2,2/sec
100	5	0,00%	20,2/sec
1000	5	0,00%	199,2/sec
5000	5	71,72%	487,2/sec
10000	5	91,96%	696,1/sec

Conclusión tercera prueba.

- Esta tercera prueba muestra un gran avance ya que el aplicativo puede recibir 1000 peticiones con un porcentaje de error de 0%. Además de esto, el rendimiento aumentó en muchos de los grupos. Aunque el hilo de un usuario tuvo un rendimiento extraño, los demás grupos tuvieron un desempeño excelente. La meta ahora es hacer que a 5000 peticiones, el sistema tenga menos porcentaje de error, por lo cual aumentaremos el periodo de subida a 7 segundos.

Cuarta prueba.

- Para esta prueba consideramos que aumentar el periodo de subida a 7 segundos mejorará el rendimiento del aplicativo. Seguiremos haciendo uso del dashboard de ventas.

Número de Hilos	Periodo de Subida (En segundos)	Porcentaje de Error	Rendimiento
-----------------	---------------------------------	---------------------	-------------

1	7	0,00%	250,0/sec (?)
10	7	0,00%	1,6/sec
100	7	0,00%	14,4/sec
1000	7	0,00%	143,0/sec
5000	7	84,82%	662,9/sec
10000	7	89,30%	793,0/sec

Conclusión cuarta prueba.

- La cuarta prueba refleja muchos avances, sin embargo, causó sorpresa que el porcentaje de error aumentara aproximadamente un 10% con respecto a la prueba pasada. Consideramos que este cambio puede estar relacionado con la máquina en la que se están haciendo las pruebas, pero continuaremos con la quinta prueba.

Quinta prueba.

- La quinta prueba consiste en aumentar el periodo de subida a 10 segundos. Utilizaremos el dashboard de ventas de nuevo para hacer esta prueba.

Número de Hilos	Periodo de Subida (En segundos)	Porcentaje de Error	Rendimiento
1	10	0,00%	250,0/sec (?)
10	10	0,00%	1,1/sec
100	10	0,00%	10,1/sec
1000	10	0,00%	100,1/sec

5000	10	66,62%	402,1/sec
10000	10	89,44%	808,5/sec

Conclusión quinta prueba.

- En esta prueba hubo una disminución de errores considerable al momento de recibir 5000 peticiones. El aplicativo se mostró más estable y pasó de una tasa de error del 84% a un 66% muy aceptable.

Sexta prueba.

- La sexta y última prueba consiste en aumentar el periodo de subida de 10 segundos a 15. Esperamos que con este aumento en el periodo de subida el aplicativo tenga menos de 50% de errores a la hora de recibir peticiones.

Número de Hilos	Periodo de Subida (En segundos)	Porcentaje de Error	Rendimiento
1	15	0,00%	200,0/sec (?)
10	15	0,00%	44,4/min (?)
100	15	0,00%	6,7/sec
1000	15	0,00%	66,8/sec
5000	15	48,58%	286,0/sec
10000	15	71,78%	442,3/sec

Conclusión final.

Consideramos que elaborar estas pruebas nos permitieron, de cierta manera, dimensionar la complejidad de hacer de un sistema una aplicación con un perfecto balance entre la rapidez y el rendimiento. En conclusión, debemos seguir aumentando el periodo de subida para que el aplicativo a las 5000 peticiones tenga una tasa de 0% de error.

Pruebas de rendimiento, accesibilidad y optimización de motores de búsqueda.

En este apartado se realizarán las pruebas de rendimiento, accesibilidad y optimización de motores de búsqueda al aplicativo para el control de inventarios realizado por CodeProject. Estas pruebas intentan medir cuán rápido, optimizado y accesible es el aplicativo.

La herramienta que se empleará para realizar las pruebas será **Lighthouse**.

Rendimiento.

En el apartado de rendimiento, la herramienta Lighthouse informó que el aplicativo tiene un rendimiento total de 100 puntos. Dentro de los factores que destacan está la velocidad y el bajo tiempo de renderizado de las imágenes.

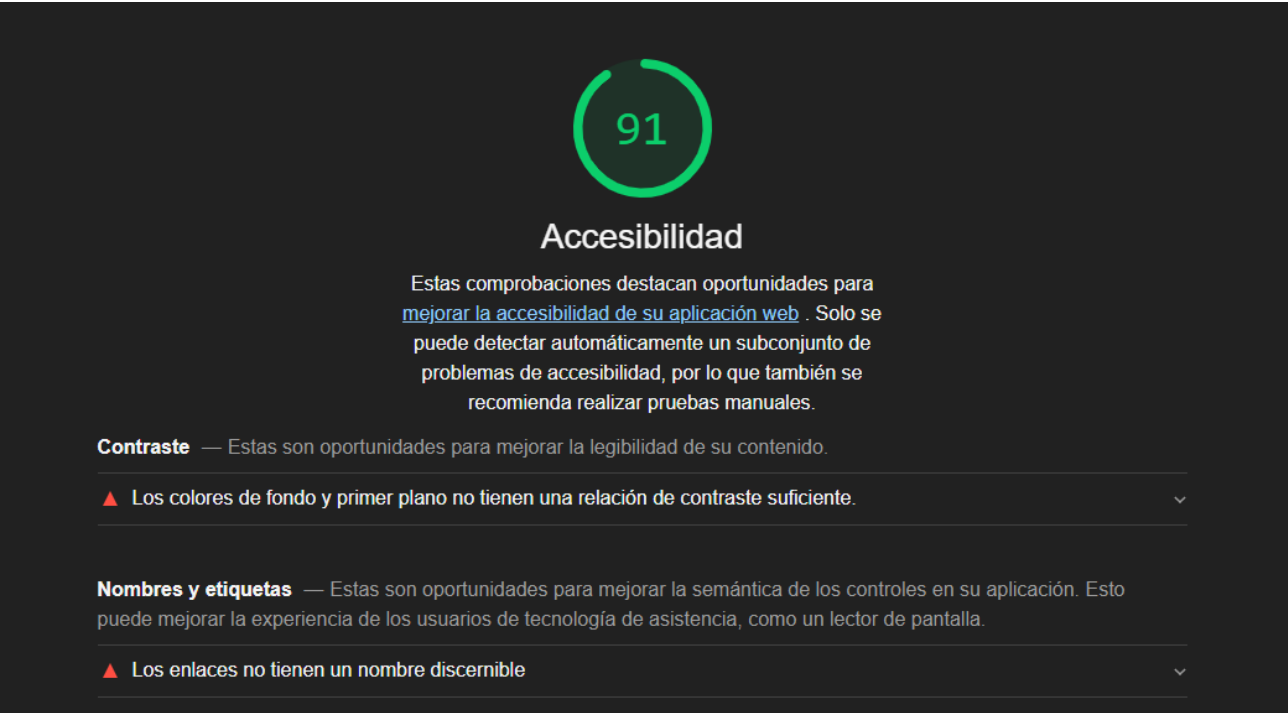


Rendimiento.

Accesibilidad.

En la sección de accesibilidad el aplicativo logró obtener un estimado de 91 puntos. Sin embargo, la herramienta nos da algunos consejos, como por ejemplo que los colores deben

contrastar un poco para que el usuario final pueda ver con claridad las divisiones de la página.



Accesibilidad

Buenas prácticas.

El aplicativo está calificado con 93 puntos de 100 posibles en el área de mejores o buenas prácticas. Como consejos, la herramienta nos dice que configurar más protocolos para evitar ataques de malware.



Mejores prácticas

Confianza y seguridad

○ Asegúrese de que CSP sea eficaz contra ataques XSS

General

▲ Los errores del navegador se registraron en la consola

Auditorías aprobadas (16)

Buenas prácticas.

SEO (Search Engine Optimization).

El SEO se encarga de posicionar nuestro sitio web basándose en la semántica que empleemos. La herramienta **Lighthouse** nos dio una puntuación de 89 puntos y nos recomendó emplear más metadatos que permitan posicionar mejor nuestro sitio web.



SEO

Estas comprobaciones garantizan que su página esté optimizada para la clasificación de los resultados de los motores de búsqueda. Hay factores adicionales que Lighthouse no verifica y que pueden afectar su clasificación de búsqueda. [Obtenga más información](#).

Prácticas — recomendadas de **contenido** — Formatee su HTML de una manera que permita a los rastreadores comprender mejor el contenido de su aplicación.

▲ El documento no tiene una meta descripción

SEO (Search Engine Optimization)

Conclusión.

Como media obtuvimos una puntuación de 93 de 100 puntos posibles. Así que, lo ideal es reforzar el área del SEO empleando más metadatos y utilizando una semántica idónea para que nuestra puntuación media aumente.

