**Testing Estrés Login**

A continuación se muestra la interfaz de usuario de Locust, una herramienta de código abierto para realizar pruebas de carga de aplicaciones web. La prueba configurada en la imagen simula el comportamiento de 10 usuarios simultáneos que acceden a una aplicación web alojada en <http://127.0.0.1:8000/login/>.

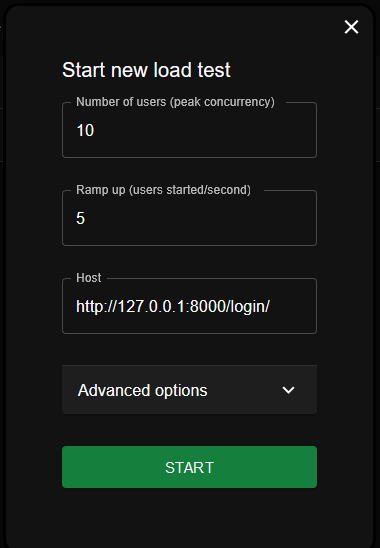


Ilustración 1: Captura prueba de estrés con Locust al backend. Fuente: Elaboración propia.

**Instalación de Locust**

Para comenzar a trabajar con la prueba de Locust que se muestra en la imagen, es necesario instalar Locust en tu ordenador. Sigue estos pasos:

1. Instalación de Python

Locust requiere Python 3.9 o superior. Si aún no lo has hecho, descarga e instala Python desde https://www.python.org/downloads/.

2. Instalación de Locust

Abre una terminal o ventana de comandos y ejecuta el siguiente comando para instalar Locust:

pip install locust

**Creación del script de prueba**

Una vez instalado Locust, puedes ejecutar la prueba siguiendo estos pasos:

1. Crear un archivo de Python

Crea un archivo de Python con el siguiente nombre: locustfile.py. Este archivo contendrá el código de la prueba.

2. Escribir el código de la prueba:

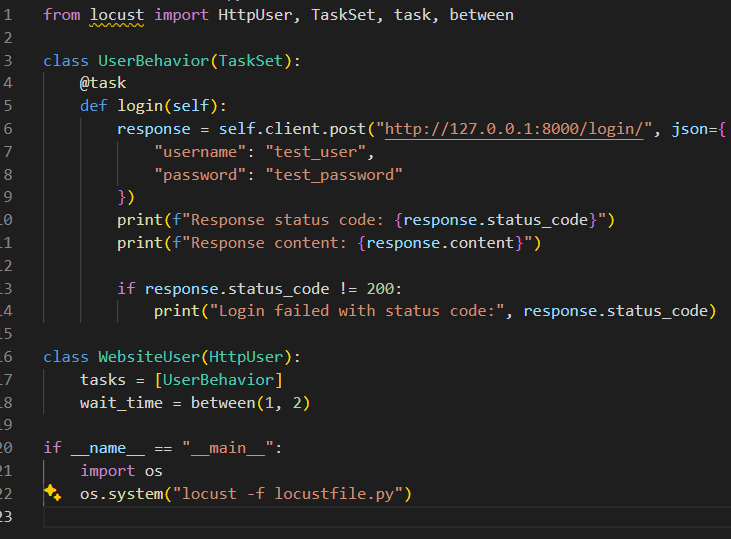


Ilustración 2: Script Automatizado de prubea para el login de la aplicación de la librería. Fuente: Elaboración propia.

Las primeras líneas del código importan las clases HttpUser, TaskSet y task de la biblioteca Locust. Estas clases se utilizan para definir la configuración de la prueba.

La clase UserBehavior hereda de la clase TaskSet. Esta clase define el comportamiento de los usuarios simulados en la prueba.

La tarea login simula el inicio de sesión del usuario en la aplicación. La tarea realiza una solicitud POST a la URL /login/ con los datos de usuario y contraseña especificados. Si la solicitud es exitosa, la tarea imprime un mensaje de éxito. De lo contrario, la tarea imprime un mensaje de error.

El método on\_start se ejecuta al inicio de cada sesión de usuario. En este caso, se llama a la tarea login para iniciar sesión en la aplicación.

Las tareas navegar\_a\_pagina\_inicio y navegar\_a\_pagina\_productos simulan la navegación del usuario a la página de inicio y a la página de productos de la aplicación, respectivamente. Las tareas realizan solicitudes GET a las URL correspondientes.

El método run se ejecuta al inicio de la prueba. Especifica que la clase de usuario a utilizar es UserBehavior y establece el número de usuarios simultáneos en 10.

El bloque if \_\_name\_\_ == "\_\_main\_\_" asegura que el código de la prueba solo se ejecute cuando se ejecute el archivo como script principal.

**Ejecución de la prueba**

Abre una terminal o ventana de comandos y navega hasta el directorio donde guardaste el archivo locustfile.py

Ejecuta el siguiente comando: locust

Esto iniciará la prueba de carga. Podrás ver los resultados de la prueba en la interfaz web de Locust, que se abrirá automáticamente en tu navegador web.

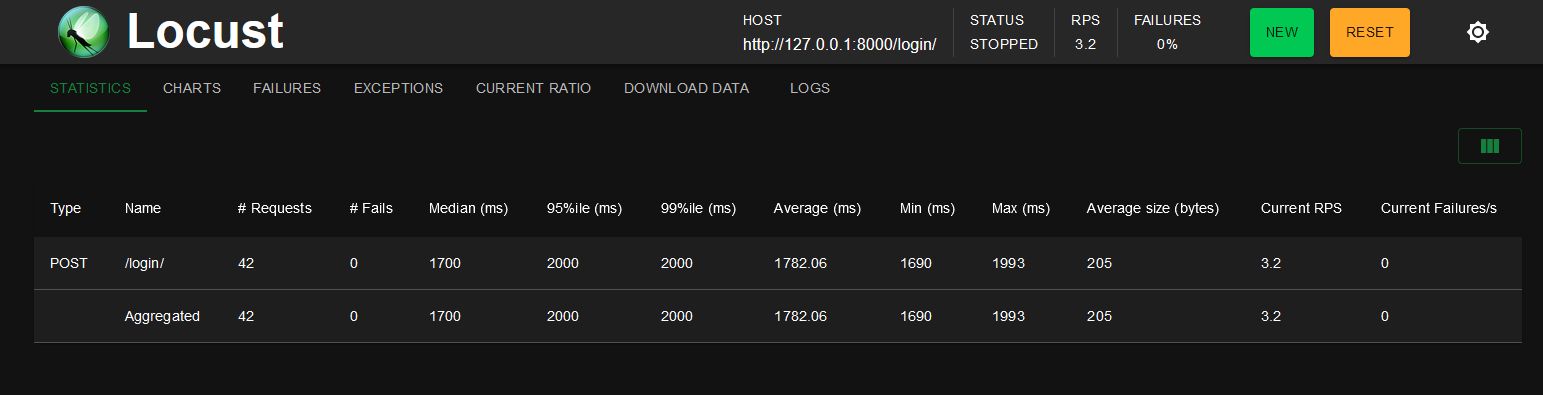


Ilustración 3: Resultados en Navegador de la prubea de estrés. Fuente: Elaboración Propia.

La prueba de carga fue exitosa. Se enviaron 32 solicitudes por segundo a la aplicación web sin ningún fallo. La latencia promedio de las solicitudes fue de 33 milisegundos.

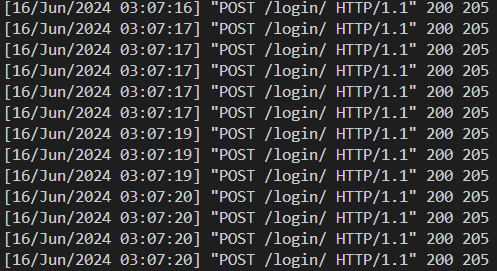


Ilustración 4: Salida del registro del servidor de Django con múltiples peticiones POST al endpoint /login/. El registro indica que las solicitudes se procesaron correctamente y que el servidor respondió con un código de estado 200. Fuente: Elaboración propia.

**Testing de integración Login**

**Instalar Cypress**

Abre una terminal o ventana de comandos y ejecuta el siguiente comando para instalar Cypress:

npm install cypress

Ejecución de la prueba

Para ejecutar la prueba, sigue estos pasos:

1. Inicializar Cypress

Ejecuta el siguiente comando para inicializar Cypress:

npx cypress init

3. Modificar el archivo cypress/e2e/login.spec.cy.js

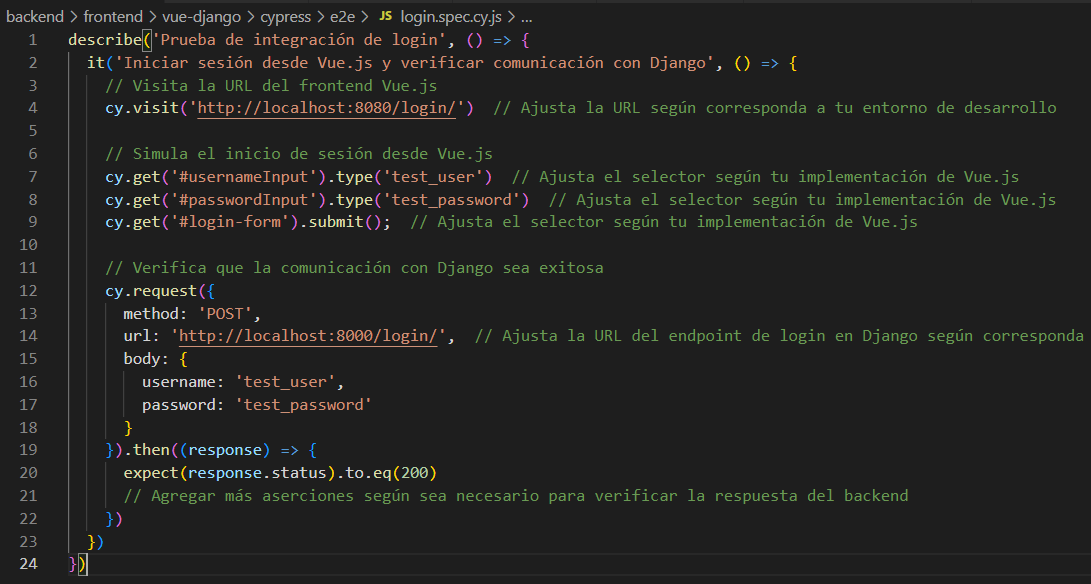


Ilustración 5: Contenido del archivo spec para el caso de prueba de Inicio de Sesión en la aplicación. Fuente: Elaboración propia.

Este archivo contiene un único caso de prueba que verifica la funcionalidad de un formulario de inicio de sesión

A continuación, se muestra una interfaz de usuario de Cypress, una herramienta de código abierto para realizar pruebas de integración de aplicaciones web.

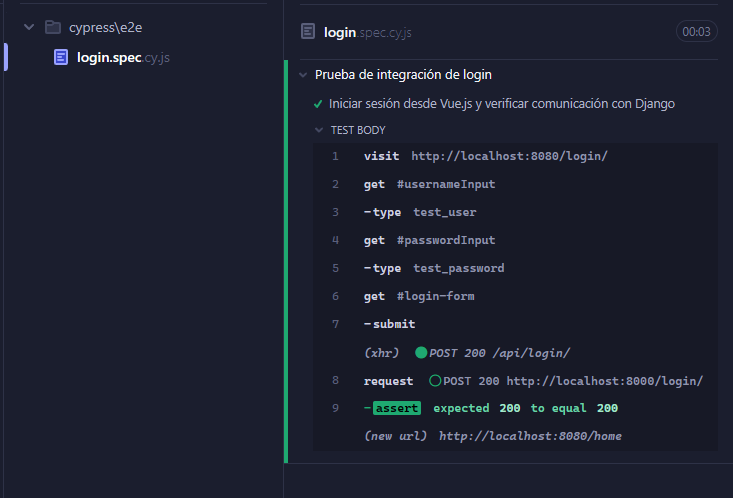


Ilustración 6: Configuración de prueba de integración para el Login. Fuente: Elaboración propia.

La prueba verifica el funcionamiento del formulario de inicio de session.



Ilustración 7: Salida del registro de servidor de Django con una petición POST al endpoint /api/login. El registro indica que la solicitud se procesó correctamente y que el servidor respondión con un código de estado 200. Fuente: Elaboración propia