# Informe de Pruebas de Carga y Estrés de API

**Objetivo de la Prueba**: Evaluar el rendimiento del API bajo diferentes niveles de carga y estrés para identificar cuellos de botella y garantizar la escalabilidad del servicio.

### Resumen de Resultados:

* **Tiempos de Respuesta Promedio**:
  + **Carga (150 usuarios concurrentes)**:
    - Tiempo promedio de respuesta para addProduct: **683.93 ms**
    - Tiempo promedio de respuesta para getAllProducts: **218.27 ms**
  + **Estrés (de 100 a 1000 usuarios)**:
    - Tiempo promedio de respuesta para addProduct: **6497.41 ms**
    - Tiempo promedio de respuesta para getAllProducts: **567.02 ms**
* **Comportamiento del API**:
  + **Carga**:
    - Respuestas exitosas: **99.97%** (29588 solicitudes, 8 errores)
    - Errores: **0.03%** (conexiones reiniciadas y fallos de respuesta)
  + **Estrés**:
    - Respuestas exitosas: **97.59%** (92580 solicitudes, 2396 errores)
    - Errores: **2.59%** (principalmente conexiones reiniciadas y fallos de respuesta)

El reporte de las **pruebas de carga** a un nivel más detallado, con estadistas y graficas para el tiempo de la prueba, rendimiento y tiempos de respuesta lo puede encontrar en está url: <https://github.com/AnderGGomez/performace_test/tree/main/NoFuncional/01-150%20Users/HTML_REPORT>

El reporte de las **pruebas de estrés** a un nivel más detallado, con estadistas y graficas para el tiempo de la prueba, rendimiento y tiempos de respuesta lo puede encontrar en está url: <https://github.com/AnderGGomez/performace_test/tree/main/NoFuncional/02-100%20to%201000%20Users/HTML_REPORT>

### Análisis de Resultados:

El API mostró un rendimiento adecuado bajo carga moderada, pero los tiempos de respuesta aumentaron drásticamente bajo condiciones de estrés. En particular, al escalar a 1000 usuarios, los tiempos de respuesta para addProduct se elevaron significativamente, alcanzando hasta **6497.41 ms**, lo que sugiere que el sistema no está optimizado para manejar altos volúmenes de solicitudes concurrentes. Además, se observó un incremento en el porcentaje de errores, lo que indica posibles problemas en la infraestructura o en la gestión de conexiones.

### Recomendaciones:

1. **Ajustar la configuración del servidor**: Aumentar los recursos del servidor y revisar la configuración para manejar conexiones concurrentes.
2. **Escalabilidad horizontal**: Considerar la implementación de un sistema que permita escalar horizontalmente, añadiendo más instancias del servicio.
3. **Revisar la gestión de conexiones**: Mejorar el manejo de conexiones para evitar excepciones y desconexiones inesperadas.

### Conclusiones:

El API cumple con los requisitos de rendimiento bajo condiciones de carga normal, pero presenta desafíos significativos para manejar situaciones de alta demanda. Implementar las recomendaciones propuestas ayudará a mejorar la capacidad de respuesta, la estabilidad del servicio y la experiencia del usuario, asegurando que el API sea capaz de soportar un mayor número de usuarios concurrentes de manera eficiente.