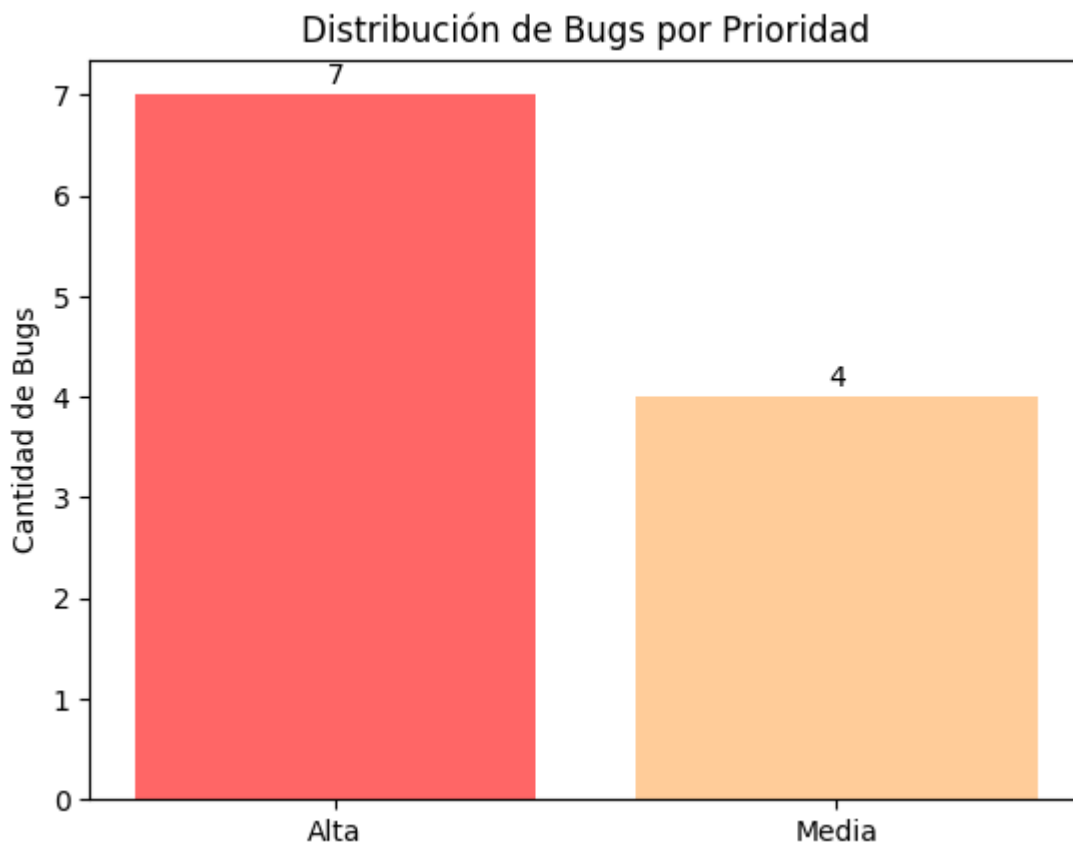


## 1. Distribución de Bugs por Prioridad

- **Prioridad:** Alta, Media.
- **Cantidad de Bugs:**
  - **Alta:** 7 bugs
  - **Media:** 4 bugs

### Análisis:

- La prioridad indica qué tan urgente o importante es resolver un bug.
- En este caso, la mayoría de los bugs (7) tienen alta prioridad, lo que sugiere que hay varios errores que afectan de manera crítica al sistema o a los usuarios. Esto implica que el equipo debería centrar sus esfuerzos en solucionar estos problemas primero.
- Solo 4 bugs tienen prioridad media, lo que indica que aunque son importantes, no requieren la misma atención inmediata que los de alta prioridad.
- **Conclusión:** El enfoque principal debe ser en la resolución de los bugs de alta prioridad para evitar un impacto negativo en los usuarios o el rendimiento del sistema.

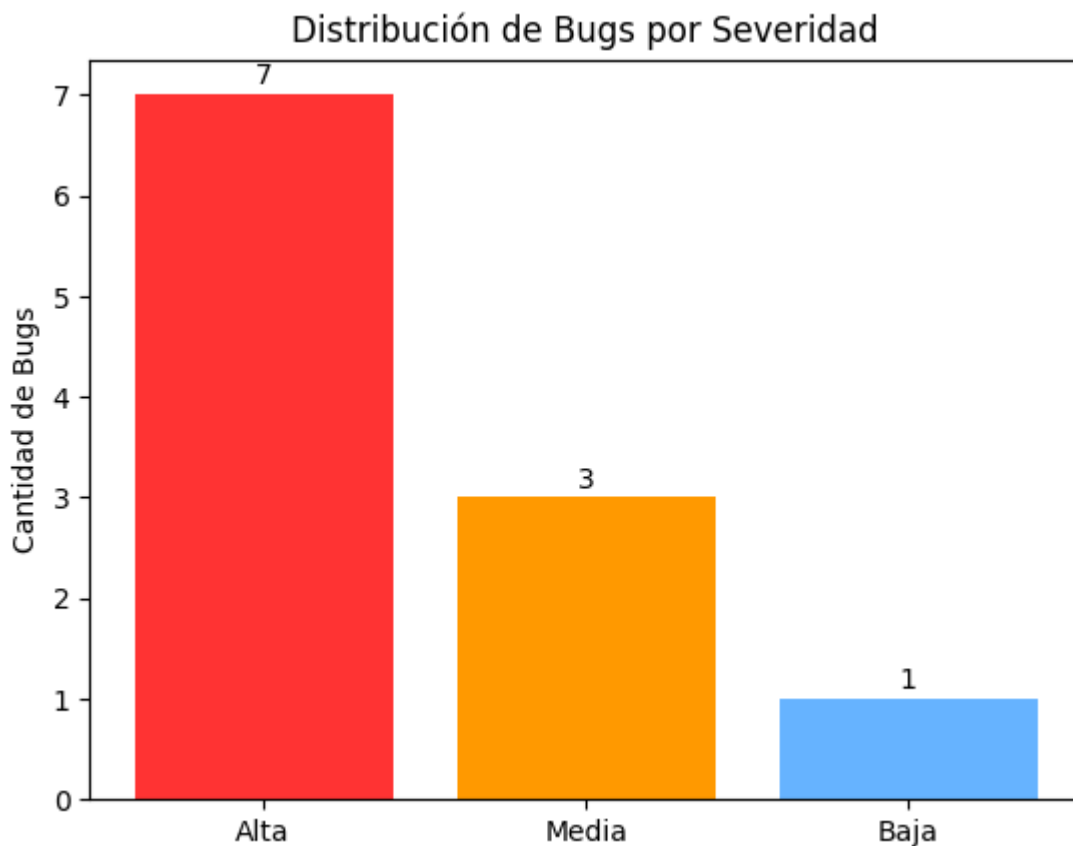


## 2. Distribución de Bugs por Severidad

- **Severidad:** Alta, Media, Baja.
- **Cantidad de Bugs:**
  - **Alta:** 7 bugs
  - **Media:** 3 bugs
  - **Baja:** 1 bug

### Análisis:

- La severidad de los bugs indica el grado de impacto o daño que causan al sistema.
- En este caso, la severidad más común es **alta**, con 7 bugs. Esto es consistente con la prioridad, ya que los bugs de alta severidad suelen tener alta prioridad.
- La severidad **media** y **baja** es menor, lo que sugiere que el impacto de estos bugs no es tan crítico o urgente.
- **Conclusión:** De los bugs identificados, la mayoría tienen un impacto alto en el sistema, por lo que se deben priorizar su solución. Los bugs de severidad baja pueden ser tratados con menor urgencia.

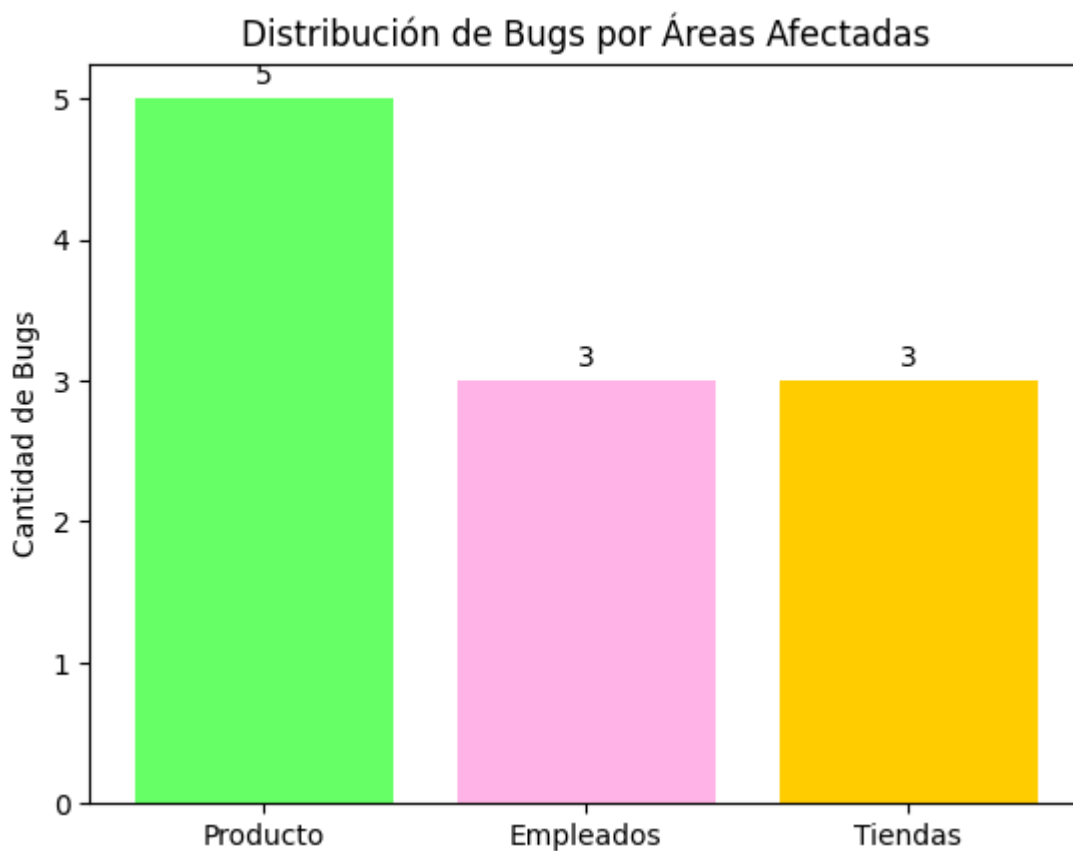


### 3. Distribución de Bugs por Áreas Afectadas

- **Áreas Afectadas:** Producto, Empleados, Tiendas.
- **Cantidad de Bugs:**
  - **Producto:** 5 bugs
  - **Empleados:** 3 bugs
  - **Tiendas:** 3 bugs

#### Análisis:

- Las áreas afectadas proporcionan información sobre en qué partes del sistema o negocio se encuentran los bugs.
- La mayor cantidad de bugs afecta al **Producto**, con 5 casos. Esto podría implicar problemas con la funcionalidad principal del producto que está en uso por los usuarios finales.
- Las áreas de **Empleados** y **Tiendas** tienen la misma cantidad de bugs (3 cada una), lo que sugiere que también hay errores que afectan procesos internos o la gestión de tiendas, pero en menor número.
- **Conclusión:** El equipo debe centrarse en mejorar la calidad del producto, ya que es donde se concentran la mayoría de los errores. Sin embargo, no se deben descuidar los errores que afectan a los empleados o las tiendas, especialmente si estos impactan operaciones críticas.

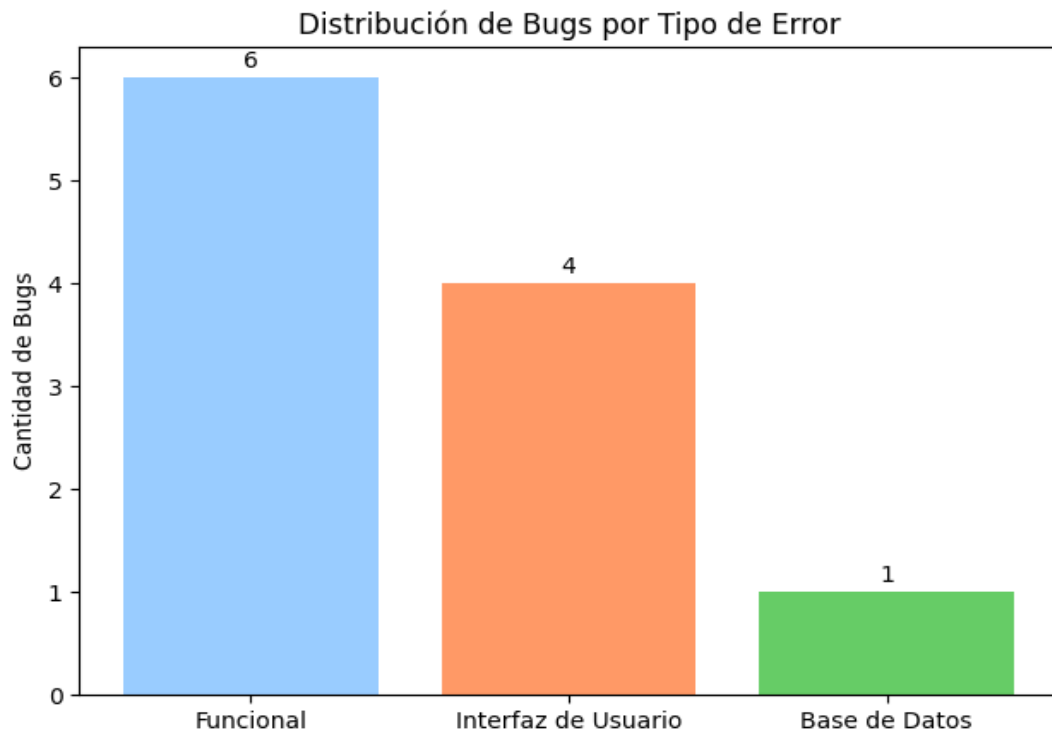


#### 4. Distribución de Bugs por Tipo de Error

- **Tipo de Bug:** Funcional, Interfaz de Usuario, Base de Datos.
- **Cantidad de Bugs:**
  - **Funcional:** 6 bugs
  - **Interfaz de Usuario:** 4 bugs
  - **Base de Datos:** 1 bug

##### Análisis:

- Los tipos de errores se refieren a las áreas técnicas específicas donde se encuentran los problemas.
- El mayor número de bugs son de tipo **Funcional** (6), lo que sugiere que la lógica del sistema, las funcionalidades y las interacciones principales están experimentando problemas. Esto podría afectar gravemente la experiencia del usuario o el rendimiento del sistema.
- Hay 4 bugs relacionados con la **Interfaz de Usuario**, lo que puede significar que los usuarios tienen dificultades para interactuar con el sistema debido a errores visuales o de diseño.
- Solo 1 bug se refiere a la **Base de Datos**, lo que podría ser un problema técnico menos frecuente pero aún relevante si afecta el rendimiento o la integridad de los datos.
- **Conclusión:** Los bugs funcionales deben ser la principal prioridad para resolver, ya que afectan la operatividad del sistema. Los bugs en la interfaz de usuario también deben ser atendidos, pero los problemas con la base de datos parecen ser menos frecuentes y pueden ser tratados después.



## Resumen Final:

- **Prioridad y Severidad:** La mayoría de los bugs tienen alta prioridad y severidad, lo que sugiere que hay errores graves que deben ser atendidos con urgencia para evitar impactos mayores.
- **Áreas Afectadas:** El producto es la área más afectada, por lo que se debe concentrar el esfuerzo de corrección allí.
- **Tipo de Error:** Los bugs funcionales son los más comunes, seguidos de los problemas de interfaz de usuario, lo que indica que el sistema podría necesitar mejoras en su lógica y diseño para mejorar la experiencia del usuario.

## Calidad de código

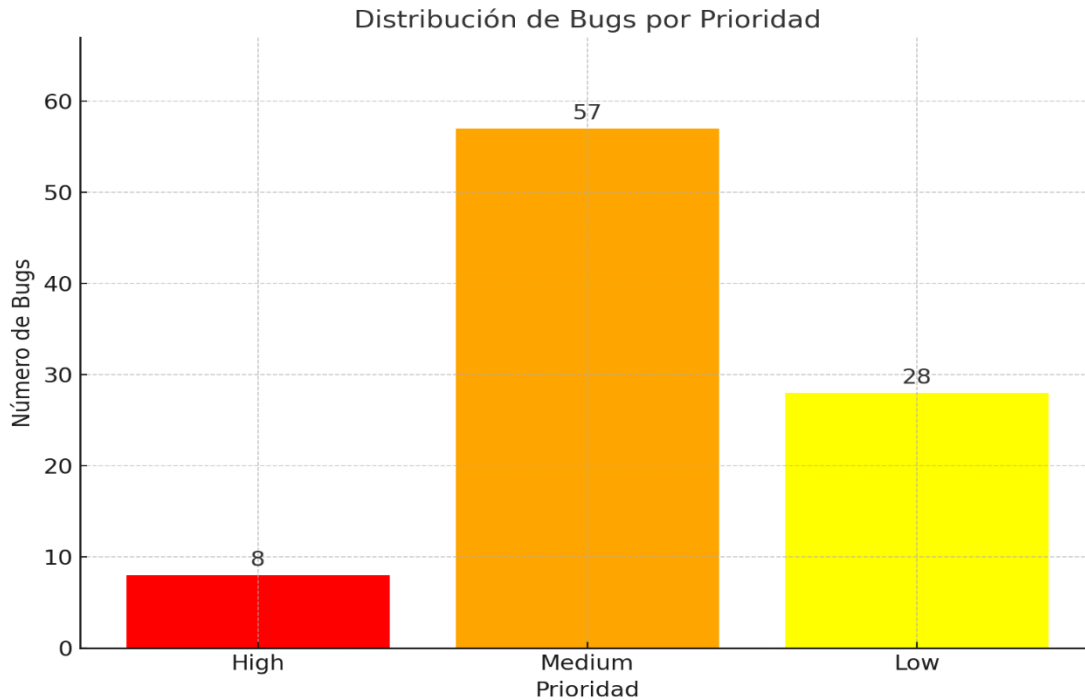
### Cantidad de Bugs por Prioridad:

- **Alta (High):** 8 bugs
- **Media (Medium):** 57 bugs
- **Baja (Low):** 28 bugs

### Análisis:

La distribución de bugs por prioridad muestra que la mayoría de los problemas se encuentran en la categoría de severidad media, con 57 bugs. Esto indica que, aunque su impacto no es crítico, representan un volumen significativo que debe ser atendido para mantener la estabilidad del sistema.

Por otro lado, los bugs de severidad alta son menos frecuentes (8 en total), pero su impacto es crítico y deben resolverse con la mayor urgencia para evitar fallos graves en el sistema. Los bugs de severidad baja, aunque presentes en menor cantidad (28 bugs), tienen un impacto menor y pueden ser tratados de manera más flexible.



### **Conclusión:**

La priorización de esfuerzos debe enfocarse inicialmente en resolver los 8 bugs de severidad alta, debido a su impacto potencialmente crítico. Posteriormente, se debe abordar el volumen significativo de bugs de severidad media, ya que su acumulación puede afectar la experiencia del usuario y la funcionalidad general del sistema. Los bugs de severidad baja pueden atenderse en etapas posteriores, según los recursos disponibles y las necesidades del proyecto.

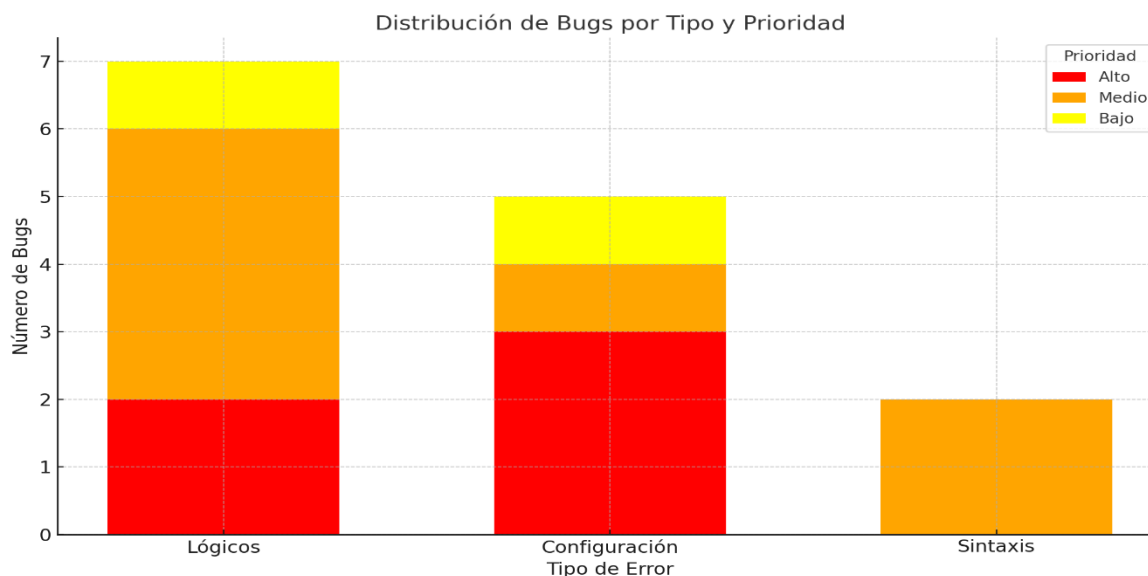
### **Cantidad de Bugs por Tipo y Prioridad:**

- **Errores Lógicos:**
  - Alta: 2 bugs
  - Media: 4 bugs
  - Baja: 1 bug
- **Errores de Configuración:**
  - Alta: 3 bugs
  - Media: 1 bug
  - Baja: 1 bug
- **Errores de Sintaxis:**
  - Alta: 0 bugs
  - Media: 2 bugs
  - Baja: 0 bugs

### **Análisis:**

La distribución por tipo y prioridad muestra que los errores de **configuración y lógicos** tienen mayor severidad alta. Esto sugiere que son los principales causantes de los problemas más críticos en el sistema. Por otro lado, los errores de **sintaxis** tienen menor impacto, ya que no se encuentran casos de severidad alta y tienen solo dos de severidad media.

El tipo de error con mayor cantidad de bugs totales es el lógico, con 7 en total, y su distribución en distintas prioridades refleja su alta frecuencia e importancia.



### Conclusión:

Los errores lógicos y de configuración de severidad alta deben ser priorizados para su resolución inmediata debido a su impacto crítico en el sistema. Los errores de severidad baja, como los de configuración o lógicos de menor prioridad, pueden tratarse en un segundo momento. Esto permitirá optimizar el desempeño y estabilidad del sistema, enfocándose en los problemas más urgentes.

### Cantidad de Bugs por Tipo de error y severidad:

- **Errores Lógicos:**
  - Alta: 2 bugs
  - Media: 4 bugs
  - Baja: 1 bug
- **Errores de Configuración:**
  - Alta: 3 bugs
  - Media: 1 bug
  - Baja: 1 bug
- **Errores de Sintaxis:**

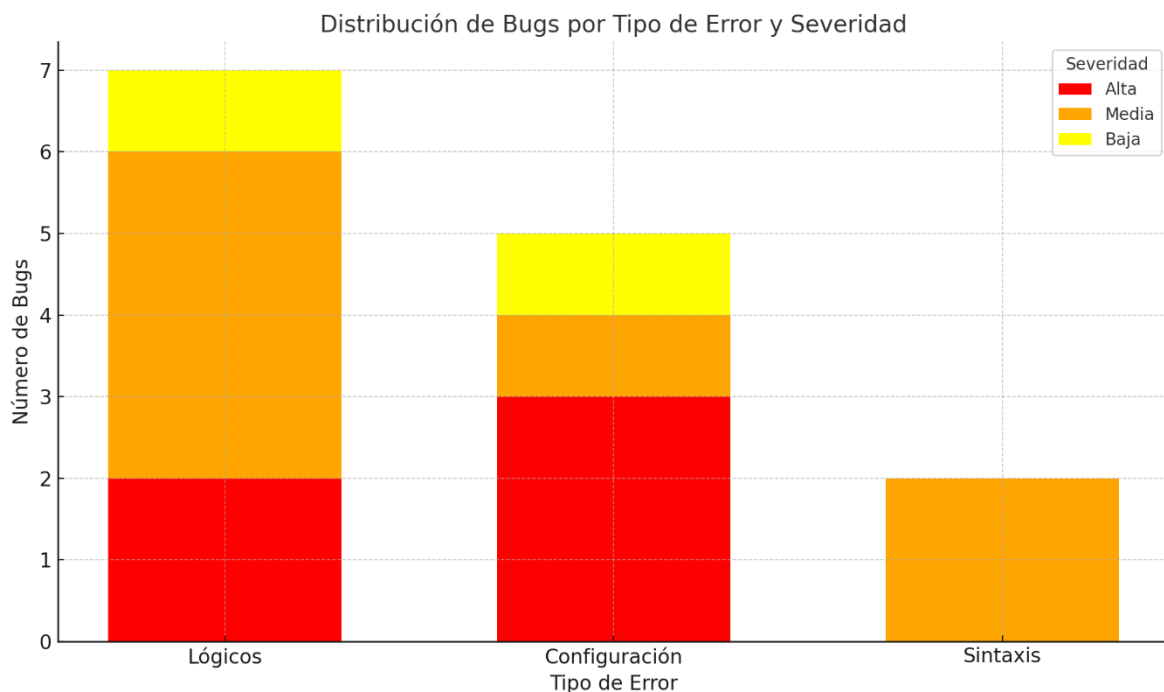
Alta: 0 bugs  
Media: 2 bugs  
Baja: 0 bugs

### Análisis General:

- **Bugs por Severidad:** Predominan los bugs de severidad alta y media. Los de alta severidad (5) se concentran en Configuración (3) y Lógicos (2), mientras que los de severidad media (7) son más frecuentes, especialmente en Lógicos (4). Los de severidad baja son pocos (2) y tienen poco impacto.

### Errores Principales:

- **Errores Lógicos:** Son los más comunes, con variada severidad (alta, media, baja), lo que sugiere problemas en el diseño o lógica del sistema.
- **Errores de Configuración:** Tienen mayor severidad alta, indicando configuraciones incorrectas.
- **Errores de Sintaxis:** Menos frecuentes, pero de severidad media, impactando funciones específicas.



### Conclusión:

1. **Priorización inmediata:** Resolver los bugs de alta severidad en Configuración y Lógicos.
2. **Planificación intermedia:** Abordar los bugs de severidad media, especialmente lógicos, para evitar impacto en el rendimiento.



3. **Mantenimiento posterior:** Los bugs de baja severidad pueden gestionarse más tarde, sin afectar objetivos inmediatos.

▼ Gravedad ?	
High	8
Medium	57
Low	28
▼ Tipo	
Bug	9
Vulnerability	0
Code Smell	79
▼ Alcance	
Main code	88
Test code	0

**Gravedad:**

**Alta (High):** 8 problemas críticos que requieren atención inmediata.

**Media (Medium):** 57 problemas de prioridad moderada.

**Baja (Low):** 28 problemas menores.

**Tipo de problema:**

**Bugs:** 9 errores en el código que podrían impactar en su funcionamiento.

**Vulnerabilidades:** Ningún problema relacionado con la seguridad (0).

**Code Smell:** 79 indicios de mal diseño o prácticas de codificación que no siguen las mejores prácticas.

**Alcance:**

Todos los problemas se encuentran en el código principal (Main code) (88 en total).

No se identificaron problemas en el código de pruebas (Test code).

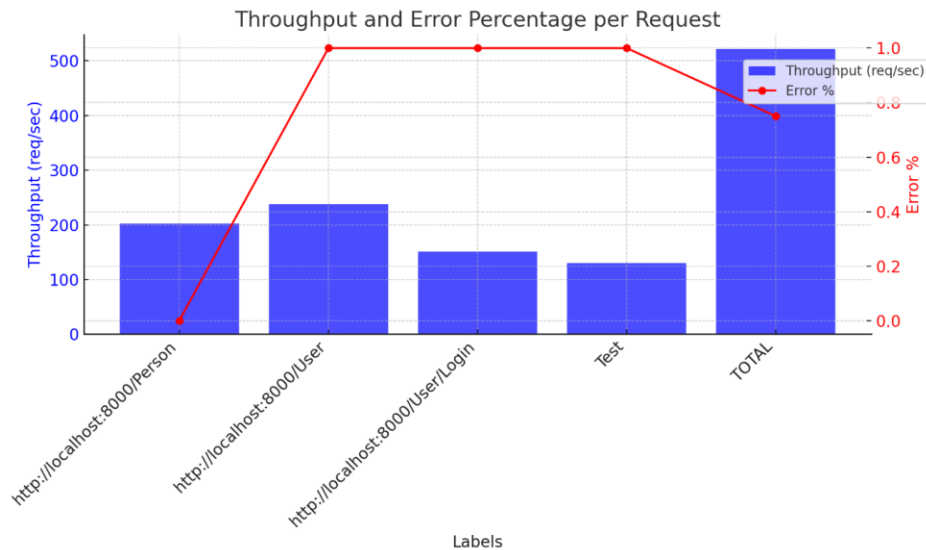
## CONCLUSION FINAL.

Se considera que la calidad del código en este proyecto tiene una puntuación de 2, ya que presenta fallos críticos, baja cobertura de pruebas y duplicidad de código, lo que demuestra la necesidad de mejorar significativamente su estructura y mantenimiento.

## PRUEBAS NO FUNCIONALES

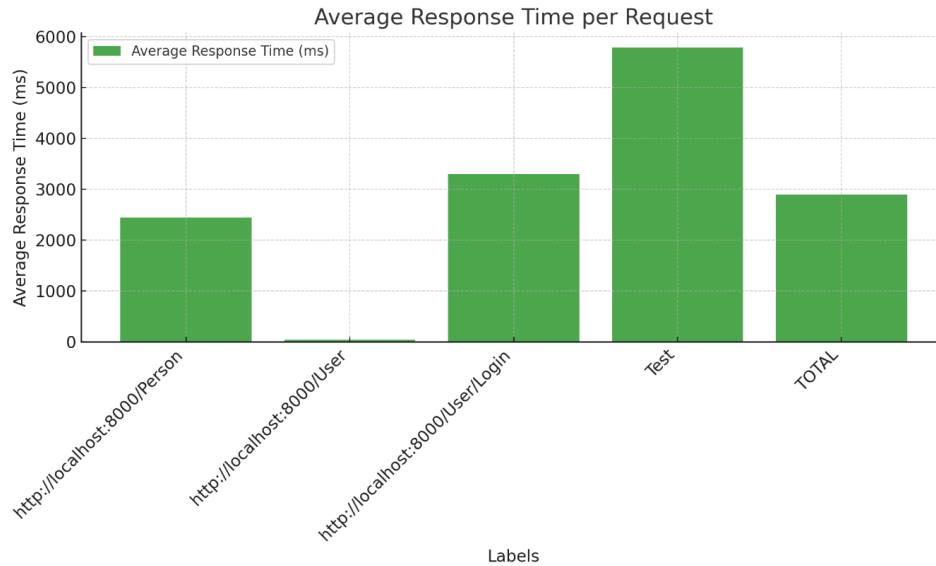
### USUARIO-LOGIN 1000 USUARIOS

#### Throughput and Error Percentage per Request



#### Throughput y Porcentaje de Errores:

- Las barras azules representan el throughput (solicitudes por segundo) para cada endpoint.
- La línea roja muestra el porcentaje de errores, lo que ayuda a identificar puntos críticos con problemas de rendimiento o fallos.

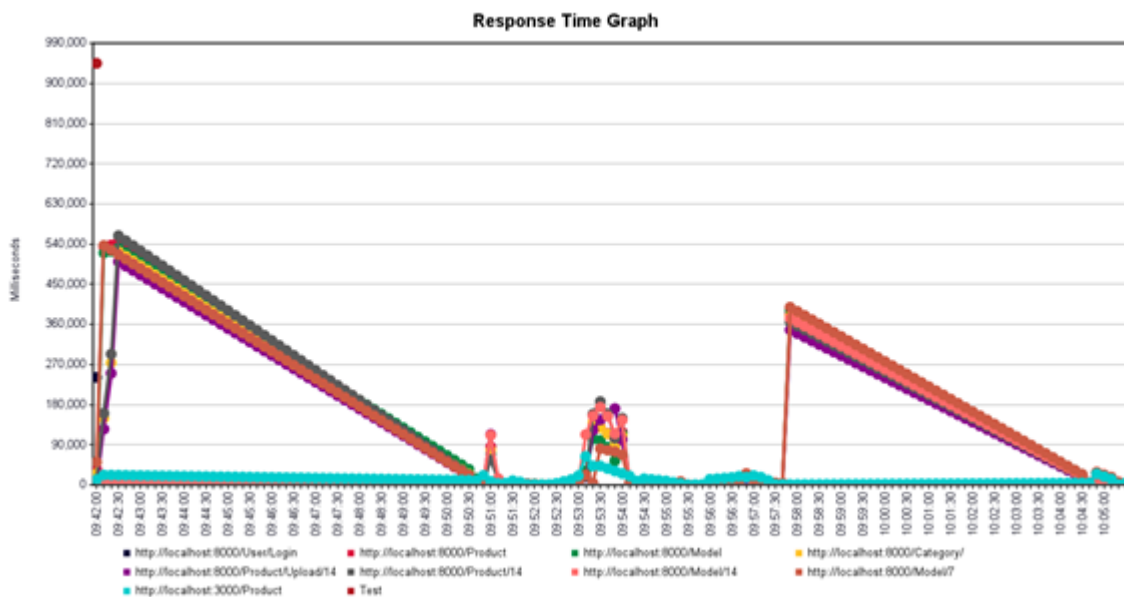


### Tiempo Promedio de Respuesta:

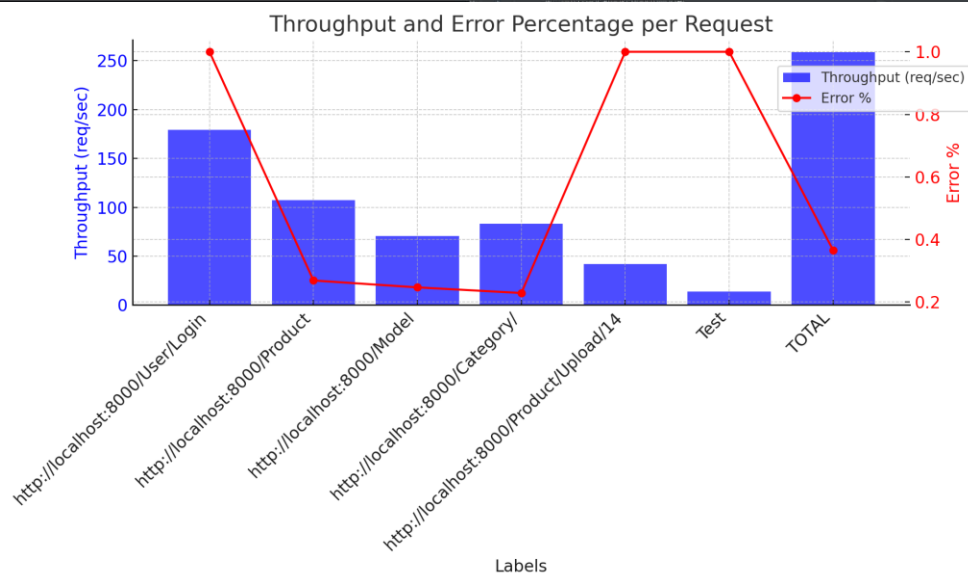
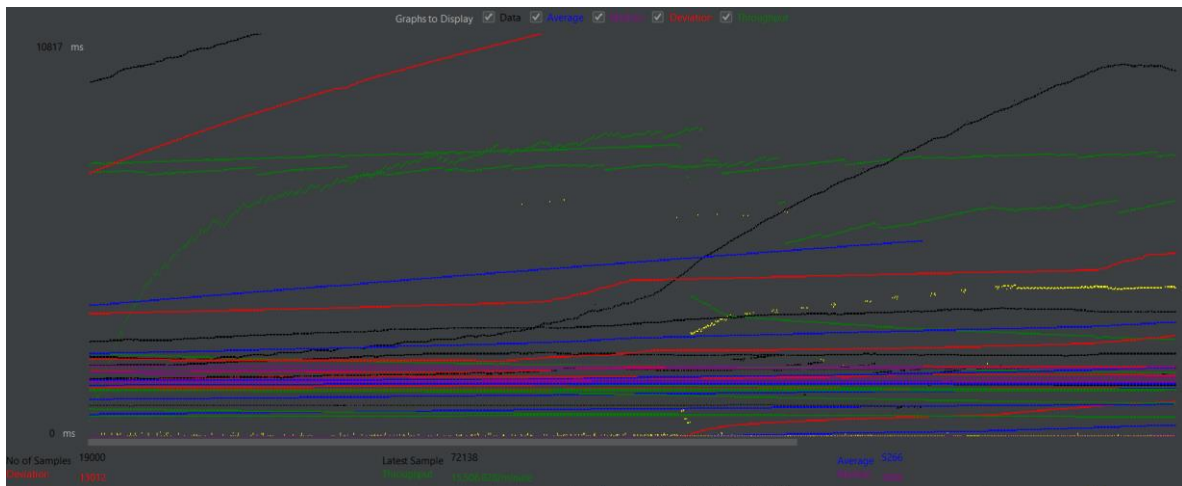
Las barras verdes representan el tiempo promedio de respuesta (en milisegundos) por solicitud, destacando cuáles endpoints tienen tiempos más altos.

Estas visualizaciones proporcionan una visión clara de las métricas de rendimiento clave y los posibles cuellos de botella en el sistema.

### CRUD LOGIN, Crear Producto (1000 usuarios)

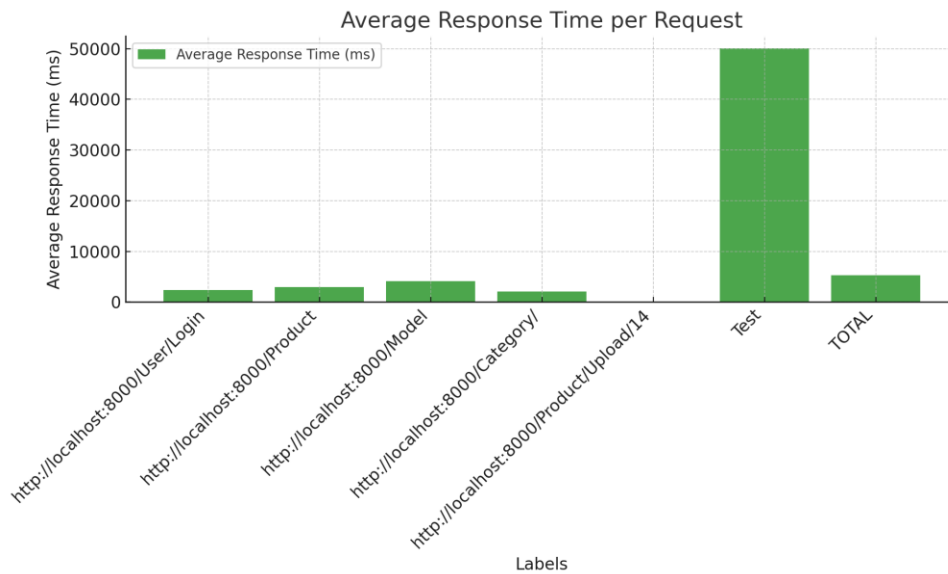


Grafica de Resultados



### Throughput y Porcentaje de Errores:

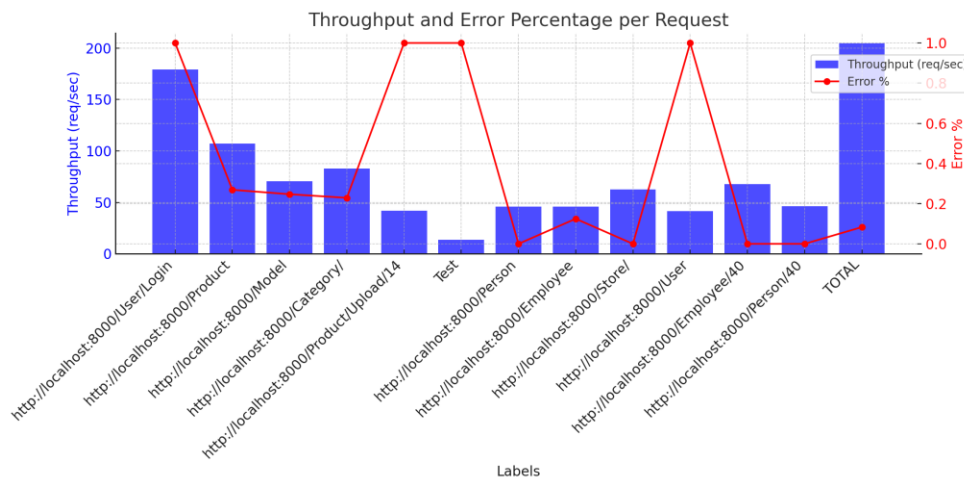
- **Throughput:**
  - La solicitud <http://localhost:8000/User/Login> tiene el mayor throughput con 179.31 solicitudes por segundo, lo cual es positivo.
  - Las solicitudes como <http://localhost:8000/Product/Upload/14> y Test tienen un throughput significativamente más bajo, indicando que estas solicitudes están sobrecargando el sistema.
- **Porcentaje de Errores:**
- Las solicitudes con 1.0% de errores (por ejemplo, <http://localhost:8000/User/Login>, Test, y <http://localhost:8000/Product/Upload/14>) requieren atención inmediata. Los errores pueden deberse a problemas de validación, manejo de carga o tiempos de espera agotados.



### Tiempo Promedio de Respuesta:

- El tiempo promedio más alto es para la solicitud Test con 50029 ms, lo cual indica un problema crítico en la aplicación que necesita optimización.
- Las solicitudes como <http://localhost:8000/Category/> y <http://localhost:8000/Product> tienen tiempos promedios aceptables en comparación, pero aún podrían mejorarse.

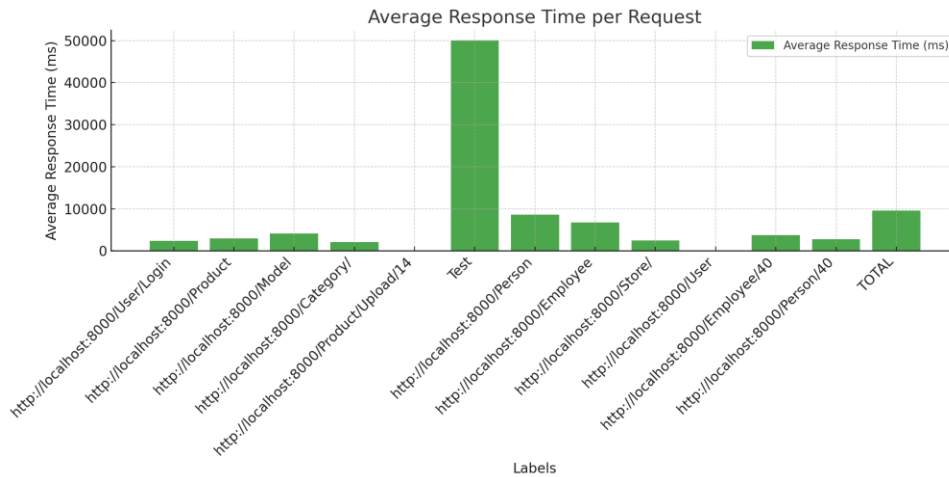
### EMPLEADOS 1000 Usuarios



### Throughput y Porcentaje de Errores:

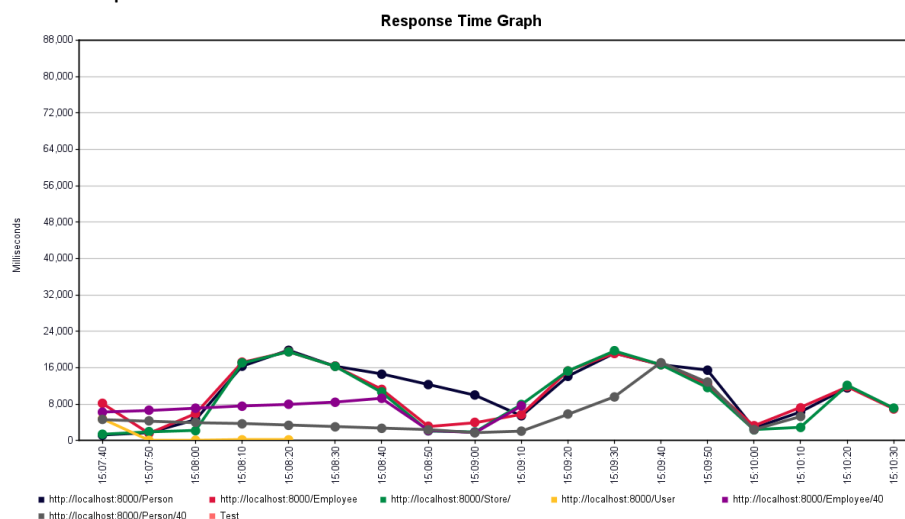
- Throughput:**
  - Las solicitudes como <http://localhost:8000/Store/> y <http://localhost:8000/Employee/40> tienen un throughput notablemente alto, lo cual es positivo.

- Las solicitudes críticas como Test tienen un throughput muy bajo, lo que indica que estas solicitudes podrían estar sobrecargando el sistema.
- **Porcentaje de Errores:**
  - Las solicitudes con un 1.0% de errores (por ejemplo, <http://localhost:8000/User/Login> y Test) representan problemas críticos. Estos errores pueden deberse a validaciones fallidas o tiempos de espera agotados.

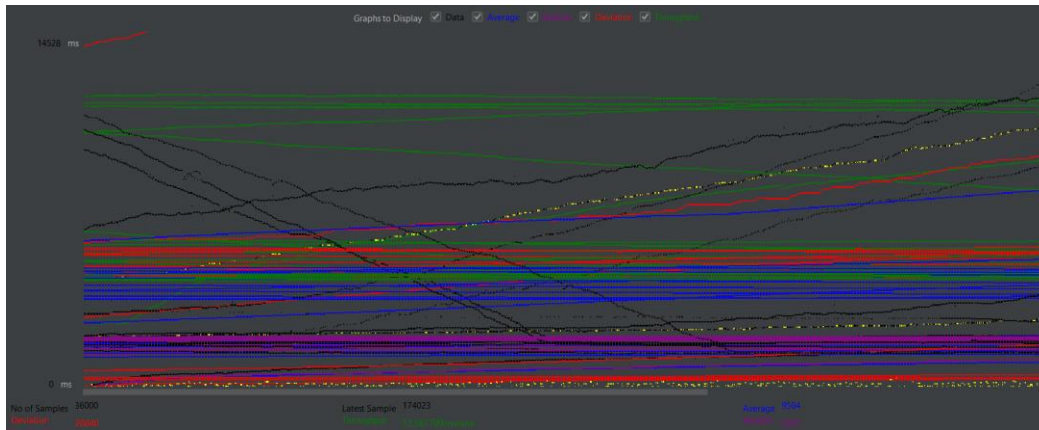


### Tiempo Promedio de Respuesta:

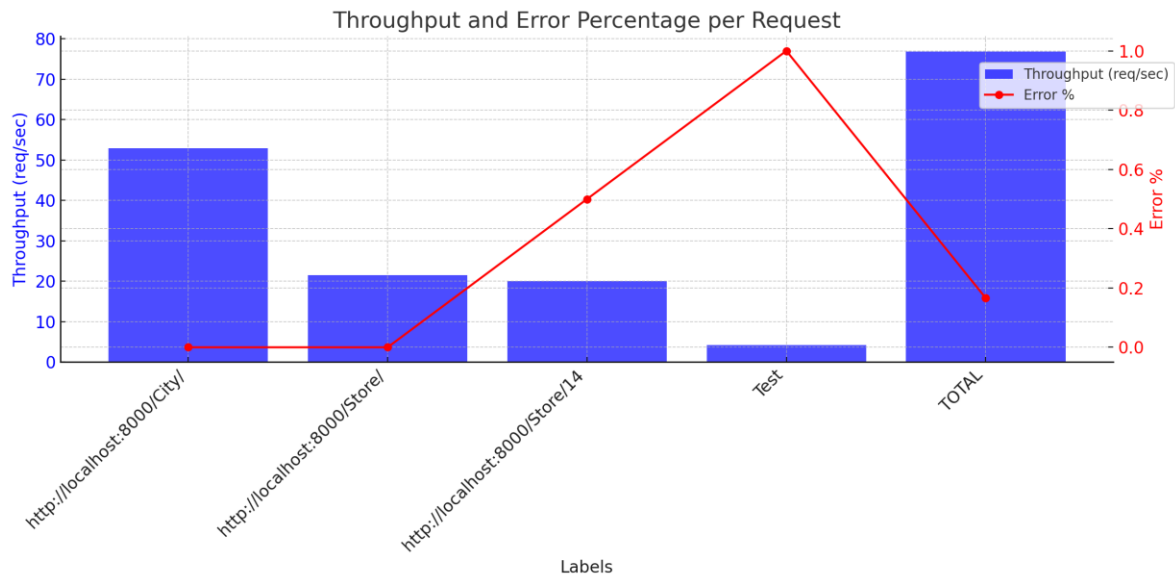
- Las solicitudes como Test tienen un tiempo promedio extremadamente alto (50,029 ms), lo que indica un problema grave en el rendimiento.
- <http://localhost:8000/Person> también muestra tiempos elevados, lo que puede afectar la experiencia del usuario.



GRAFICA DE RESULTADOS



## CRUD TIENDA-1000 USUARIO

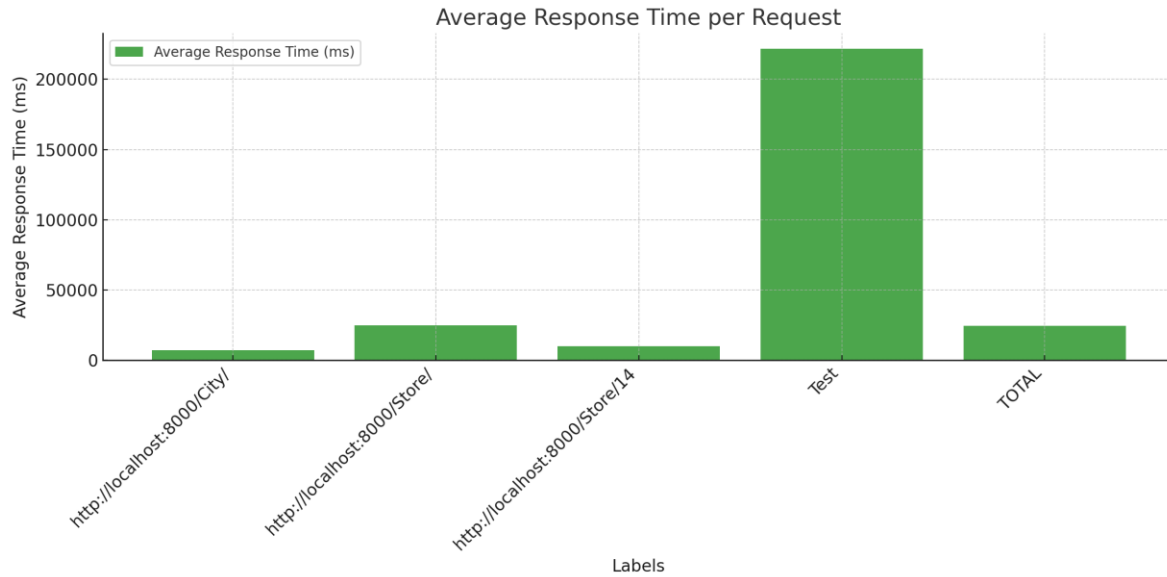


### Throughput y Porcentaje de Errores:

- **Throughput:**
  - <http://localhost:8000/City/> tiene el throughput más alto con 52.88 solicitudes por segundo, lo que indica una buena capacidad de manejo de solicitudes para este endpoint.
  - Test tiene el throughput más bajo (4.26 solicitudes por segundo), lo que indica que esta solicitud es una carga pesada y podría estar limitando el rendimiento del sistema.

### Porcentaje de Errores:

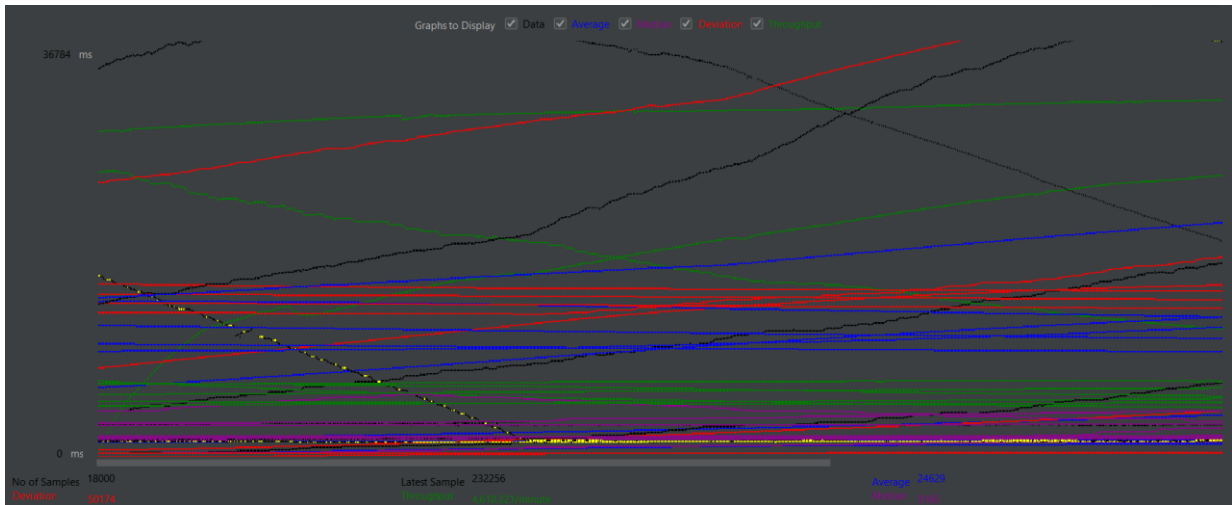
- <http://localhost:8000/Store/14> tiene un 0.5% de errores y Test tiene un 1.0%, lo que sugiere problemas de fiabilidad en estas operaciones.



**Tiempo Promedio de Respuesta:**

- Las solicitudes como Test tienen un tiempo promedio extremadamente alto (221,662 ms), lo que indica un problema crítico de rendimiento.
- <http://localhost:8000/Store/> también muestra tiempos promedio elevados (24,843 ms), lo cual necesita ser optimizado para mejorar la experiencia del usuario.





Response Time Graph

