

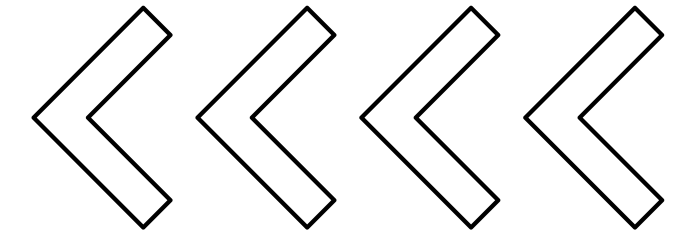
PRESENTACIÓN DE PROYECTO

# CREACIÓN DE UNA API PARA ANALIZAR EL RENDIMIENTO DE APLICACIONES WEB

Alejandro Becerra Acevedo  
Johan Sebastian Henao Cañas



# AGENDA



**1**

Objetivos

**2**

Marco teórico

**3**

Implementación

**4**

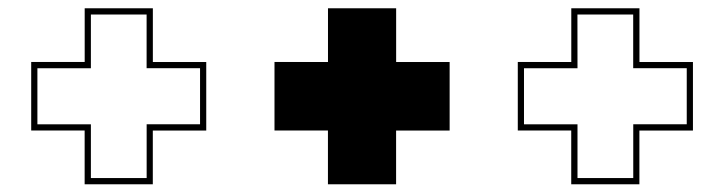
Experimentos y Resultados

**5**

Conclusiones

**6**

Referencias



# OBJETIVO

El objetivo principal de este proyecto fue implementar un sistema de monitorización y perfilado para analizar el desempeño de aplicaciones en diferentes entornos de hardware.

También se buscó:

- Evaluar métricas como el uso de CPU y memoria en tiempo real.
- Realizar análisis detallados del desempeño de funciones internas a través de perfilado.
- Comparar el impacto de diferentes escenarios de carga en dos configuraciones de hardware."

# MARCO TEORICO

**1**

**FastAPI**

**2**

**CPU y memoria**

**3**

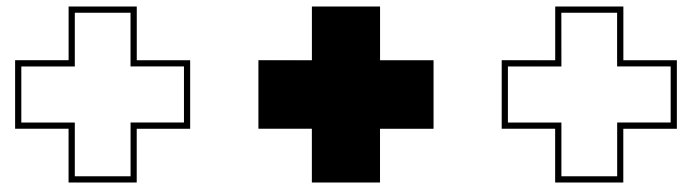
**psutil**

**4**

**cProfile**

**5**

**PostgreSQL**



# IMPLEMENTACIÓN

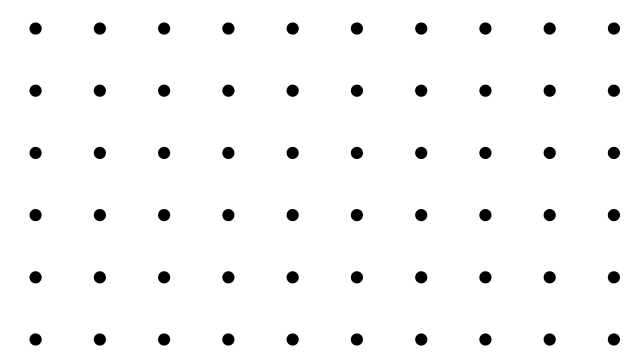
La evaluación se llevó a cabo en dos entornos de hardware:

- Equipo A: Procesador Ryzen 7 8840HS, 16 GB de RAM.
- Equipo B: Procesador Intel Core i7 de 11<sup>a</sup> generación, 16 GB de RAM.

Las pruebas se diseñaron para reproducir escenarios típicos de carga, incluyendo:

- Reproducción de contenido multimedia: Navegación en YouTube usando Chrome.
- Tareas ofimáticas: Edición de documentos en Microsoft Word.
- Navegación web: Uso de múltiples pestañas en sitios de carga similar.

# EXPERIMENTOS Y RESULTADOS

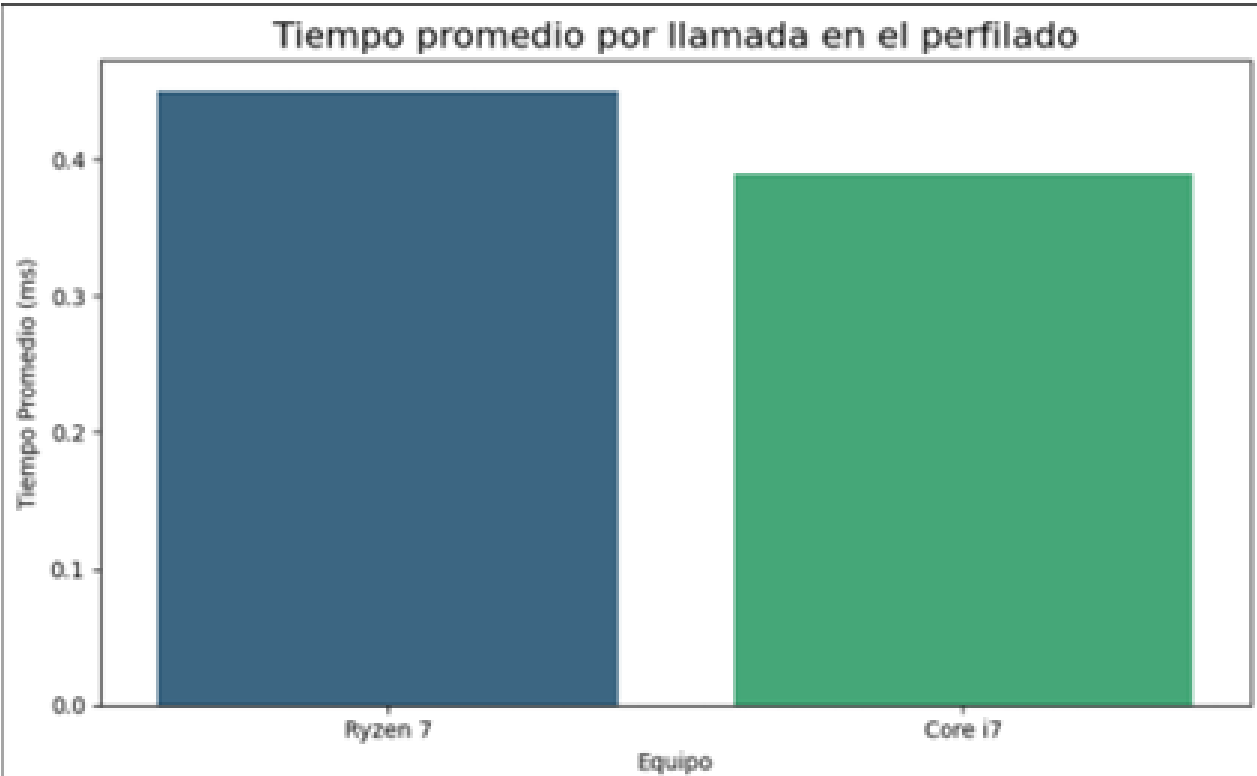
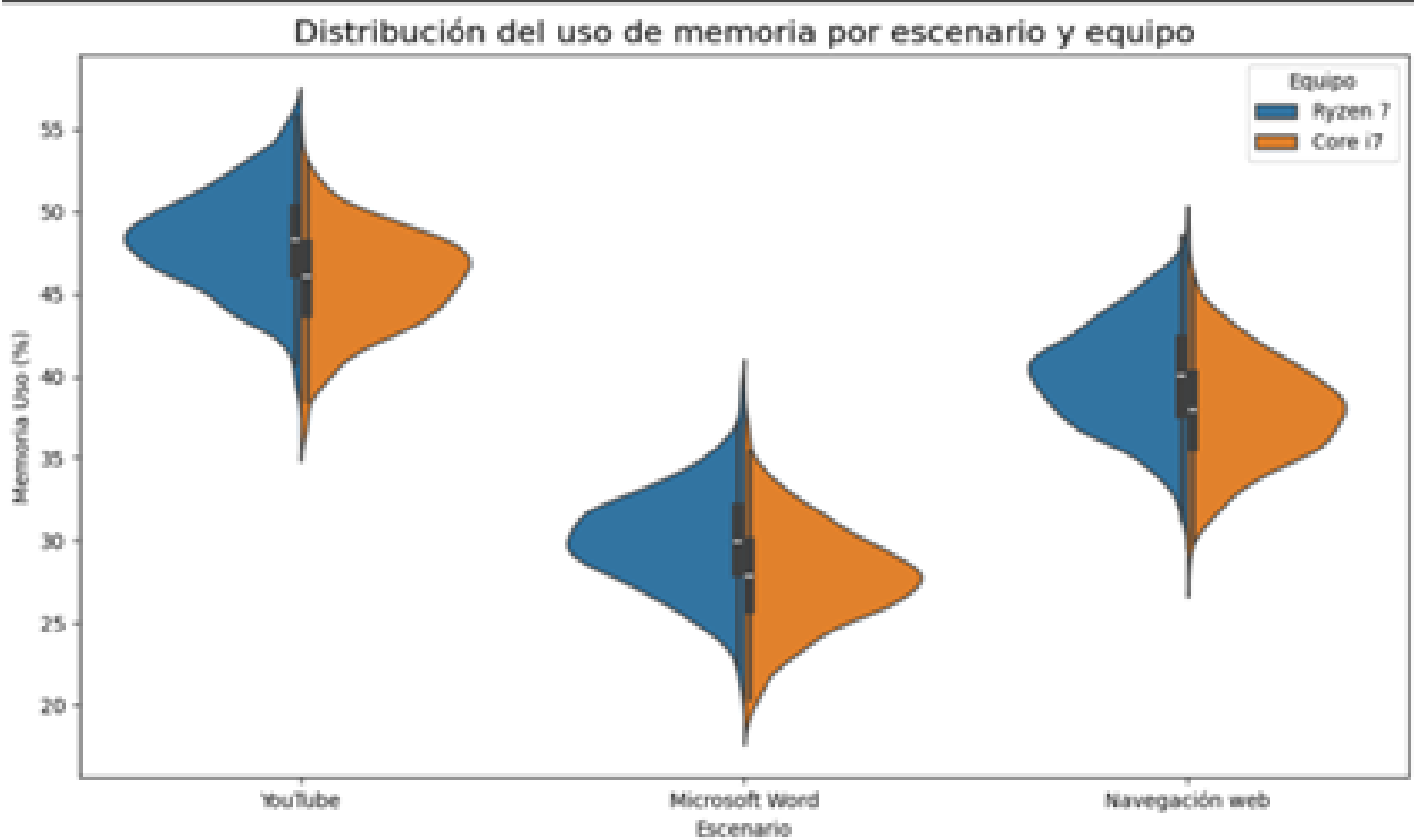
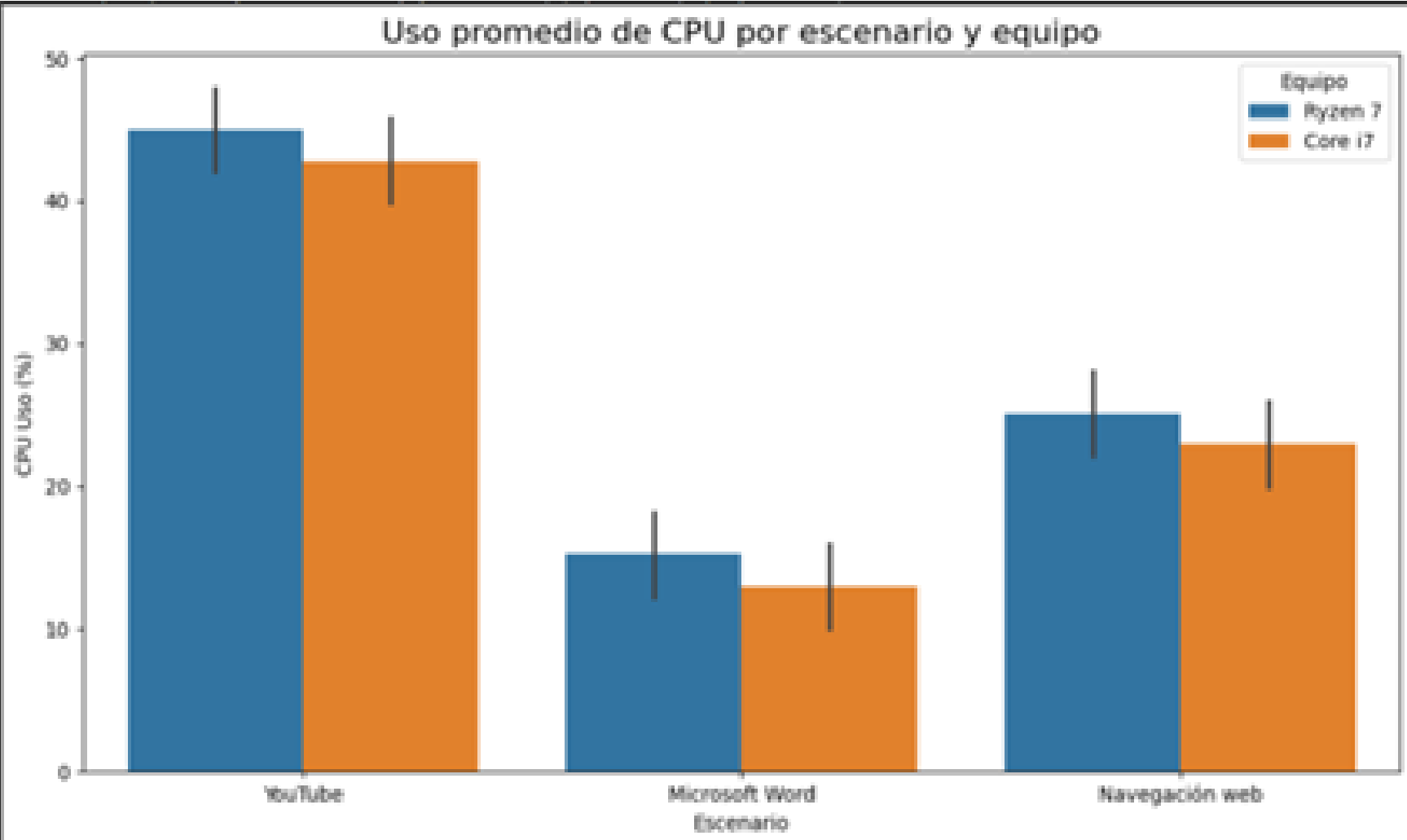


Cada tarea se ejecutó durante periodos de 10 minutos por equipo, asegurando consistencia entre los experimentos y replicabilidad en el análisis de datos.

La validación de los datos recopilados se realizó mediante pruebas piloto en ambos equipos, ejecutando tareas controladas para verificar la consistencia de las métricas reportadas. Se evaluaron diferentes periodos de monitoreo y escenarios de carga para identificar posibles anomalías en los registros.

El análisis estadístico posterior incluyó cálculos de la media y la desviación estándar, asegurando que las métricas presentaran comportamientos esperados bajo condiciones similares. Además, se verificó que las estadísticas de perfilado reflejaran la eficiencia esperada en función de la capacidad del hardware utilizado.

# EXPERIMENTOS Y RESULTADOS



# CONCLUSIÓN

- El sistema desarrollado demostró ser eficiente, estable y confiable para recolectar métricas en tiempo real.
- Las herramientas utilizadas, como psutil y cProfile, ofrecieron datos precisos para el análisis y validación.
- Las diferencias observadas entre los equipos resaltan la importancia de considerar el hardware en el diseño de aplicaciones.
- El proyecto sienta una base sólida para futuras optimizaciones y ampliaciones del sistema



# REFERENCIAS

- [1] FastAPI, "Official Documentation for FastAPI," [Online]. Available: <https://fastapi.tiangolo.com/>.
  - [2] psutil, "psutil Documentation," [Online]. Available: <https://psutil.readthedocs.io/en/latest/>.
  - [3] Python, "Profile Module Documentation," [Online]. Available: <https://docs.python.org/3/library/profile.html>.
  - [4] Swagger, "API Documentation Tools," [Online]. Available: <https://swagger.io/>.
  - [5] seaborn, "Statistical Data Visualization with Seaborn," [Online]. Available: <https://seaborn.pydata.org/>.
  - [6] Matplotlib, "Comprehensive 2D Plotting in Python," [Online]. Available: <https://matplotlib.org/>.
- [7] National Library of Medicine, "Understanding Standard Deviation," [Online]. Available: [https://www.nlm.nih.gov/oet/ed/stats/02-900.html#:~:text=A%20standard%20deviation%20\(or%20%CF%83,data%20are%20more%20spread%20out.](https://www.nlm.nih.gov/oet/ed/stats/02-900.html#:~:text=A%20standard%20deviation%20(or%20%CF%83,data%20are%20more%20spread%20out.)