

Caso 2 – Homework 1

Juan David Vargas Laverde
202210212

Diego Juan Vicente Benavides
202310310

Santiago Rodriguez Mora
202110332

April 3, 2025

Sección 1

Contrato de trabajo del grupo:

Contrato de equipo

Expectativas del equipo:

Calidad de trabajo:

1. **Estándares del proyecto:** Se espera que las presentaciones, la escritura colaborativa, la investigación individual, la preparación de borradores y las revisiones por pares alcancen un nivel profesional, con revisiones constantes y retroalimentación constructiva.
2. **Estrategias para cumplir con estos estándares:** Se llevarán a cabo reuniones periódicas para revisar avances, elaboración conjunta de borradores y sesiones de retroalimentación para garantizar la calidad del trabajo.

Participación del equipo:

1. **Estrategias para asegurar la cooperación y distribución equitativa de tareas:** Se asignarán roles y tareas de manera rotativa y se utilizará una herramienta de seguimiento (como Trello o una hoja compartida) para monitorear el progreso.
2. **Estrategias para fomentar la inclusión de ideas:** Se promoverán sesiones de brainstorming y reuniones en las que todos los miembros puedan expresar sus opiniones y propuestas.
3. **Estrategias para mantenerse en la tarea:** Se establecerán fechas límite intermedias y se enviarán recordatorios periódicos para asegurar el cumplimiento de los objetivos.
4. **Preferencias de liderazgo:** Se optará por un liderazgo compartido y colaborativo, en el que la toma de decisiones se realice de forma consensuada.

Responsabilidad personal:

1. **Asistencia, puntualidad y participación:** Se espera que todos los miembros asistan a las reuniones, sean puntuales y participen activamente.

2. **Cumplimiento de tareas y plazos:** Cada integrante se compromete a realizar y entregar sus tareas en los tiempos establecidos, notificando de antemano cualquier inconveniente.
3. **Comunicación:** Se mantendrá una comunicación fluida a través de un medio en línea (como WhatsApp o Slack) para coordinar el trabajo y resolver dudas.
4. **Compromiso:** Todos los miembros se comprometen a respetar las decisiones grupales y a cumplir con las responsabilidades asignadas.

Consecuencias por no seguir los procedimientos y cumplir con las expectativas:

1. **Manejo de infracciones:** El equipo abordará cualquier incumplimiento en reuniones de revisión y establecerá medidas correctivas de forma colectiva.
2. **Acciones ante infracciones continuas:** Si los incumplimientos persisten, se redistribuirán las tareas y se notificará al coordinador del curso para tomar medidas adicionales.

Acciones para agradecer y fortalecer al equipo:

1. **Reconocimiento de aportes:** Se realizarán reconocimientos públicos durante las reuniones y en el informe final, destacando tanto logros individuales como colectivos.
2. **Celebración de logros:** Al finalizar el proyecto, se organizará una actividad de celebración (por ejemplo, una reunión especial o una salida grupal) para festejar el logro de las metas del equipo.

Compromiso individual:

- a. He participado en la formulación de los estándares, funciones y procedimientos establecidos en este acuerdo.
- b. Entiendo que estoy obligado a cumplir con estos términos y condiciones.
- c. Entiendo que si no acato estos términos y condiciones, sufriré las consecuencias indicadas.

Firmas:

Nombre, fecha: Juan David Vargas Laverde, April 3, 2025

Nombre, fecha: Diego Juan Vicente Benavides, April 3, 2025

Nombre, fecha: Santiago Rodriguez Mora, April 3, 2025

Sección 2

Descripción del algoritmo usado para generar las referencias de página (Opción 1):

El algoritmo para la generación de referencias simula la ejecución del filtro Sobel sobre una imagen BMP de 24 bits. Primero, se lee la imagen y se extraen sus dimensiones y datos de píxeles, generando matrices correspondientes a la imagen original, los filtros (SOBEL_X y SOBEL_Y) y la imagen de respuesta. El algoritmo recorre la imagen en orden row-major y, para cada píxel (exceptuando bordes), calcula la magnitud del gradiente aplicando las máscaras de Sobel. Durante este proceso se generan referencias de memoria que incluyen:

- Identificación de la celda (p.ej., `Imagen[0][0].r`).
- Número de página virtual asignada.
- Desplazamiento en la página.
- Tipo de operación (R para lectura, W para escritura).

Estas referencias se registran en un archivo junto con metadatos (TP, NF, NC, NR, NP) que permiten simular el comportamiento del sistema de paginación.

Sección 3

Descripción de las estructuras de datos y su actualización en la simulación:

El sistema de simulación utiliza las siguientes estructuras:

- **Tabla de Páginas:** Registra el estado de cada página virtual, incluyendo los bits de referencia (R) y modificación (M).
- **RAM:** Simula la memoria principal mediante un arreglo de marcos de página, en donde se cargan las páginas según su necesidad.
- **SWAP:** Emula el área de intercambio en disco, donde se almacenan las páginas modificadas al producirse un fallo de página.

El hilo *Reader* procesa cada referencia, actualizando la tabla de páginas y cargando o reemplazando páginas en la RAM de acuerdo con el algoritmo NRU. Paralelamente, el hilo *Inspector* resetea periódicamente los bits de referencia para permitir que la clasificación de páginas se actualice de manera dinámica, asegurando que las páginas que no han sido utilizadas recientemente sean candidatas al reemplazo.

Sección 4

Esquema de sincronización usado:

Debido a la concurrencia entre el hilo *Reader* y el hilo *Inspector*, es fundamental emplear sincronización para evitar condiciones de carrera en:

- La actualización de la tabla de páginas (modificación de los bits de referencia y modificación).
- La asignación y liberación de marcos en la RAM.
- La ejecución del algoritmo NRU durante la selección y reemplazo de páginas.

Para ello se utilizan mecanismos de *locks* o monitores que aseguran que las operaciones críticas se ejecuten de forma atómica, garantizando la integridad y coherencia de los datos compartidos entre los hilos.

Sección 5

Tabla de datos recopilados:

A continuación se presenta una tabla de ejemplo con los resultados obtenidos en las pruebas, donde se varían el tamaño de página y el número de marcos asignados:

Tamaño de Página (B)	Marcos Asignados	Total Referencias	Hits	Fallas de Página
512	4	756756	743318	13438
512	6	756756	756556	200
1024	4	756756	756518	238
1024	6	756756	756701	55
2048	4	756756	756716	40
2048	6	756756	756728	28

Sección 6

Gráficas del comportamiento del sistema:

Se presentan las siguientes gráficas para ilustrar el desempeño del sistema:

- **Gráfica 1:** Para un tamaño de página fijo, se grafica el número de marcos asignados vs. el número de fallas de página (misses).
- **Gráfica 2:** Para el mismo tamaño de página, se grafica el número de marcos asignados vs. el número de aciertos (hits).
- **Gráfica 3:** Gráfica de tiempos de acceso, comparando el tiempo real (considerando hits y misses) con los escenarios ideales (todas las referencias en RAM o todas con fallos).

Estas gráficas fueron generadas mediante scripts en Python, que procesan los datos de salida del sistema de simulación.

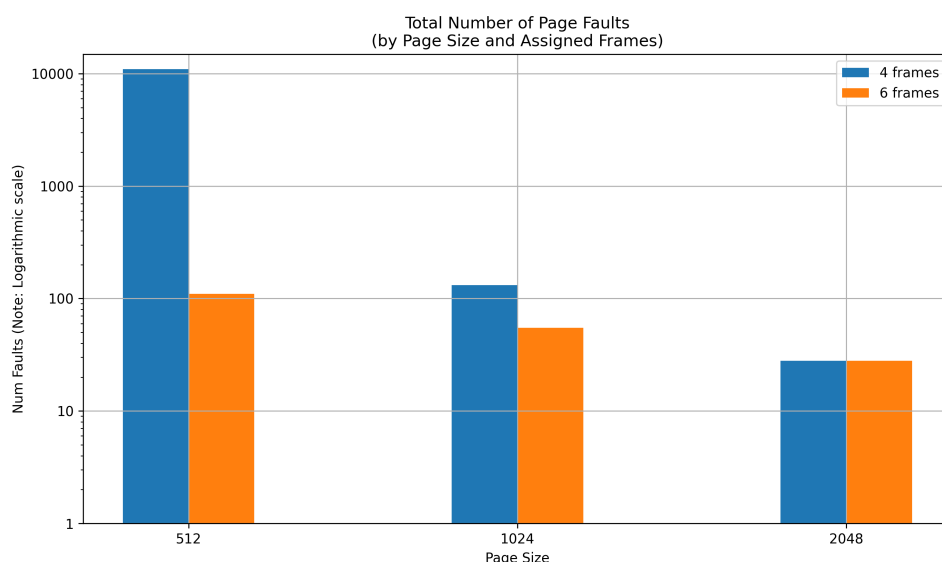


Figure 1: Número de marcos asignados vs. Fallas de página (Misses) para un tamaño de página fijo.

Descripción: Esta gráfica muestra cómo varía el número de fallas de página al modificar el número de marcos asignados en la RAM, manteniendo constante el tamaño de página. Se observa que, al aumentar los marcos, disminuye significativamente la cantidad de misses, lo que evidencia una mejora en la eficiencia del sistema de memoria.

Descripción: En esta gráfica se ilustra la relación entre el número de marcos disponibles y la cantidad de aciertos en memoria. Un incremento en los marcos asignados permite que más referencias sean resueltas directamente en la RAM, aumentando así el número de hits y reduciendo la dependencia de SWAP.

Descripción: Esta gráfica compara los tiempos de acceso del sistema en condiciones reales—donde se mezclan aciertos y fallos de página—con los tiempos teóricos de dos escenarios extremos: cuando todas las referencias se resuelven en RAM y cuando todas generan fallos de página. Se evidencia así el impacto crítico que tienen los fallos de página (debido al elevado tiempo de acceso a SWAP) en el desempeño global del sistema.

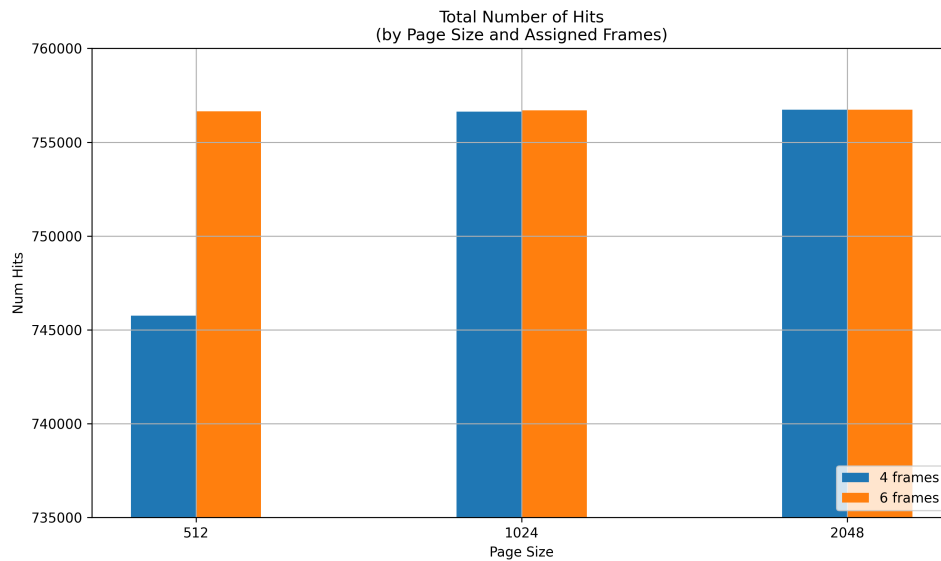


Figure 2: Número de marcos asignados vs. Aciertos (Hits) para el mismo tamaño de página.

Sección 7

Escenarios adicionales y análisis de configuraciones:

Además de los escenarios definidos en el enunciado, se exploraron otras configuraciones:

- Variación del tamaño de la imagen para evaluar su efecto sobre la cantidad de referencias generadas.
- Pruebas con tamaños de página intermedios para identificar puntos óptimos en la tasa de hits.
- Experimentación con distintos números de marcos para verificar la robustez del algoritmo NRU.

Los resultados obtenidos confirman que un mayor número de marcos y tamaños de página mayores tienden a mejorar el desempeño, al reducir significativamente los fallos de página y los tiempos de acceso.

Sección 8

Interpretación de los resultados:

Los datos recopilados son consistentes con las expectativas teóricas: a medida que se incrementa el número de marcos asignados, se observa una reducción notable en los fallos de página, lo que a su vez reduce los tiempos de acceso. Esto se debe a que el algoritmo NRU, al seleccionar la página menos útil, minimiza la penalización asociada a los accesos a SWAP (10 ms) en comparación con los accesos directos a RAM (50 ns). La experimentación confirma la importancia de disponer de suficientes marcos para mantener un alto rendimiento en el sistema de memoria virtual.

Sección 9

Análisis de la localidad en la aplicación del filtro Sobel:

La aplicación del filtro Sobel representa un problema de localidad **alta**. Esto se debe a que,

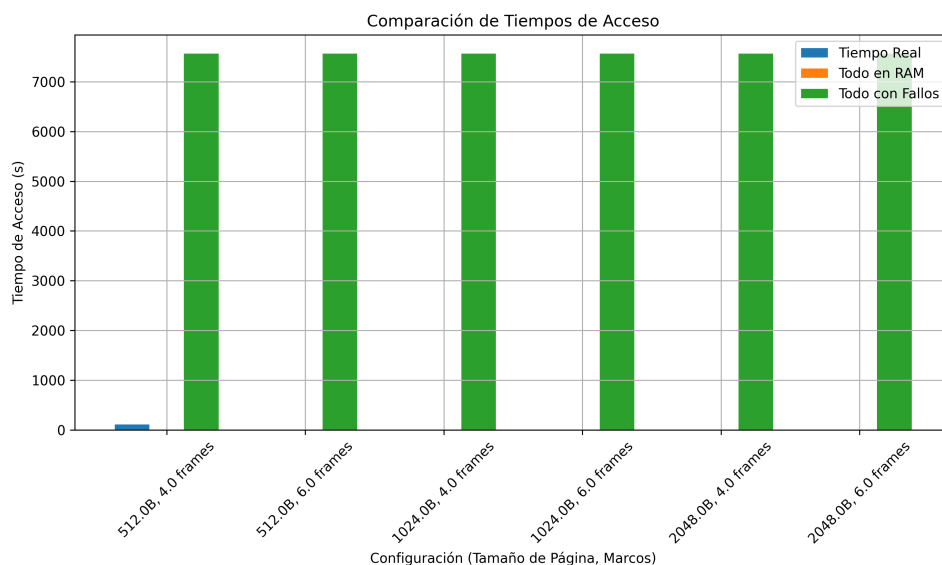


Figure 3: Comparación de tiempos de acceso: Real (hits y misses) vs. Escenarios ideales (todas en RAM o todas con fallos).

para cada píxel, se accede a sus píxeles vecinos en forma secuencial y contigua (orden row-major). Como resultado, las referencias de memoria se agrupan, favoreciendo la reutilización de datos ya cargados en la RAM. Esta alta localidad espacial permite que el sistema registre una mayor cantidad de hits cuando la memoria está bien configurada, mejorando significativamente el desempeño del proceso.

Sección 10

Descripción detallada del algoritmo NRU:

El algoritmo NRU (Not Recently Used) se implementa clasificando las páginas en cuatro clases, basándose en dos bits: el bit de referencia (R) y el bit de modificación (M):

- **Clase 0:** Páginas no referenciadas y no modificadas (la mejor candidata para reemplazo).
- **Clase 1:** Páginas no referenciadas pero modificadas.
- **Clase 2:** Páginas referenciadas y no modificadas.
- **Clase 3:** Páginas referenciadas y modificadas (la peor candidata para reemplazo).

El algoritmo selecciona la página víctima recorriendo la lista de páginas en memoria y eligiendo la primera página disponible en la clase de menor número. Además, un hilo *Inspector* se encarga de reiniciar periódicamente el bit de referencia, permitiendo que la clasificación se actualice dinámicamente.

Sección 11

Detalles de los tiempos calculados:

Se calcularon tres tipos de tiempos de acceso:

- **Tiempo real:** Se considera la suma de los tiempos de acceso de los hits (50 ns cada uno) y los fallos (10 ms cada uno).

- **Tiempo si todas las referencias estuvieran en RAM:** Se asume que cada acceso se resuelve en 50 ns, sin incurrir en fallos.
- **Tiempo si todas las referencias generaran fallos:** Se asume que cada acceso incurre en un fallo, costando 10 ms por referencia.

Estos cálculos permiten comparar el desempeño real con escenarios teóricos ideales y extremos.

Sección 12

Análisis de concurrencia:

El sistema utiliza dos hilos principales:

- **Reader:** Procesa las referencias en lotes de 10,000, seguidos de una espera de 1 ms, simulando el paso del tiempo durante la ejecución del proceso.
- **Inspector:** Se ejecuta cada 1 ms para actualizar los bits de referencia de las páginas, lo que permite que el algoritmo NRU tenga información actualizada sobre el uso reciente de cada página.

Esta coordinación evita condiciones de carrera y asegura la integridad de la tabla de páginas, aunque introduce variaciones temporales que pueden afectar ligeramente los resultados medidos.

Sección 13

Implicaciones para sistemas reales:

Los resultados obtenidos en esta simulación tienen relevancia para el diseño de sistemas operativos reales. En entornos de producción, una administración de memoria eficiente es crucial para minimizar los tiempos de acceso y mejorar el rendimiento global del sistema. La simulación demuestra cómo la elección del tamaño de página y el número de marcos impacta directamente en la tasa de fallos de página y, por ende, en la eficiencia del procesamiento. Además, la estrategia NRU y la gestión concurrente mediante hilos ofrecen ideas valiosas para la optimización de la memoria virtual en sistemas reales.

Sección 14

Conclusiones finales:

En este proyecto se ha implementado una simulación integral de un sistema de administración de memoria virtual utilizando el algoritmo NRU en el contexto de la aplicación del filtro Sobel sobre imágenes BMP. Los resultados demuestran que un mayor número de marcos y un tamaño de página adecuado reducen significativamente los fallos de página, lo que se traduce en tiempos de acceso mucho menores. La coordinación entre el hilo *Reader* y el *Inspector* asegura una actualización precisa de la tabla de páginas, y los distintos escenarios analizados permiten comprender las implicaciones de la memoria virtual en sistemas reales. En síntesis, este ejercicio refuerza la importancia de una buena gestión de la memoria para el rendimiento global de los sistemas operativos.