



Caso 1

Integrantes (Grupo 4, Sección 4)

- Juan José Díaz Ortega - 202220657
- Rodrigo Paz Londoño - 202225425

1. Objetivos

- Entender la importancia de contar con una administración “apropiada” de la memoria: considerando infraestructura de soporte, número de procesos en ejecución, demanda del recurso por proceso y decisiones para adicionar y/o remover marcos de página asignados.
- Construir un prototipo a escala del sistema de administración de memoria virtual que permita simular y evaluar el comportamiento de un proceso de acuerdo con los recursos disponibles.

2. Problemática

La memoria es un recurso limitado que debe administrarse con cuidado para garantizar el avance en la ejecución de los procesos creados. En este contexto surge el concepto de memoria virtual, el cual ofrece varias ventajas: independencia de direcciones físicas, posibilidad de compartir memoria y de correr programas más grandes que la memoria física que se les asigna. Queremos comprender un poco mejor cómo varía el comportamiento de un proceso de acuerdo con la memoria que el sistema le asigna.

Tareas:

1. Escribir un programa que simule la administración de memoria y calcule el número de fallas de página y el porcentaje de *hits* de datos en RAM (un *hit* es la búsqueda de una página que sí está en la RAM y un *miss* es la búsqueda de una página que genera una falla de página).
2. Además, calcule tiempo con *hits* y *misses* vs. tiempo si todas las referencias estuvieran en RAM vs. tiempo si todas las referencias condujeran a fallas de página.
3. Analice los resultados y escriba el informe correspondiente.

3. Contexto

Para simplificar el problema del manejo de memoria, y así poder concentrarnos en entender los retos asociados, estudiaremos las referencias de páginas generadas por un solo proceso que recupera un mensaje escondido en una imagen. Este proceso es conocido como esteganografía; a continuación, se incluye la definición de la Real Academia Española (RAE): "Técnica criptográfica que consiste en ocultar mensajes en archivos digitales."

Ejemplo: Ocultar en la figura 1 el mensaje de la figura 2

Figure 1

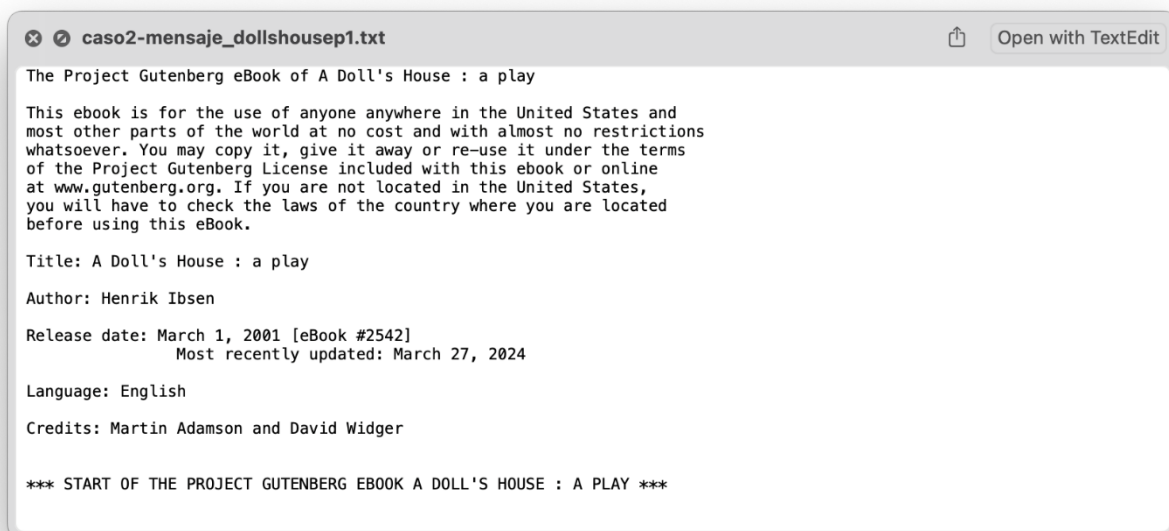
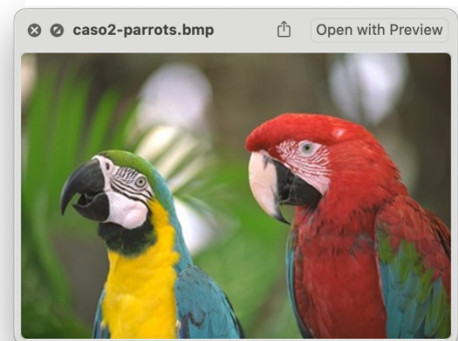


Figure 2

El formato BMP de 24 bits permite almacenar imágenes guardando cada uno de sus píxeles. Para ello, se representa cada píxel de la imagen utilizando tres bytes; cada byte guarda un valor que corresponde a un color: un byte para el rojo (R), uno para el verde (G) y uno para el azul (B). Al combinar estos tres valores, se forma el color final del píxel.

En este caso particular, el mensaje se esconderá distribuido bit por bit, en el bit menos significativo de cada byte de una imagen. Usar el bit menos significativo es conveniente porque de esta manera las diferencias de color, entre la imagen original y la imagen que esconde el mensaje, son imperceptibles.

4. Tenga en cuenta

Por desempeño, no es conveniente usar imágenes muy grandes. Imágenes de 500 x 300 píxeles son suficientes. ¿Cuál es la longitud máxima de mensaje que puede almacenar una imagen de este tamaño?

Cada píxel en una imagen BMP de 24 bits se compone de tres bytes, uno para cada color del RGB y cada uno de estos bytes puede almacenar 1 bit del mensaje en su bit menos significativo.

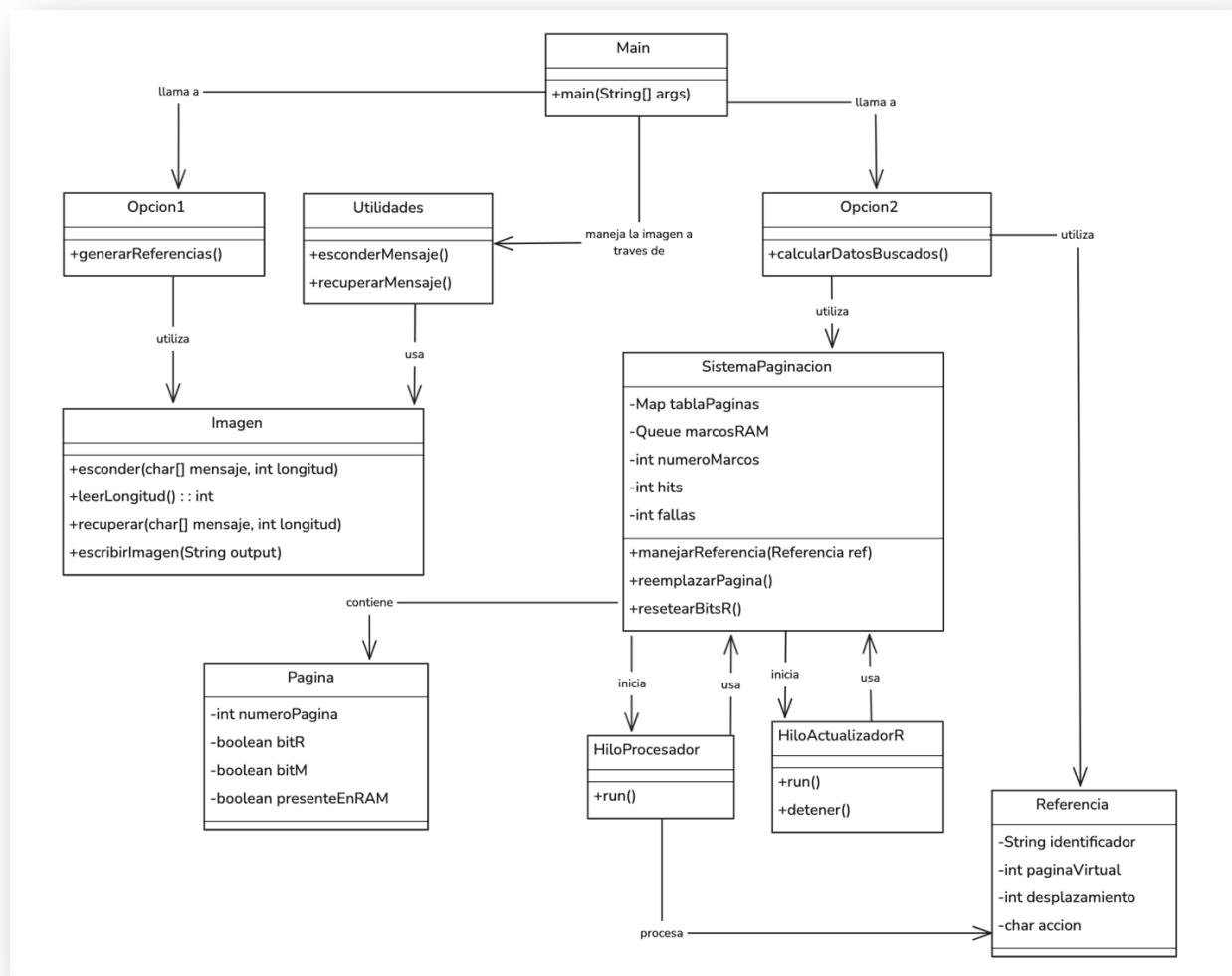
Total de píxeles = $500 \times 300 = 150,000$ píxeles

Total de bytes = $150,000 \times 3 = 450,000$ bytes

Longitud máxima del mensaje = $450,000/8 = 56,250$ caracteres

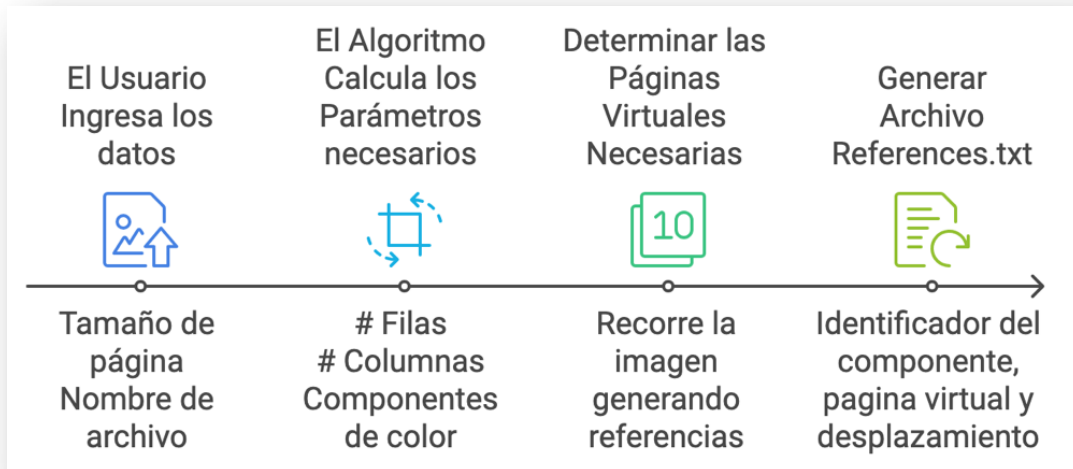
Entonces la longitud máxima del mensaje que se puede almacenar en una imagen de 500 x 300 píxeles es **56,250** caracteres.

5. Diseño (Diagrama de clases)

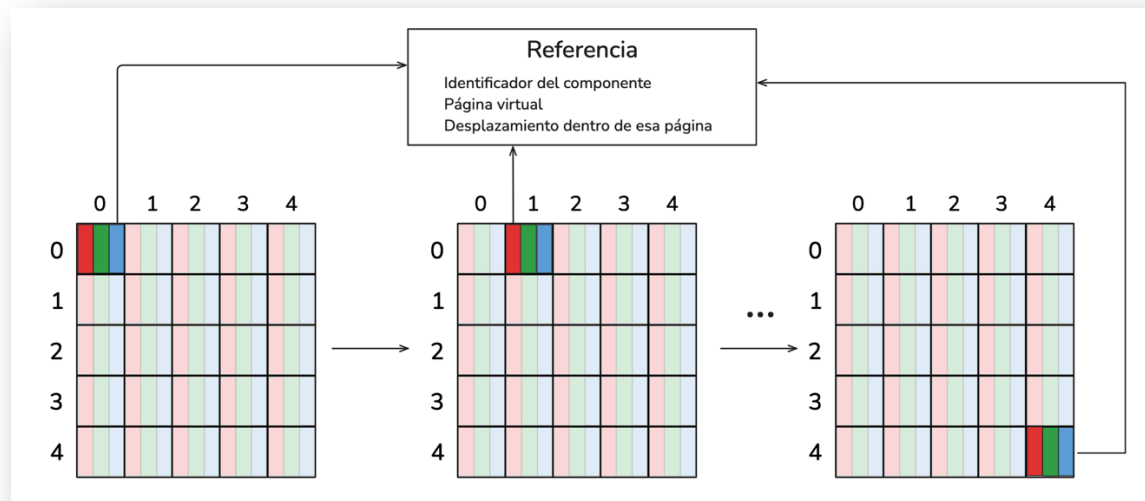


6. Descripción del algoritmo usado para generar las referencias de página. Opc 1

El algoritmo tiene como objetivo generar un archivo de referencias de página para una imagen que contiene un mensaje oculto. Primero, el usuario ingresa el tamaño de página y el nombre del archivo de la imagen. Con estos datos, el algoritmo calcula el número de filas, columnas y componentes de color de la imagen (rojo, verde y azul), y determina cuántas páginas virtuales serán necesarias para almacenar todas las referencias. Luego, se genera un archivo llamado "referencias.txt" que contiene información sobre el tamaño de página, el número de filas y columnas, el número total de referencias y las páginas virtuales necesarias. El algoritmo se ejemplifica mejor en la siguiente gráfica:

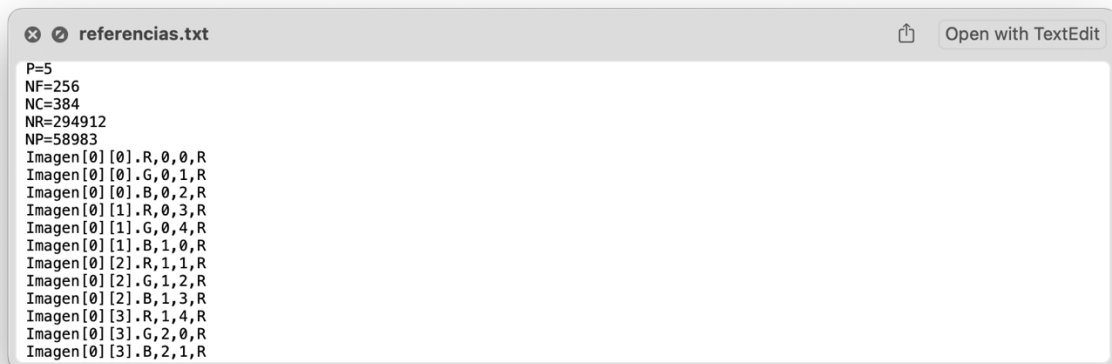


El algoritmo recorre la matriz de la imagen por filas y columnas, generando una referencia para cada componente de color (R, G, B) de cada píxel. Cada referencia incluye el identificador del componente, la página virtual y el desplazamiento dentro de esa página. Estas referencias se almacenan en el archivo de salida, organizando la información de manera que el método de recuperación pueda acceder eficientemente a los datos de la imagen según el esquema de paginación definido.



7. Descripción de las estructuras de datos usadas para simular el comportamiento del sistema de paginación y cómo usa dichas estructuras.

El código utiliza un archivo de texto para simular el comportamiento del sistema de paginación. Este archivo almacena una lista secuencial de referencias de la imagen, donde cada referencia corresponde a un componente de color (R, G, B) de un píxel. Las referencias incluyen el identificador del píxel, el número de página virtual calculado, el desplazamiento dentro de la página, y el tipo de operación. La generación de estas referencias se realiza recorriendo la imagen píxel por píxel, y calculando la página virtual y el desplazamiento con base en el tamaño de la página ingresado por el usuario. En la siguiente imagen se muestra el resultado de la opción 1 al ingresarle parrots.bmp:



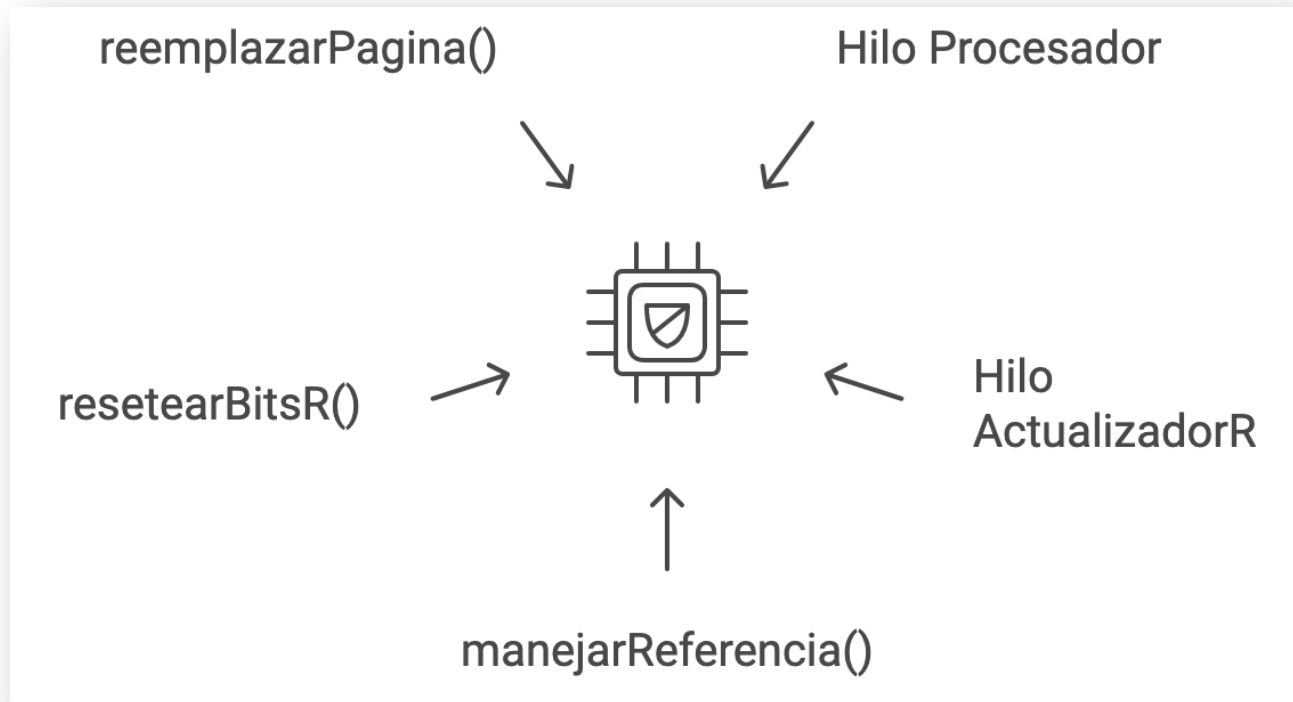
```
P=5
NF=256
NC=384
NR=294912
NP=58983
Imagen[0][0].R,0,0,R
Imagen[0][0].G,0,1,R
Imagen[0][0].B,0,2,R
Imagen[0][1].R,0,3,R
Imagen[0][1].G,0,4,R
Imagen[0][1].B,1,0,R
Imagen[0][2].R,1,1,R
Imagen[0][2].G,1,2,R
Imagen[0][2].B,1,3,R
Imagen[0][3].R,1,4,R
Imagen[0][3].G,2,0,R
Imagen[0][3].B,2,1,R
```

Las estructuras de datos se actualizan cuando se escribe cada referencia al archivo durante la iteración sobre los píxeles. El cálculo de la página virtual se basa en la posición lineal del componente dentro de la imagen, usando la relación fila, columna y color, mientras que el desplazamiento indica la posición dentro de la página virtual. Esta simulación permite tener una representación estática de cómo se organizan y localizan las referencias de memoria en un sistema de paginación.



8. Esquema de sincronización usado. Justifique brevemente dónde es necesario usar sincronización y por qué.

En el caso se utiliza sincronización en los métodos de la clase SistemaPaginacion para evitar condiciones de carrera entre los hilos Hilo Procesador y Hilo ActualizadorR, que acceden y modifican la misma tabla de páginas y marcos de RAM. Los métodos manejarReferencia(), resetearBitsR() y reemplazarPagina() están sincronizados porque gestionan el acceso concurrente a estos recursos compartidos, como se muestra en la siguiente gráfica.



Sin esta sincronización, podría haber inconsistencias en la actualización de los bits R y M, así como en el manejo de fallas de página, lo que afectaría la integridad del sistema de paginación.

9. Una tabla con los datos recopilados (y porcentaje de hits y misses por cada escenario simulado).

Para los casos usamos 2 archivos de tamaños diferentes, los samples de estos archivos los obtuvimos de <https://filesampleshub.com/format/image/bmp> y utilizamos el de 804 KB y 11 MB. A continuación, se muestran los 60 escenarios, 20 de los cuales eran los que pedían en el enunciado y 40 más de los escenarios extras que piden en el punto 12.

Para páginas de tamaño 512:

Tamaño de pagina	Nombre de imagen	Tamaño de Imagen	Tamaño de Mensaje	Marcos Asignados	Total Referencias	Hits	Fallas	Tiempo Total (ms)	%Hits	%Fallas
512	ImgSmall	640x 427	100	4	1733	1730	3	20,043325	99,83%	0,17%
512	ImgSmall	640x 427	100	8	1733	1730	3	20,043325	99,83%	0,17%
512	ImgSmall	640x 427	1000	4	17016	16995	21	210,4254	99,88%	0,12%
512	ImgSmall	640x 427	1000	8	17016	16997	19	190,4254	99,89%	0,11%
512	ImgSmall	640x 427	2000	4	34016	33973	43	430,850400	99,87%	0,13%
512	ImgSmall	640x 427	2000	8	34016	33977	39	390,8504000	99,89%	0,11%
512	ImgSmall	640x 427	4000	4	68016	67930	86	861,700400	99,87%	0,13%
512	ImgSmall	640x 427	4000	8	68016	67938	78	781,700400	99,89%	0,11%
512	ImgSmall	640x 427	8000	4	136016	135843	173	1733,4004	99,87%	0,13%
512	ImgSmall	1920 x 1280	8000	8	136016	135859	157	1573,400400	99,88%	0,12%
512	ImgBig	1920 x 1280	100	4	1733	1730	3	20,043325	99,83%	0,17%
512	ImgBig	1920 x 1280	100	8	1733	1730	3	20,043325	99,83%	0,17%
512	ImgBig	1920 x 1280	1000	4	17016	16995	21	210,4254	99,88%	0,12%
512	ImgBig	1920 x 1280	1000	8	17016	16997	19	190,4254	99,89%	0,11%
512	ImgBig	1920 x 1280	2000	4	34016	33973	43	430,8504	99,87%	0,13%
512	ImgBig	1920 x 1280	2000	8	34016	33977	39	390,8504	99,89%	0,11%
512	ImgBig	1920 x 1280	4000	4	68016	67930	86	861,7004	99,87%	0,13%
512	ImgBig	1920 x 1280	4000	8	68016	67938	78	781,7004	99,89%	0,11%
512	ImgBig	1920 x 1280	8000	4	136016	135843	173	1733,4004	99,87%	0,13%
512	ImgBig	1920 x 1280	8000	8	136016	135859	157	1573,4004	99,88%	0,12%

Para páginas de tamaño 1024:

Tamaño de pagina	Nombre de imagen	Tamaño de Imagen	Tamaño de Mensaje	Marcos Asignados	Total Referencias	Hits	Fallas	Tiempo Total (ms)	%Hits	%Fallas
1024	ImgSmall	640x 427	100	4	1733	1731	2	20,04332500	99,88%	0,12%
1024	ImgSmall	640x 427	100	8	1733	1731	2	20,04332500	99,88%	0,12%
1024	ImgSmall	640x 427	1000	4	17016	17006	10	100,42540000	99,94%	0,06%
1024	ImgSmall	640x 427	1000	8	17016	17007	9	90,42540000	99,95%	0,05%
1024	ImgSmall	640x 427	2000	4	34016	33995	21	210,85040000	99,94%	0,06%
1024	ImgSmall	640x 427	2000	8	34016	33997	19	190,85040000	99,94%	0,06%
1024	ImgSmall	640x 427	4000	4	68016	67973	43	431,700400	99,94%	0,06%
1024	ImgSmall	640x 427	4000	8	68016	67977	39	391,70040000	99,94%	0,06%
1024	ImgSmall	640x 427	8000	4	136016	135930	86	863,40040000	99,94%	0,06%
1024	ImgSmall	640x 427	8000	8	136016	135938	78	783,40040000	99,94%	0,06%
1024	ImgBig	1920 x 1280	100	4	1733	1731	2	20,043325	99,88%	0,12%
1024	ImgBig	1920 x 1280	100	8	1733	1731	2	20,043325	99,88%	0,12%
1024	ImgBig	1920 x 1280	1000	4	17016	17006	10	100,4254	99,94%	0,06%
1024	ImgBig	1920 x 1280	1000	8	17016	17007	9	90,4254	99,95%	0,05%
1024	ImgBig	1920 x 1280	2000	4	34016	33995	21	210,8504	99,94%	0,06%
1024	ImgBig	1920 x 1280	2000	8	34016	33997	19	190,85040000	99,94%	0,06%
1024	ImgBig	1920 x 1280	4000	4	68016	67973	43	431,700400	99,94%	0,06%
1024	ImgBig	1920 x 1280	4000	8	68016	67977	39	391,7004	99,94%	0,06%
1024	ImgBig	1920 x 1280	8000	4	136016	135930	86	863,4004	99,94%	0,06%
1024	ImgBig	1920 x 1280	8000	8	136016	135938	78	783,4004	99,94%	0,06%

Para páginas de tamaño 2048:

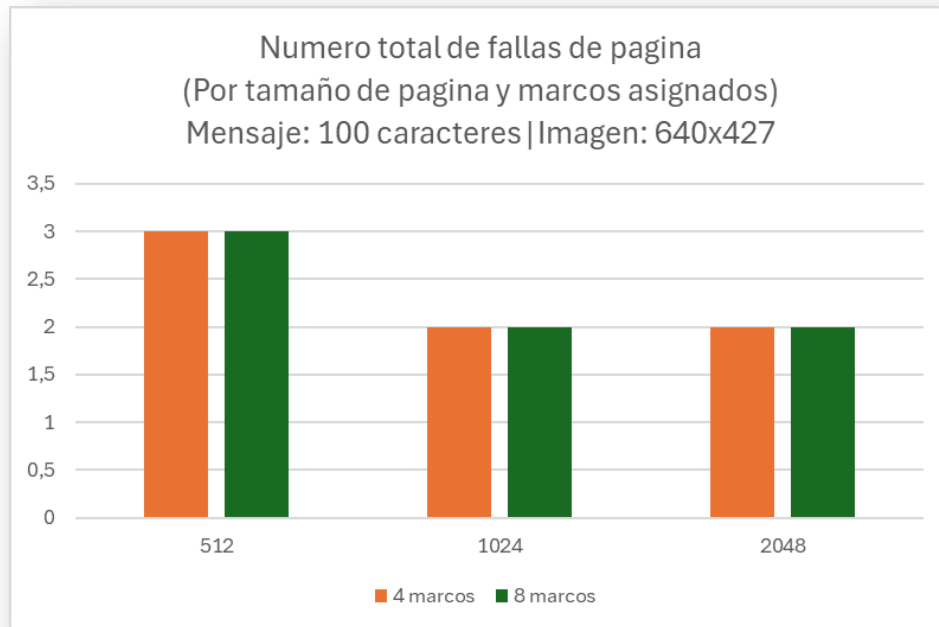
Tamaño de pagina	Nombre de imagen	Tamaño de Imagen	Tamaño de Mensaje	Marcos Asignados	Total Referencias	Hits	Fallas	Tiempo Total (ms)	%Hits	%Fallas
2048	ImgSmall	640x 427	100	4	1733	1731	2	20,043	99,88%	0,12%
2048	ImgSmall	640x 427	100	8	1733	1731	2	20,043	99,88%	0,12%
2048	ImgSmall	640x 427	1000	4	17016	17011	5	50,425	99,97%	0,03%
2048	ImgSmall	640x 427	1000	8	17016	17011	5	50,425	99,97%	0,03%
2048	ImgSmall	640x 427	2000	4	34016	34006	10	100,850	99,97%	0,03%
2048	ImgSmall	640x 427	2000	8	34016	34007	9	90,850	99,97%	0,03%
2048	ImgSmall	640x 427	4000	4	68016	67973	43	431,700	99,94%	0,06%
2048	ImgSmall	640x 427	4000	8	68016	67977	39	391,700	99,94%	0,06%
2048	ImgSmall	640x 427	8000	4	136016	135973	43	433,400	99,97%	0,03%
2048	ImgSmall	640x 427	8000	8	136016	135977	39	393,400	99,97%	0,03%
2048	ImgBig	1920 x 1280	100	4	1733	1731	2	20,043325	99,88%	0,12%
2048	ImgBig	1920 x 1280	100	8	1733	1731	2	20,043325	99,88%	0,12%
2048	ImgBig	1920 x 1280	1000	4	17016	17011	5	50,4254	99,97%	0,03%
2048	ImgBig	1920 x 1280	1000	8	17016	17011	5	50,4254	99,97%	0,03%
2048	ImgBig	1920 x 1280	2000	4	34016	34006	10	100,8504	99,97%	0,03%
2048	ImgBig	1920 x 1280	2000	8	34016	34007	9	190,4254	99,97%	0,03%
2048	ImgBig	1920 x 1280	4000	4	68016	67973	43	431,7004	99,94%	0,06%
2048	ImgBig	1920 x 1280	4000	8	68016	67977	39	391,7004	99,94%	0,06%
2048	ImgBig	1920 x 1280	8000	4	136016	135973	43	433,4004	99,97%	0,03%
2048	ImgBig	1920 x 1280	8000	8	136016	135977	39	393,4004	99,97%	0,03%

10. Una serie de gráficas que ilustren el comportamiento del sistema.

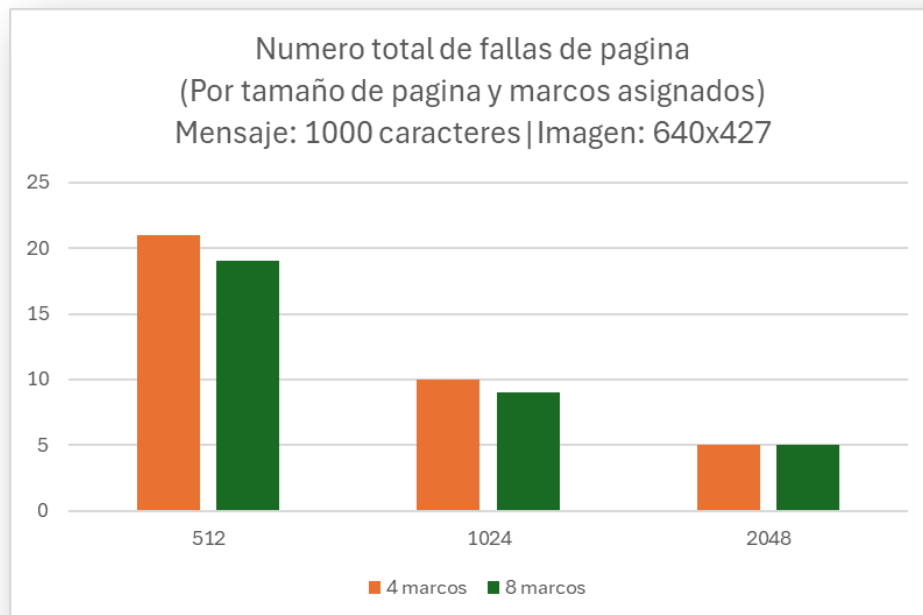
Graficas por tamaño de mensaje e imagen:

Imagen pequeña 640x427

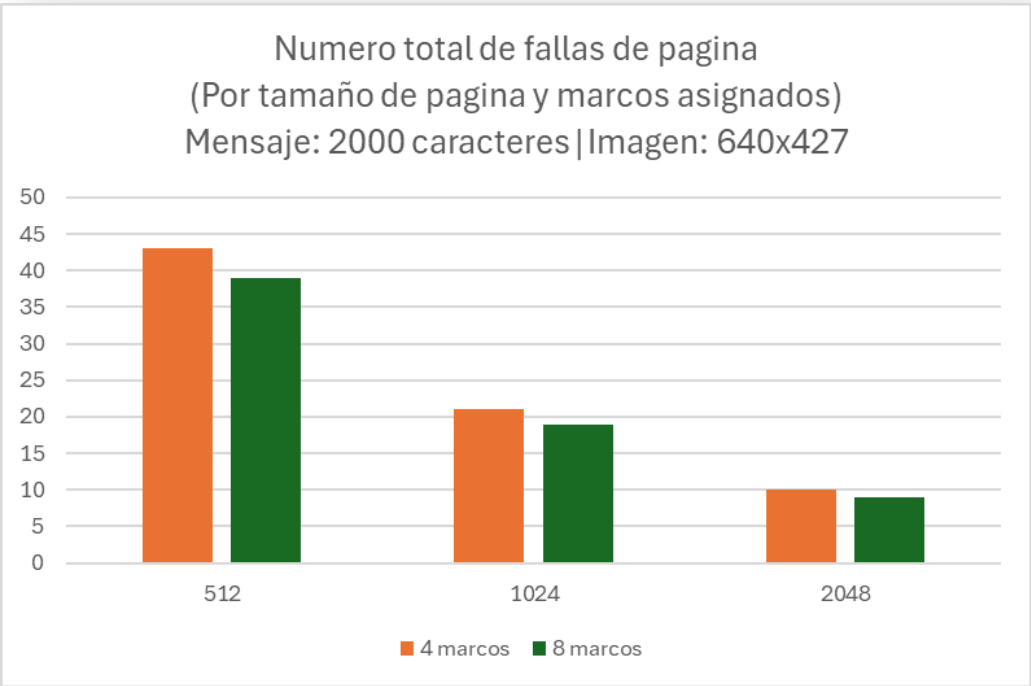
100 Caracteres:



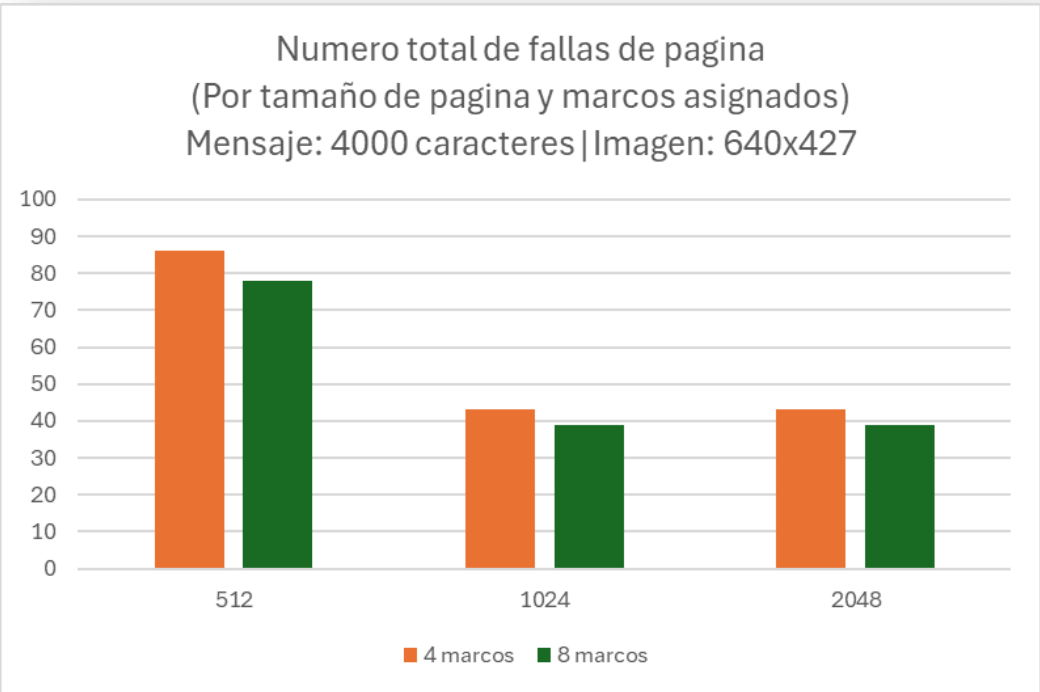
1000 Caracteres:



2000 Caracteres:



4000 Caracteres:



8000 Caracteres:

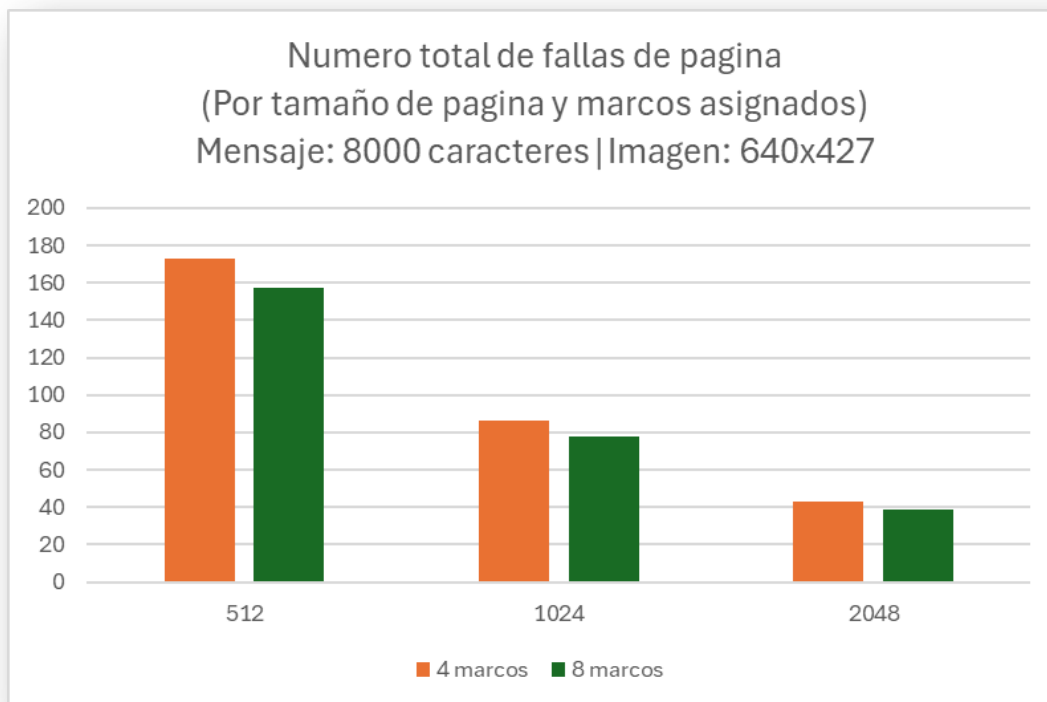
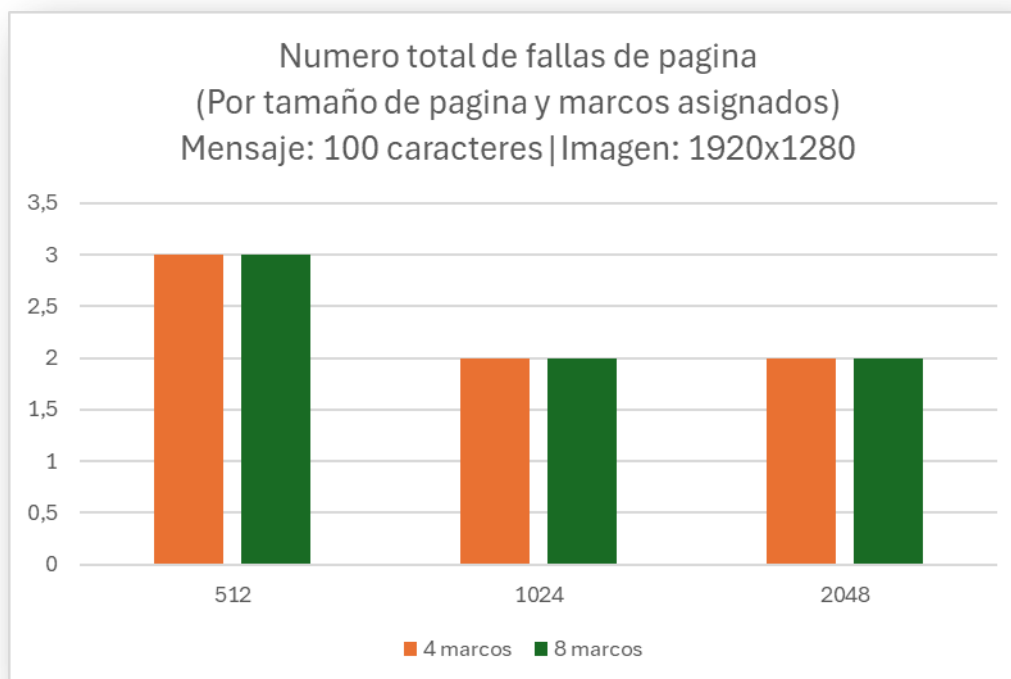
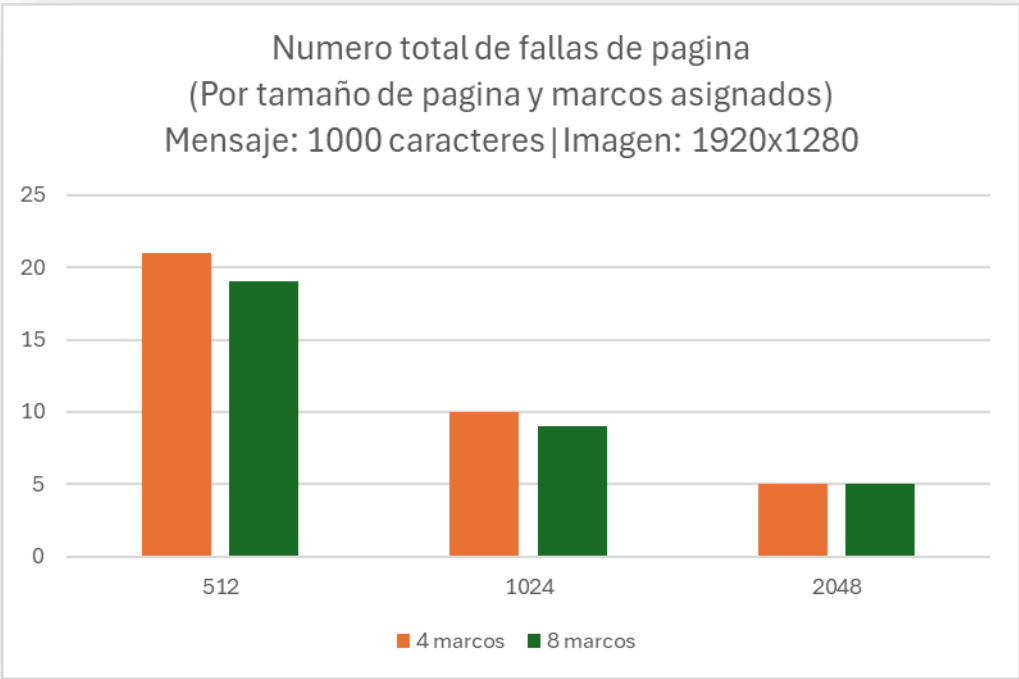


Imagen grande 1920x1280

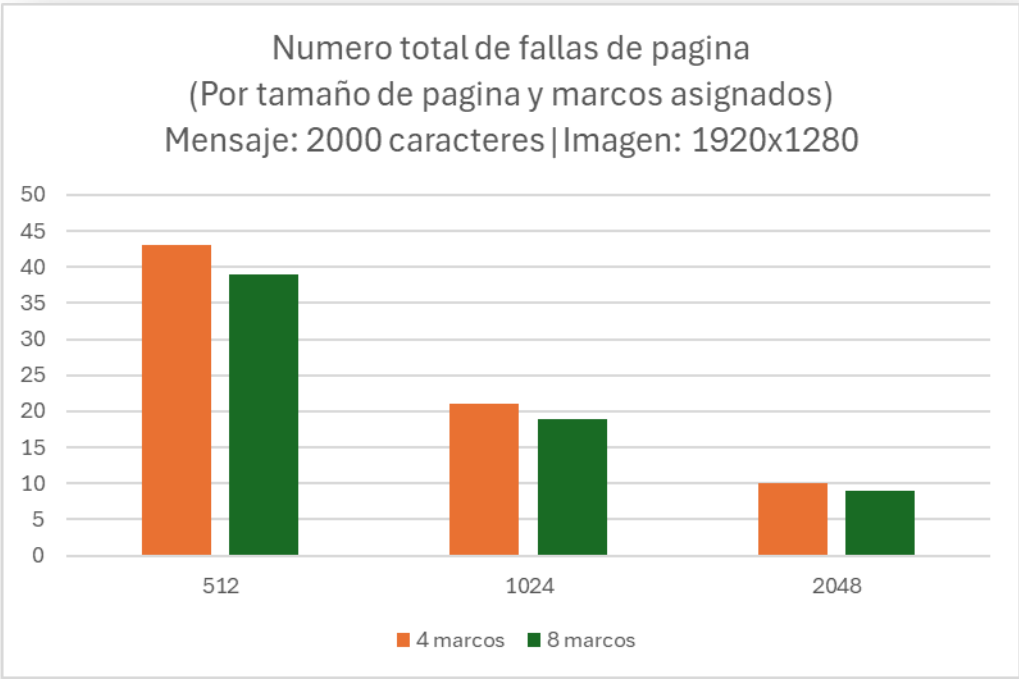
100 caracteres:



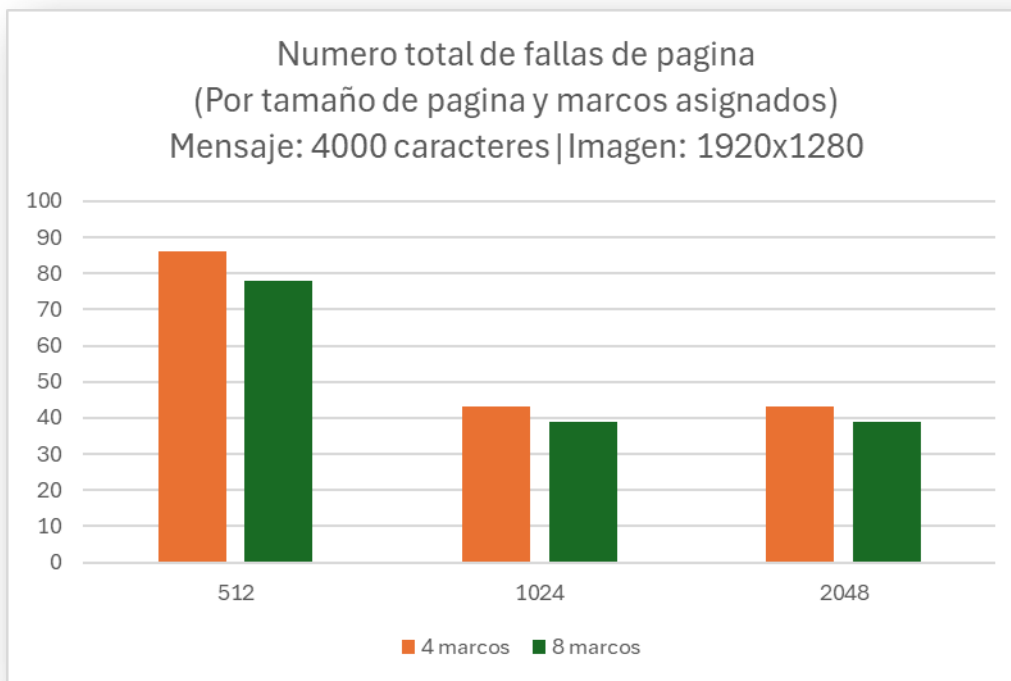
1000 caracteres:



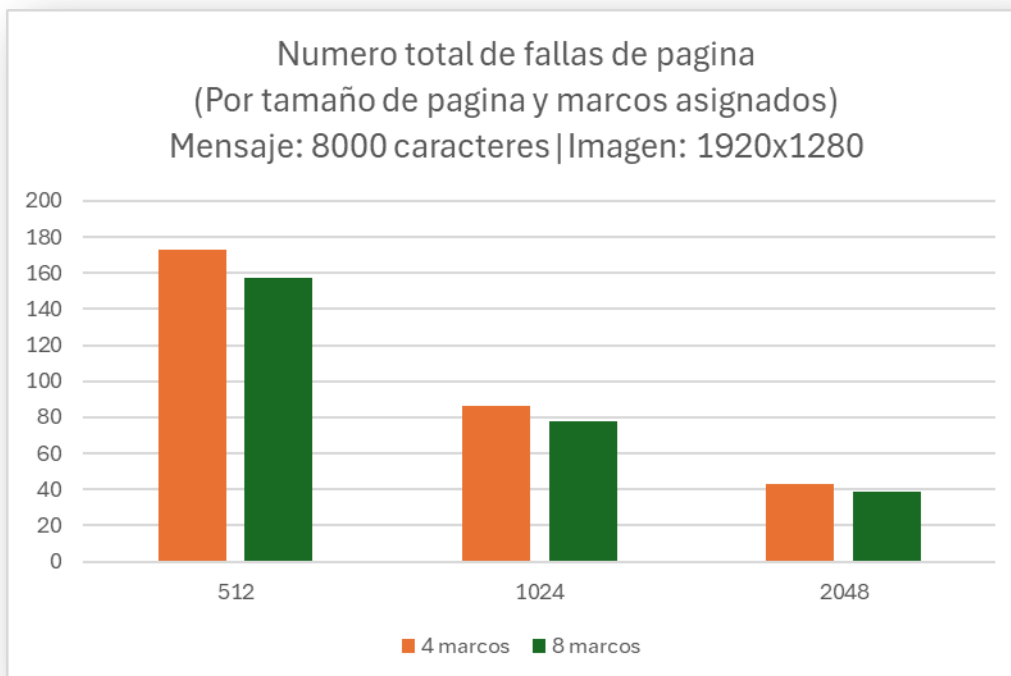
2000 caracteres:



4000 caracteres:

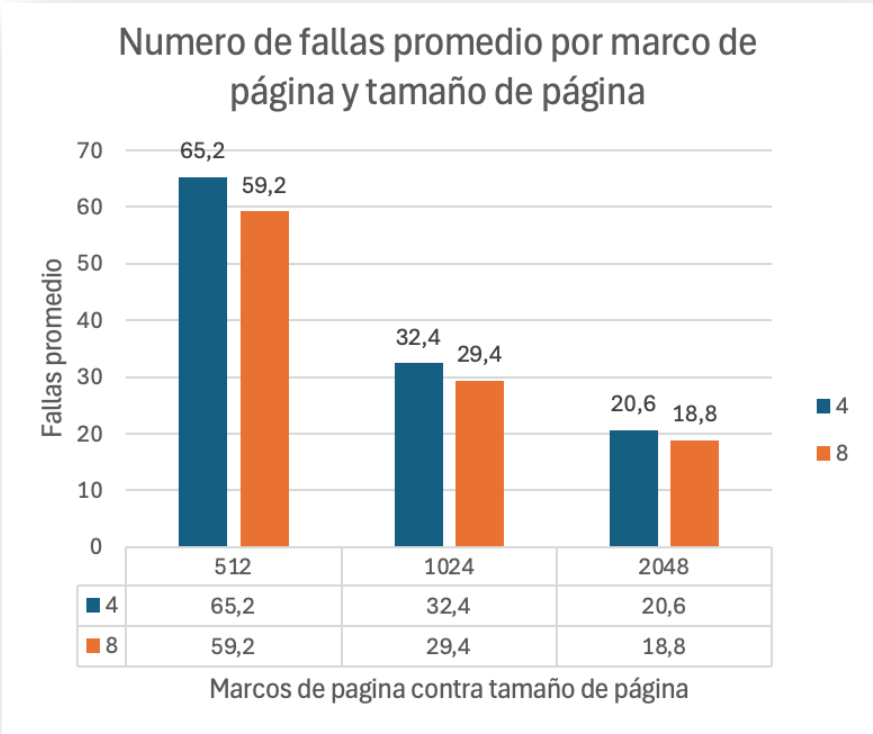


8000 caracteres:

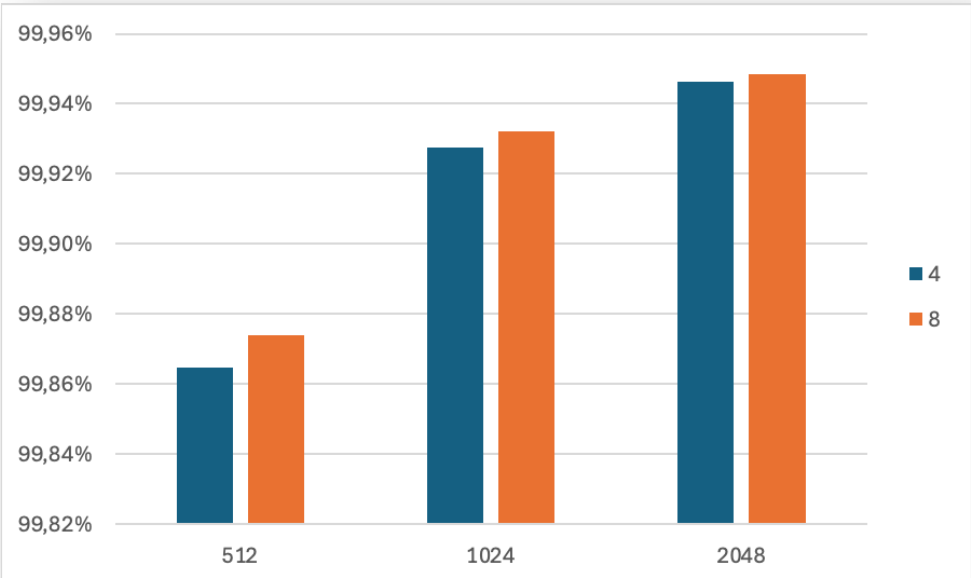


Se observa que a menor tamaño de página, mayor fallas de página, lo cual es totalmente lógico y predecible. Continuaremos con gráficas más generales.

Promedio de los 60 casos:

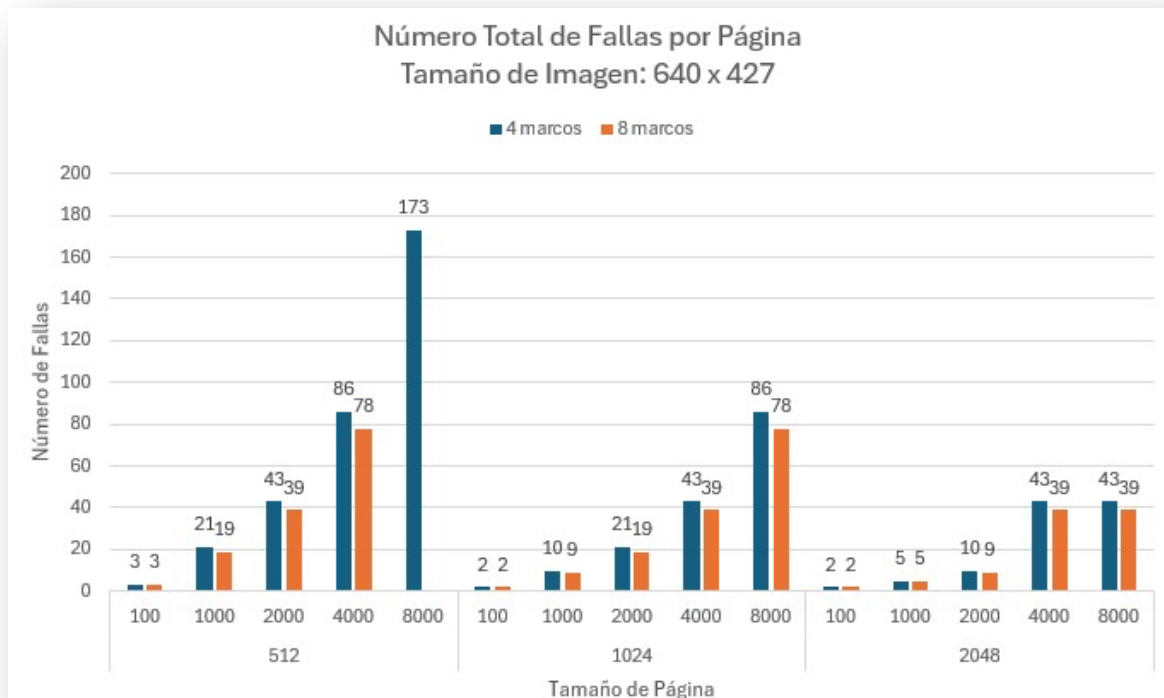


Porcentaje de hits promedio para cada tamaño de página:

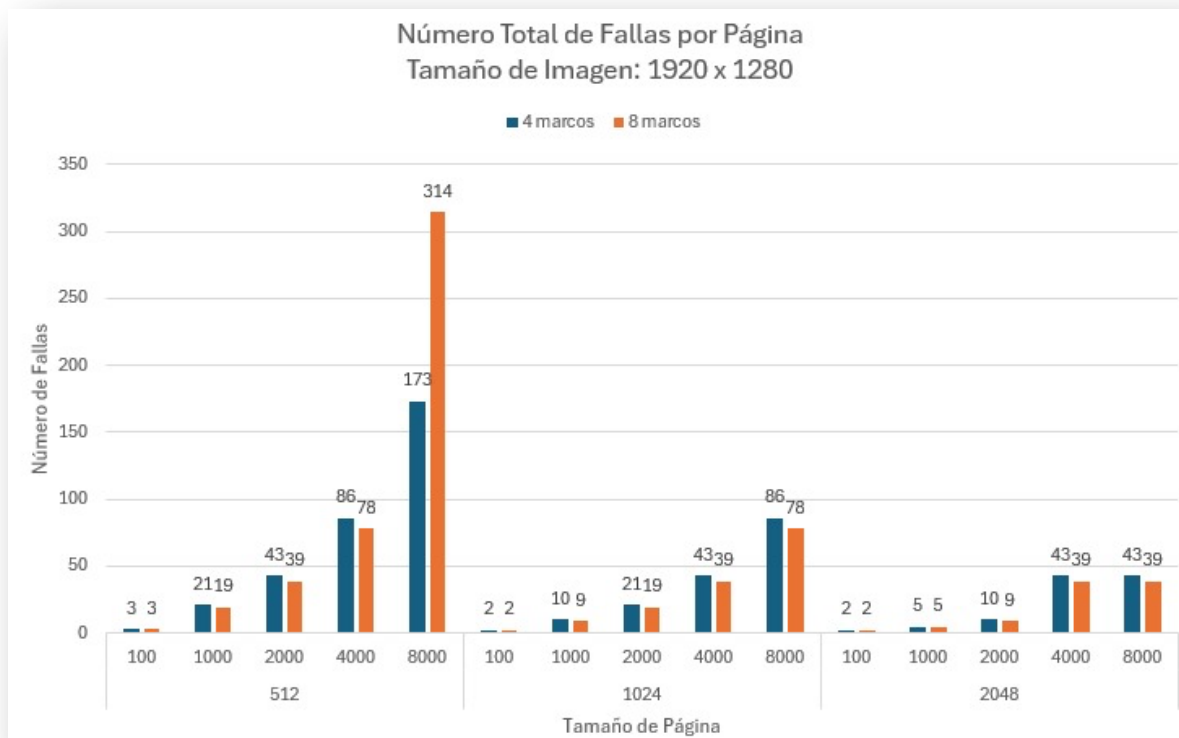


Como se puede ver en las anteriores gráficas, el comportamiento del sistema se comporta de la siguiente manera: a menor tamaño de página, más fallas y a mayor tamaño de página mayor porcentaje de hits.

Ahora procedemos a poner graficas que agrupan por tamaño de imagen para ver como este parámetro afecta al resto del programa:

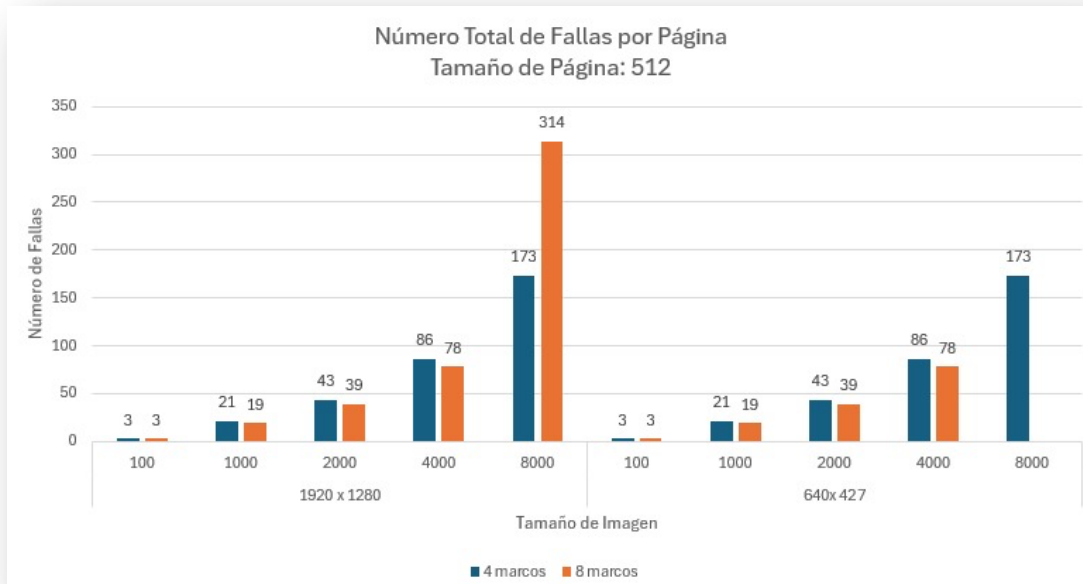


Para el tamaño más grande:

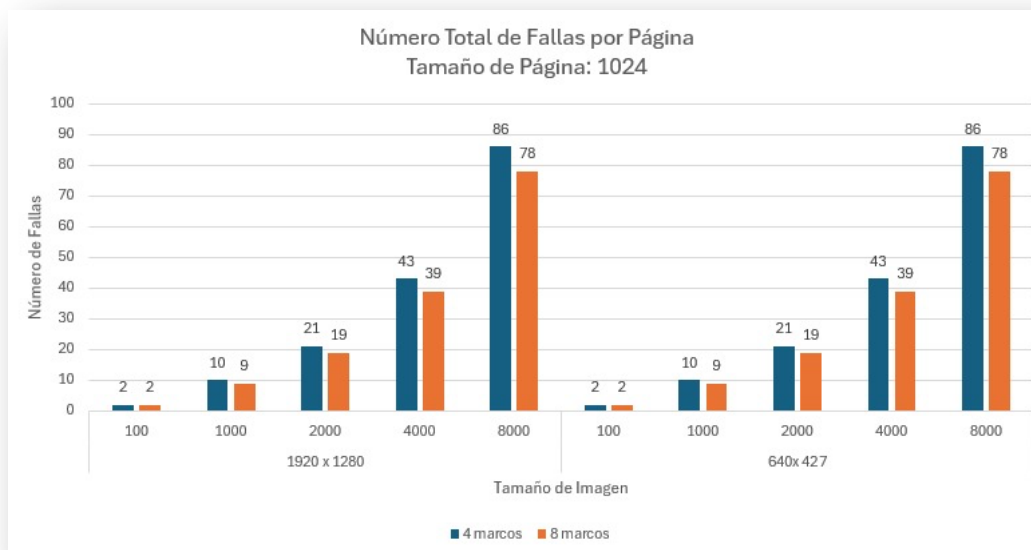


11. Corra los escenarios y genere gráficas que muestren los datos recopilados para los diferentes escenarios.

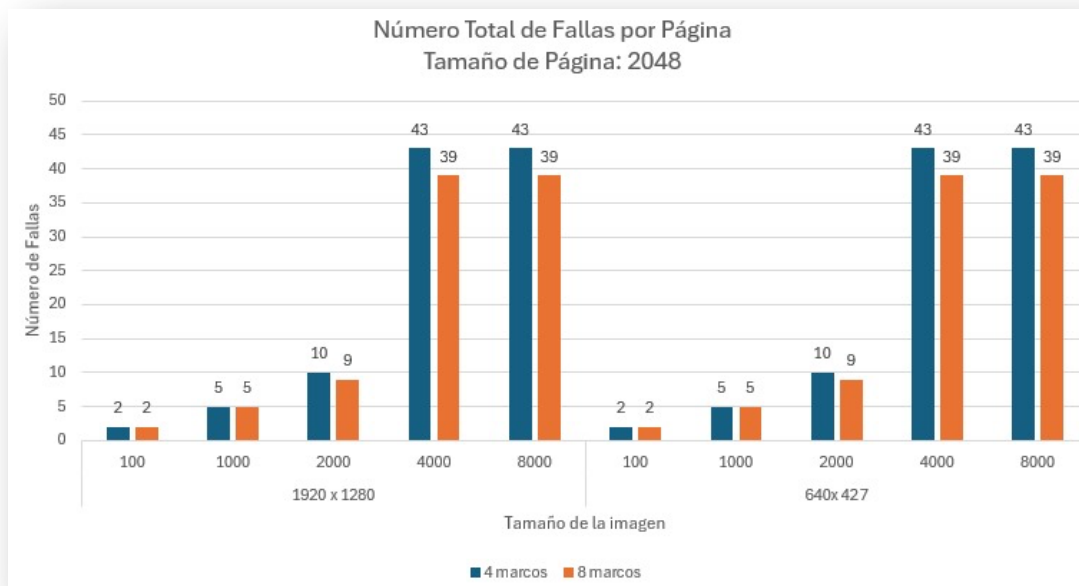
Ahora vamos a hacer diferentes graficas más detalladas de los escenarios. Para 512:



Para 1024:

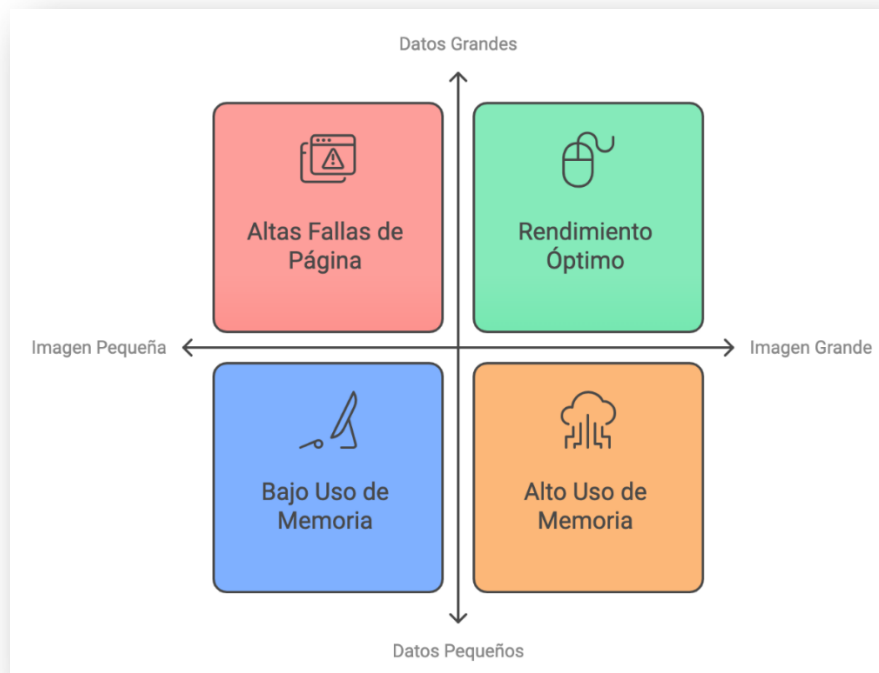


Para 2048:



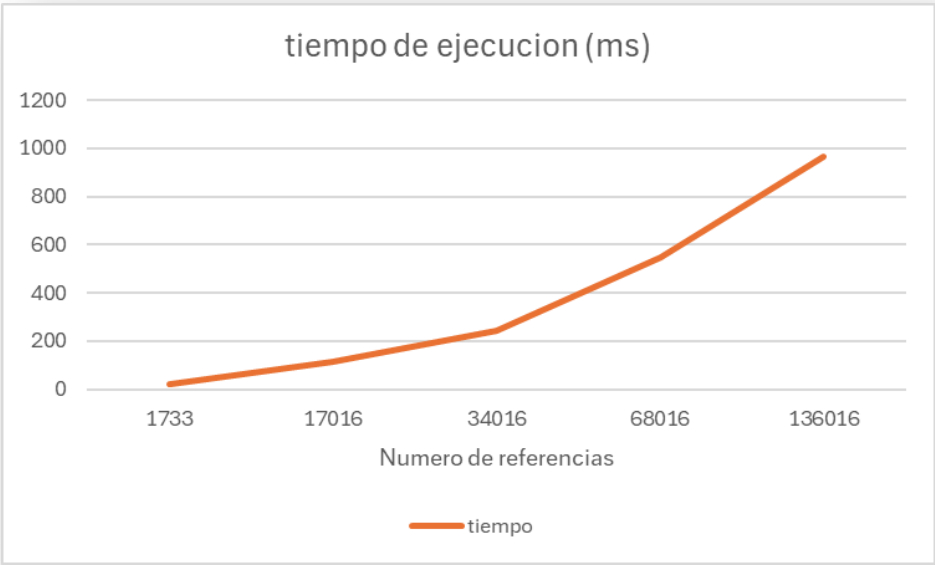
12. Además de los escenarios definidos, considere otras configuraciones que le permitan entender cómo afecta la memoria virtual el desempeño del programa.

Diseñamos 40 escenarios más para entender cómo se afecta el desempeño incluyendo diferentes tamaños de mensaje y tamaño de imagen. Después de ver su comportamiento en las gráficas anteriores podemos concluir que la memoria virtual y el desempeño del programa se comportan de la siguiente manera:

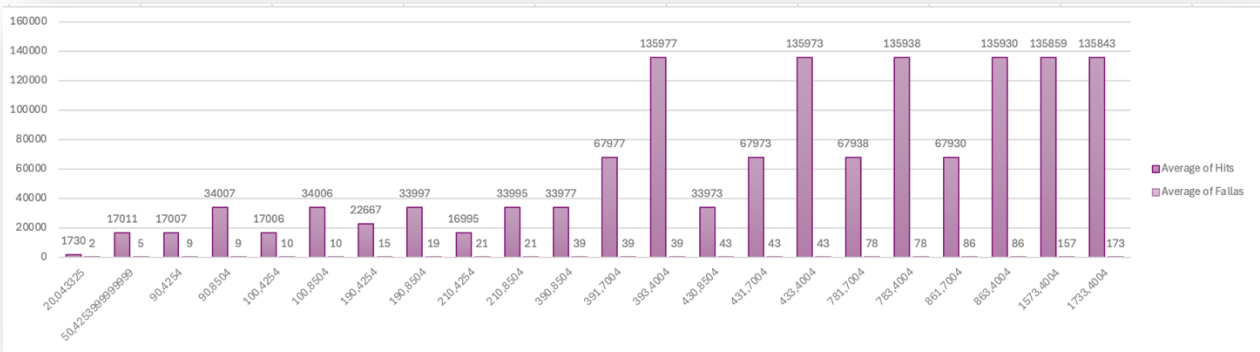


13.Incluya las gráficas de tiempo (hits, misses, total).

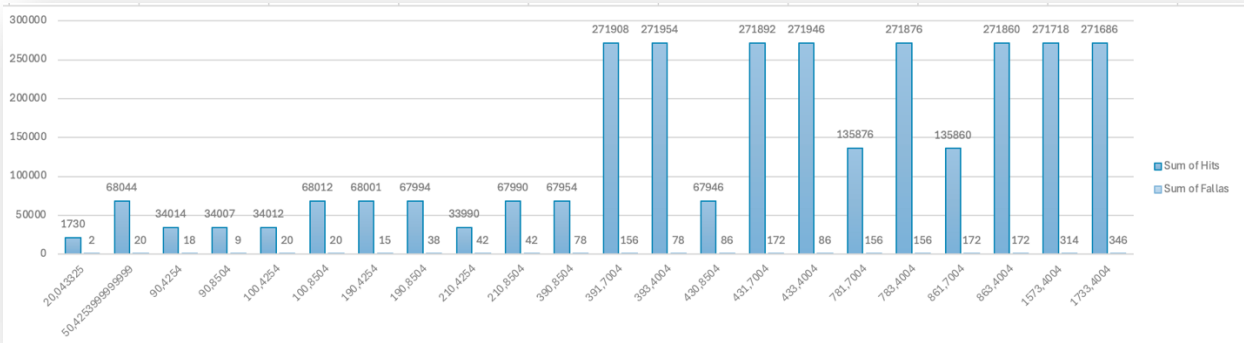
Tiempo de ejecución contra número de referencias:



Tiempo de ejecución contra hits y misses promedio:



Tiempo de ejecución contra hits y misses total:



14. Escriba su interpretación de los resultados: ¿corresponden a los resultados que esperaba, con respecto al número de marcos asignados? Explique su respuesta.

Con base en los resultados de las gráficas, se puede interpretar que el comportamiento del sistema corresponde a lo esperado con respecto al número de marcos asignados, esto lo podemos argumentar por las siguientes razones.

1. Cuando se asignan más marcos de página (8 en lugar de 4), el sistema tiende a tener menos fallas de página. Esto es lógico porque con más marcos disponibles, más páginas pueden permanecer en la memoria, reduciendo la necesidad de reemplazos.
2. A medida que el tamaño de la página aumenta (de 512 a 2048 bytes), también se reducen las fallas de manera extraordinaria, ya que se pueden almacenar más datos en cada página, lo ideal sería hacer esto sin embargo en la práctica no se hace porque lleva a desperdicio de espacio si las referencias no ocupan completamente las páginas.
3. Los mensajes más grandes y las imágenes de mayor resolución requieren más memoria, lo que puede incrementar las fallas si no se asignan suficientes marcos. Se observa en las gráficas que para tamaños de página pequeños y marcos limitados, las fallas aumentan significativamente.

Concluimos entonces que sí, eran los resultados que esperábamos de manera teórica, lo único que nos sorprendió fue que el número de fallas nunca fue lo suficientemente alto o al menos en porcentaje, ya que en los ejemplos de clase por ejemplo, en una pequeña cadena usando cualquiera de los algoritmos de reemplazo ya sea LRU o FIFO daban un montón de fallos, claramente son ejemplos totalmente distintos pero teníamos la concepción errónea de que habrían 50 / 50 en fallos contra hits pero en realidad no es así.

15. ¿Si la localidad del problema manejado fuera diferente cómo variarían los resultados? Explique su respuesta.

Con una **mayor localidad**, el programa tiende a acceder repetidamente a las mismas páginas o a páginas cercanas en un corto período de tiempo. Esto aumentaría el número de hits (cuando una página solicitada ya está en RAM) y disminuiría las fallas de página, ya que las páginas accedidas permanecerían más tiempo en la memoria. El sistema de paginación sería más eficiente, y se observaría un mejor desempeño debido a un menor número de accesos a disco o a la memoria secundaria.

En cambio, con una **menor localidad**, el programa accede a páginas de manera dispersa o sin reutilizar las mismas páginas en intervalos cortos. Esto aumentaría las fallas de página, ya que el sistema necesitaría cargar nuevas páginas más frecuentemente, expulsando páginas que podrían ser reutilizadas en el futuro. Se incrementaría el número de accesos a disco, lo que afectaría negativamente el desempeño del sistema.



Podemos resumirlo en este gráfico: A mayor localidad más hits y a menor más fallas.