

Informe Caso 2

Infraestructura Computacional

Santiago Casasbuenas - 202214932

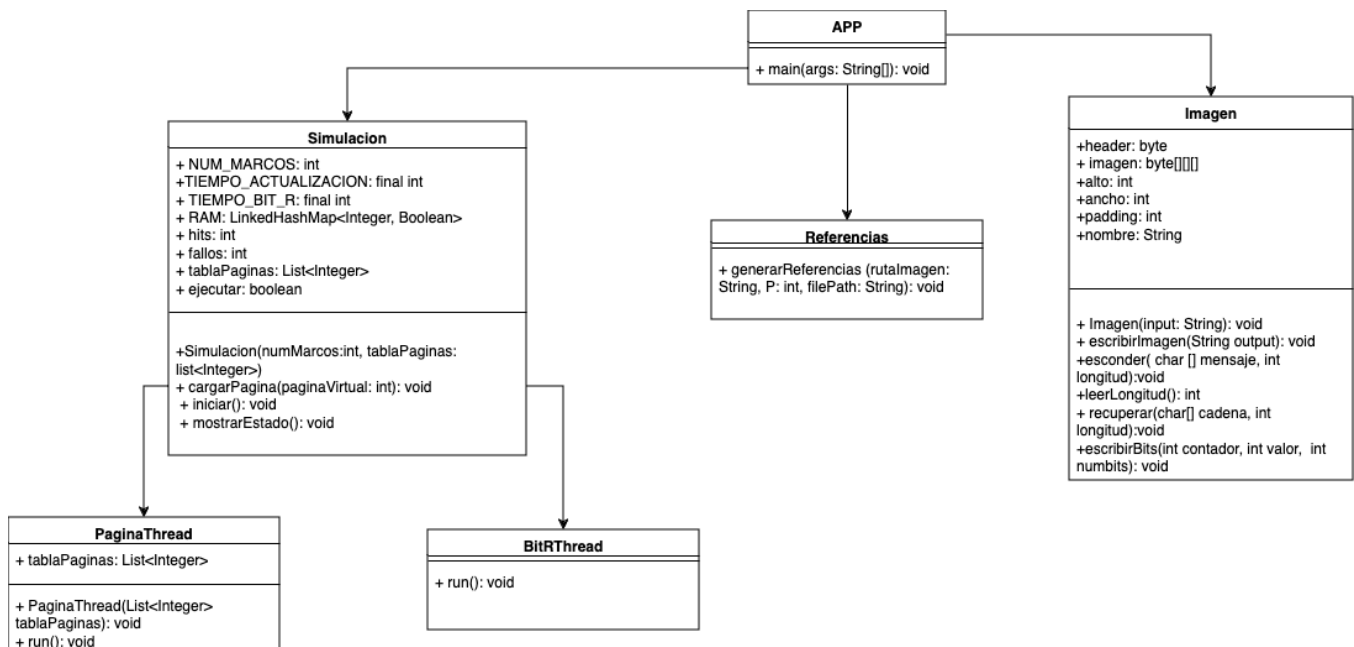
Cesar Avellaneda – 202214746

Juana Mejia – 202021512

Introducción

El objetivo de este proyecto es descubrir lo importante que es contar con una administración apropiada de la memoria al construir un prototipo del sistema de administración de memoria virtual que permita calcular el número de fallas de página e hits de los datos en RAM. Las páginas saldrán de las referencias sobre el proceso de recuperación de un mensaje escondido dentro de una imagen.

Diagrama UML



Clases Principales

1. **Imagen:** Esta clase es la que se encarga de manejar la lectura, modificación y escritura de imágenes en formato BMP, para poder esconder y recuperar los mensajes embebidos.

- **Métodos Clave:**

- **Imagen(String input):** Este método crea una matriz a partir de un archivo entregado
- **escribirBits(int contador, int valor, int numbits):** permite esconder los bits en la matriz de la imagen, es decir en los bits.
- **esconder(char[] mensaje, int longitud):** este método oculta el mensaje proporcionado por el usuario en la imagen.
- **escribirImagen(String output):** escribe la imagen en un nuevo archivo en formato BMP.

- Recuperar(char[] cadena, int longitud): este método permite recuperar el mensaje que ha sido escondido en la imagen
- 2. **Referencias:** Esta clase genera un archivo de referencias que muestran la forma en la que se escondió el mensaje en la imagen.
 - **Métodos Clave:**
 - generarReferencias(String rutaImagen, int P, String filePath): Genera el archivo de referencias con información sobre la imagen y la ubicación del mensaje escondido.
- 3. **Simulación:** Clase que simula la paginación de un archivo de referencias utilizando un número dado de marcos de página.
 - **Subclases:**
 - **PaginaThread:** es la encargada de recorrer las referencias una por una y hacer los cambios de página necesarios.
 - **BitRThread:** es la encargada de pasar todos los bits R a false cada 2 ms.
 - **Métodos Clave:**
 - run(): Método que ejecuta la simulación de la paginación.
- 4. **App:** Clase principal por la cual se interactúa con el usuario. Esta despliega un menú que permite al usuario seleccionar diferentes opciones como generar referencias, simular la paginación, esconder y revelar mensajes.

Descripción del algoritmo usado para generar las referencias de página (modo uno)

El algoritmo utilizado para generar las referencias de página funciona de la siguiente manera:

Este algoritmo recibe los parámetros rutaImagen (string) que es la ruta de la imagen en el computador, P (int) este es el tamaño de las páginas, filePath (String) que es el nombre que se debe poner al archivo resultante para las referencias generadas.

- El primer paso es cargar el archivo de la imagen y se leen las dimensiones (ancho y altura). Un array en 3D se usa para representar los píxeles de colores (R, G, B). También se inicializan las variables para calcular los números de páginas, el número total de bytes y la longitud del mensaje. El número de páginas se obtiene como la división de la longitud total de la imagen por el tamaño de página (P).
- El siguiente paso es la iteración para recorrer cada píxel que tiene la imagen y para cada color se calcula la página a la que pertenece y cuál es el desplazamiento que tiene dentro de esa página.
- Luego se escribe la referencia. Para cada píxel y cada color, el algoritmo escribe una referencia en el siguiente formato:

```
Imagen[i][j][k].color,pagina,desplazamiento,R
```

La R representa que esta referencia está relacionada con la lectura de datos de la imagen. Cuando los bits del mensaje empiezan a ser procesados, que sucede luego de la lectura del byte 16, se escribe la ubicación de los bits del mensaje en el formato:

```
Mensaje[posCaracter],pagMensaje,desplazamientoMensaje,W
```

La W representa que esta es una operación de modificación de data, en este caso se encuentra relacionada con el embedding del mensaje secreto.

- Luego se hace la calculación de las páginas. Esto se obtiene al dividir la posición del byte por el tamaño de la página ingresado por el usuario y el desplazamiento dentro de la página con el residuo esta división. Para el mensaje escondido se calcula la página y el desplazamiento basado en la posición del bit del mensaje secreto, esto permite que el mensaje escondido este distribuido en los pixeles de la imagen.
- El último paso de este algoritmo es generar el archivo resultante y se hace luego de que todas las referencias hayan sido generadas.

Descripción de las estructuras de datos usadas para simular el comportamiento del sistema de paginación y cómo usa dichas estructuras (cuándo se actualizan, con base en qué y en qué consiste la actualización.

La simulación consta de:

- Un array de referencias que representa la tabla de páginas con todas las referencias cargadas se pasa por parámetro para iniciar la simulación.
- Una LinkedList para la RAM, que tiene la página virtual y el bitR como booleano.

A medida que se recorren las referencias, se revisa si la página actual está en la RAM, si se encuentra marca un hit y asigna el bitR a true, de lo contrario marca un fallo y revisa si hay espacio en la RAM. Si se encuentra el espacio, asigna el marco a la RAM y lo pone la página en la tabla de páginas con bitR en true. Si no hay espacio, revisa los bits R de los marcos para saber cuál debe de remplazar, la elimina de la tabla de páginas, obtiene el marco donde está la página, la remplaza por la nueva e introduce la nueva página en la tabla de páginas.

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

La sincronización es necesaria en los siguientes métodos:

- CargarPagina(): hay que sincronizar este método debido a que en este se revisan los bitsR que cambia el BitRThread.
- En el método run () de BitRThread, se sincroniza la parte que hace los cambios a los valores de bitR en la RAM, para que no haga el cambio a medida que se estan revisando.

El método sleep() con un tiempo predeterminado implica que no hay que hacer notify() en ningún momento, además de que los threads terminan cuando se acaban las referencias y esto hace terminar al thread de bitsR

Resultados de las Pruebas Realizadas:

Para probar la funcionalidad de las diferentes opciones se hicieron las combinaciones de las siguientes variables:

```
imagenes=["Imagen A", "Imagen B"]
caracteres= ['100','1000','2000','4000','8000']
sizes=['512','1024','2048']
marcos=['4','8']
```

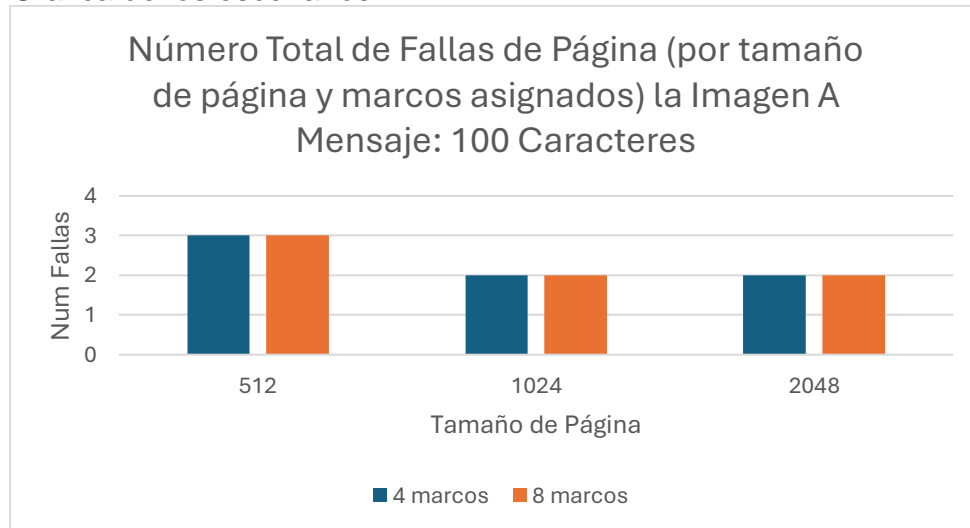
De esta forma se obtuvieron 60 escenarios diferentes. (2*5*3*2)

Escenario 1: Imagen A con 100 caracteres

Tabla con los datos recopilados:

Páginas de 512, pruebaA100.bmp: 100 Caracteres					
Marcos Asignados	Total Referencias	Hits	Fallas	Porcentaje de Hits	Porcentaje de Misses
4	1716	1713	3	99,825%	0,175%
8	1716	1713	3	99,825%	0,175%
Páginas de 1024, pruebaA100.bmp: 100 Caracteres					
Marcos Asignados	Total Referencias	Hits	Fallas	Porcentaje de Hits	Porcentaje de Misses
4	1716	1714	2	99,883%	0,117%
8	1716	1714	2	99,883%	0,117%
Páginas de 2048, pruebaA100.bmp: 100 Caracteres					
Marcos Asignados	Total Referencias	Hits	Fallas	Porcentaje de Hits	Porcentaje de Misses
4	1716	1714	2	99,883%	0,117%
8	1716	1714	2	99,883%	0,117%

Grafica de los escenarios:

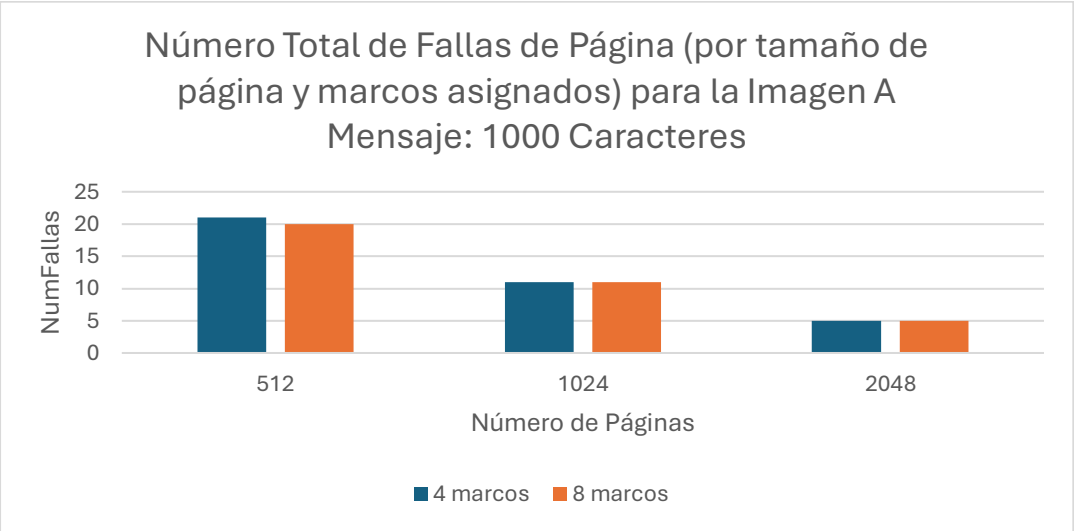


Escenario 2: Imagen A con 1000 caracteres

Tabla con los datos recopilados:

Páginas de 512, pruebaA1000.bmp: 1000 Caracteres					
Marcos Asignados	Total Referencias	Hits	Fallas	Porcentaje de Hits	Porcentaje de Misses
4	17016	16995	21	99,877%	0,123%
8	17016	16996	20	99,882%	0,118%
Páginas de 1024, pruebaA1000.bmp: 1000 Caracteres					
Marcos Asignados	Total Referencias	Hits	Fallas	Porcentaje de Hits	Porcentaje de Misses
4	17016	17005	11	99,935%	0,065%
8	17016	17005	11	99,935%	0,065%
Páginas de 2048, pruebaA1000.bmp: 1000 Caracteres					
Marcos Asignados	Total Referencias	Hits	Fallas	Porcentaje de Hits	Porcentaje de Misses
4	17016	17011	5	99,971%	0,029%
8	17016	17011	5	99,971%	0,029%

Grafica de los escenarios:

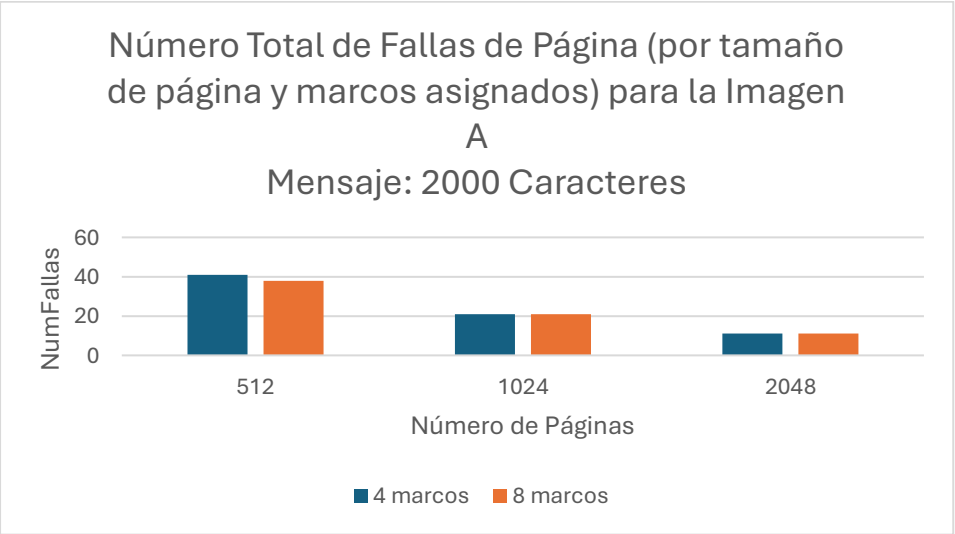


Escenario 3: Imagen A con 2000 caracteres

Tabla con los datos recopilados:

Páginas de 512, pruebaA2000.bmp: 2000 Caracteres					
Marcos Asignados	Total Referencias	Hits	Fallas	Porcentaje de Hits	Porcentaje de Misses
4	34016	33975	41	99,879%	0,121%
8	34016	33978	38	99,888%	0,112%
Páginas de 1024, pruebaA2000.bmp: 2000 Caracteres					
Marcos Asignados	Total Referencias	Hits	Fallas	Porcentaje de Hits	Porcentaje de Misses
4	34016	33995	21	99,938%	0,062%
8	34016	33995	21	99,938%	0,062%
Páginas de 2048, pruebaA2000.bmp: 2000 Caracteres					
Marcos Asignados	Total Referencias	Hits	Fallas	Porcentaje de Hits	Porcentaje de Misses
4	34016	34005	11	99,968%	0,032%
8	34016	34005	11	99,968%	0,032%

Grafica de los escenarios:

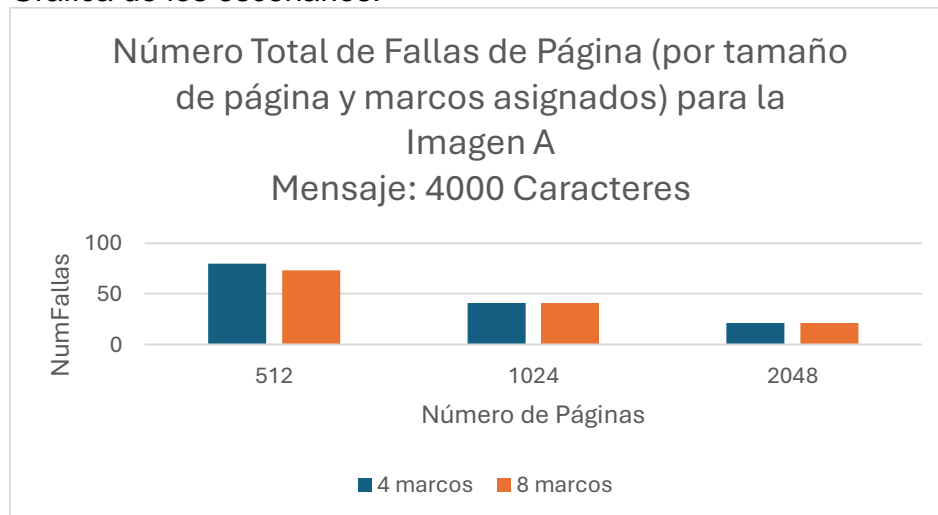


Escenario 4: Imagen A con 4000 caracteres

Tabla con los datos recopilados:

Páginas de 512, pruebaA4000.bmp: 4000 Caracteres					
Marcos Asignados	Total Referencias	Hits	Fallas	Porcentaje de Hits	Porcentaje de Misses
4	68033	67953	80	99,882%	0,118%
8	68033	67960	73	99,893%	0,107%
Páginas de 1024, pruebaA4000.bmp: 4000 Caracteres					
Marcos Asignados	Total Referencias	Hits	Fallas	Porcentaje de Hits	Porcentaje de Misses
4	68033	67992	41	99,940%	0,060%
8	68033	67992	41	99,940%	0,060%
Páginas de 2048, pruebaA4000.bmp: 4000 Caracteres					
Marcos Asignados	Total Referencias	Hits	Fallas	Porcentaje de Hits	Porcentaje de Misses
4	68033	68012	21	99,969%	0,031%
8	68033	68012	21	99,969%	0,031%

Grafica de los escenarios:

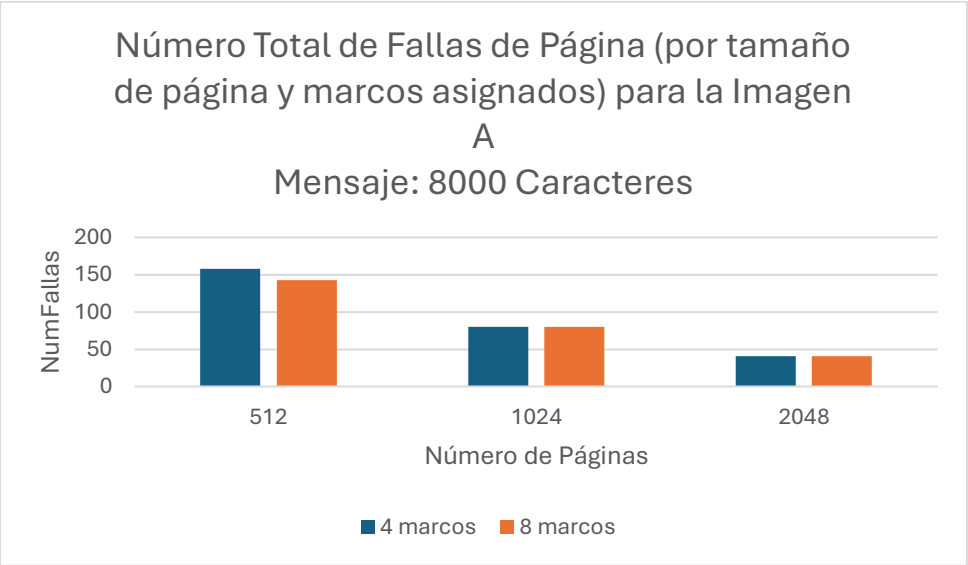


Escenario 5: Imagen A con 8000 caracteres

Tabla con los datos recopilados:

Páginas de 512, pruebaA8000.bmp: 8000 Caracteres					
Marcos Asignados	Total Referencias	Hits	Fallas	Porcentaje de Hits	Porcentaje de Misses
4	136016	135858	158	99,884%	0,116%
8	136016	135873	143	99,895%	0,105%
Páginas de 1024, pruebaA2000.bmp: 8000 Caracteres					
Marcos Asignados	Total Referencias	Hits	Fallas	Porcentaje de Hits	Porcentaje de Misses
4	136016	135936	80	99,941%	0,059%
8	136016	135936	80	99,941%	0,059%
Páginas de 2048, pruebaA2000.bmp: 8000 Caracteres					
Marcos Asignados	Total Referencias	Hits	Fallas	Porcentaje de Hits	Porcentaje de Misses
4	136016	135975	41	99,970%	0,030%
8	136016	135975	41	99,970%	0,030%

Grafica de los escenarios:

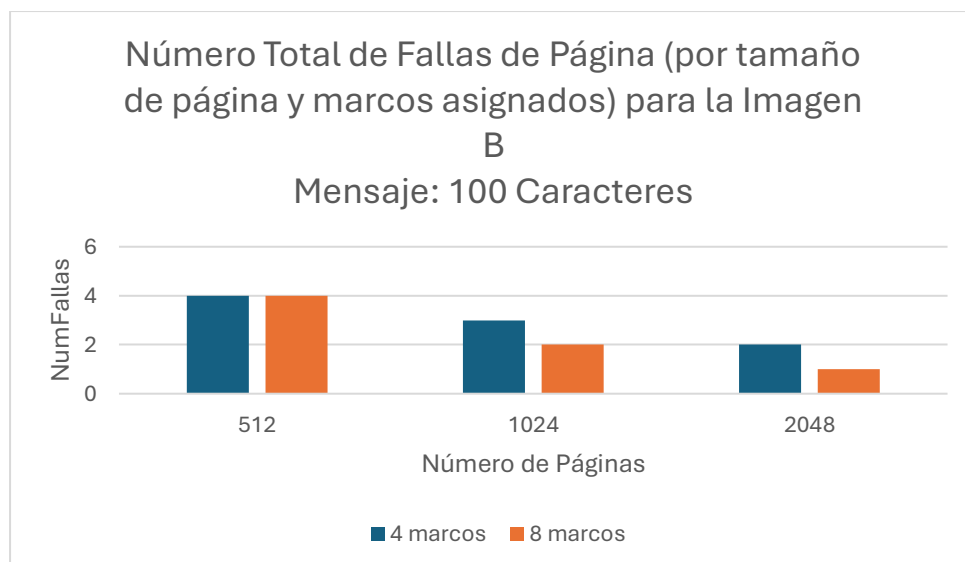


Escenario 6: Imagen B con 100 caracteres

Tabla con los datos recopilados:

Páginas de 512, pruebaB100.bmp: 100 Caracteres					
Marcos Asignados	Total Referencias	Hits	Fallas	Porcentaje de Hits	Porcentaje de Misses
4	1716	1712	4	99,767%	0,233%
8	1716	1712	4	99,767%	0,233%
Páginas de 1024, pruebaB100.bmp: 100 Caracteres					
Marcos Asignados	Total Referencias	Hits	Fallas	Porcentaje de Hits	Porcentaje de Misses
4	1716	1713	3	99,825%	0,175%
8	1716	1714	2	99,883%	0,117%
Páginas de 2048, pruebaB100.bmp: 100 Caracteres					
Marcos Asignados	Total Referencias	Hits	Fallas	Porcentaje de Hits	Porcentaje de Misses
4	1716	1714	2	99,883%	0,117%
8	1716	1715	1	99,942%	0,058%

Grafica de los escenarios:

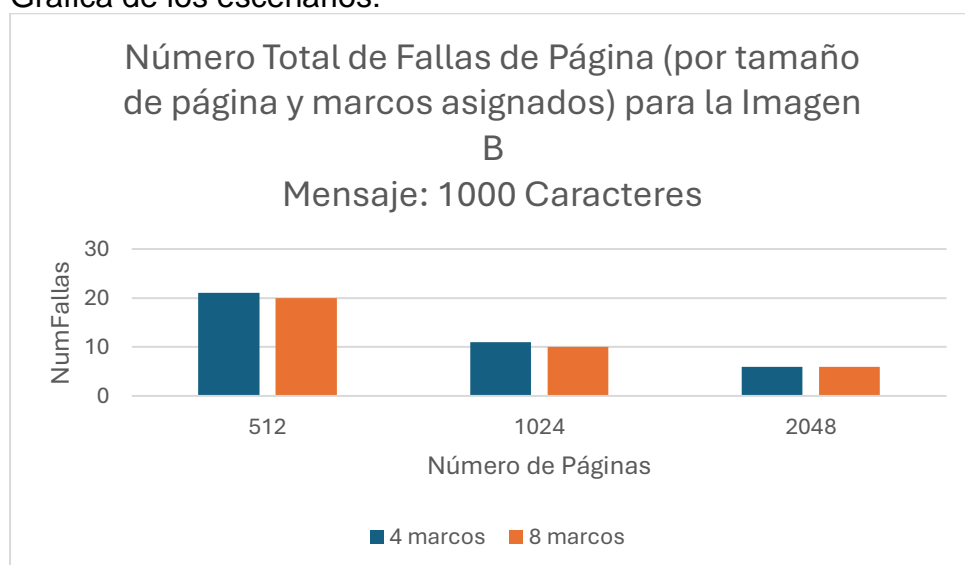


Escenario 7: Imagen B con 1000 caracteres

Tabla con los datos recopilados:

Páginas de 512, pruebaB1000.bmp: 1000 Caracteres					
Marcos Asignados	Total Referencias	Hits	Fallas	Porcentaje de Hits	Porcentaje de Misses
4	17016	16995	21	99,877%	0,123%
8	17016	16996	20	99,882%	0,118%
Páginas de 1024, pruebaB1000.bmp: 1000 Caracteres					
Marcos Asignados	Total Referencias	Hits	Fallas	Porcentaje de Hits	Porcentaje de Misses
4	17016	17005	11	99,935%	0,065%
8	17016	17006	10	99,941%	0,059%
Páginas de 2048, pruebaB1000.bmp: 1000 Caracteres					
Marcos Asignados	Total Referencias	Hits	Fallas	Porcentaje de Hits	Porcentaje de Misses
4	17016	17010	6	99,965%	0,035%
8	17016	17010	6	99,965%	0,035%

Grafica de los escenarios:

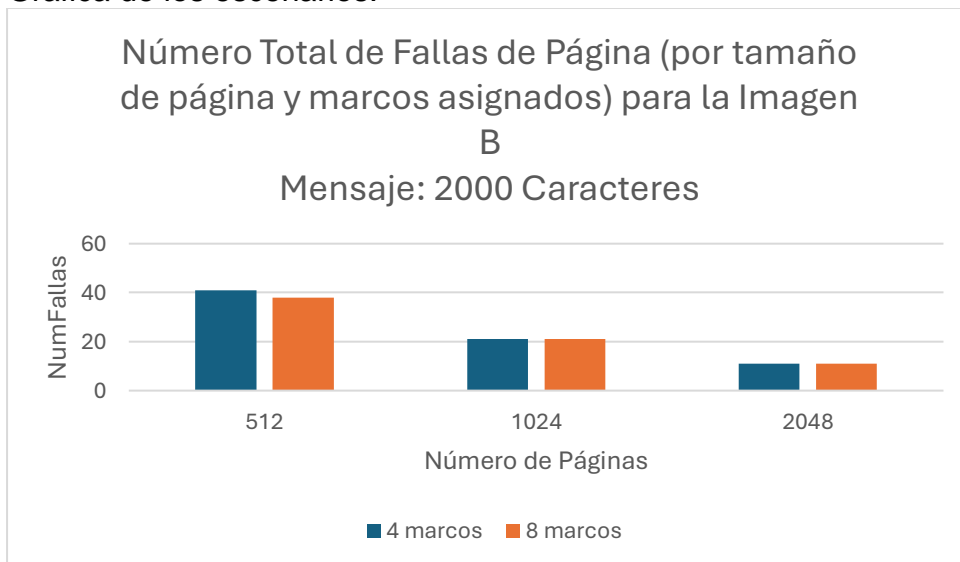


Escenario 8: Imagen B con 2000 caracteres

Tabla con los datos recopilados:

Páginas de 512, pruebaB2000.bmp: 2000 Caracteres					
Marcos Asignados	Total Referencias	Hits	Fallas	Porcentaje de Hits	Porcentaje de Misses
4	34016	33975	41	99,879%	0,121%
8	34016	33978	38	99,888%	0,112%
Páginas de 1024, pruebaB2000.bmp: 2000 Caracteres					
Marcos Asignados	Total Referencias	Hits	Fallas	Porcentaje de Hits	Porcentaje de Misses
4	34016	33995	21	99,938%	0,062%
8	34016	33995	21	99,938%	0,062%
Páginas de 2048, pruebaB2000.bmp: 2000 Caracteres					
Marcos Asignados	Total Referencias	Hits	Fallas	Porcentaje de Hits	Porcentaje de Misses
4	34016	34005	11	99,968%	0,032%
8	34016	34005	11	99,968%	0,032%

Grafica de los escenarios:

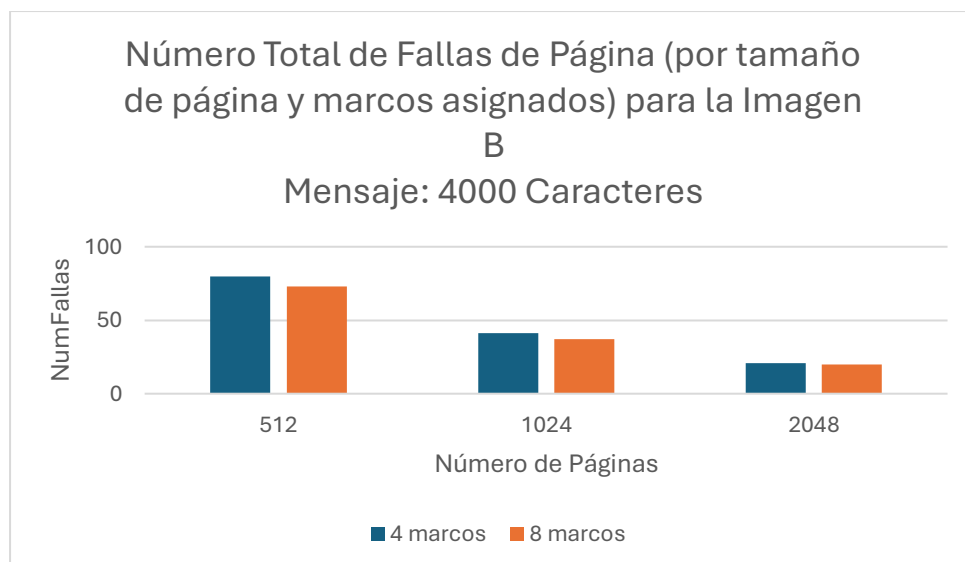


Escenario 9: Imagen B con 4000 caracteres

Tabla con los datos recopilados:

Páginas de 512, pruebaB4000.bmp: 4000 Caracteres					
Marcos Asignados	Total Referencias	Hits	Fallas	Porcentaje de Hits	Porcentaje de Misses
4	68033	67953	80	99,882%	0,118%
8	68033	67960	73	99,893%	0,107%
Páginas de 1024, pruebaB4000.bmp: 4000 Caracteres					
Marcos Asignados	Total Referencias	Hits	Fallas	Porcentaje de Hits	Porcentaje de Misses
4	68033	67992	41	99,940%	0,060%
8	68033	67996	37	99,946%	0,054%
Páginas de 2048, pruebaB4000.bmp: 4000 Caracteres					
Marcos Asignados	Total Referencias	Hits	Fallas	Porcentaje de Hits	Porcentaje de Misses
4	68033	68012	21	99,969%	0,031%
8	68033	68013	20	99,971%	0,029%

Grafica de los escenarios:

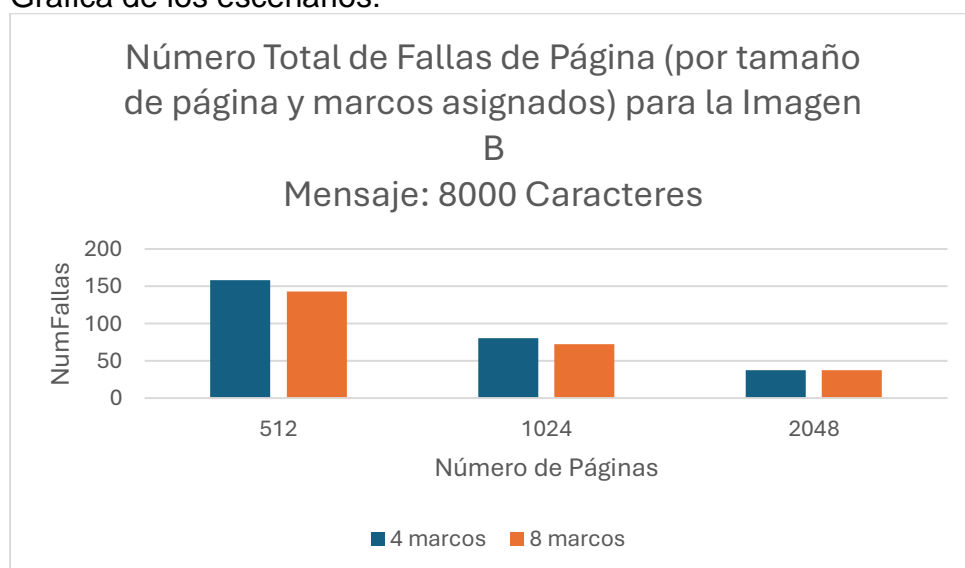


Escenario 10: Imagen B con 8000 caracteres

Tabla con los datos recopilados:

Páginas de 512, pruebaB8000.bmp: 8000 Caracteres					
Marcos Asignados	Total Referencias	Hits	Fallas	Porcentaje de Hits	Porcentaje de Misses
4	136016	135858	158	99,884%	0,116%
8	136016	135873	143	99,895%	0,105%
Páginas de 1024, pruebaB8000.bmp: 8000 Caracteres					
Marcos Asignados	Total Referencias	Hits	Fallas	Porcentaje de Hits	Porcentaje de Misses
4	136016	135936	80	99,941%	0,059%
8	136016	135944	72	99,947%	0,053%
Páginas de 2048, pruebaB8000.bmp: 8000 Caracteres					
Marcos Asignados	Total Referencias	Hits	Fallas	Porcentaje de Hits	Porcentaje de Misses
4	136016	135979	37	99,973%	0,027%
8	136016	135979	37	99,973%	0,027%

Grafica de los escenarios:



Interpretación de los resultados: ¿corresponden a los resultados que esperaba, con respecto al número de marcos asignados?

Al analizar los objetivos obtenidos con respecto a los diferentes casos planteados se observaron tendencias claras con respecto a lo siguiente:

- Cuando se tenían más marcos el número de fallos de página era menor y cuando se tenían menos marcos se observaba un incremento en los fallos de página.
Este era el comportamiento esperado debido a que más marcos permiten que el sistema tenga más espacio de memoria física para almacenar las páginas. Esto es debido a que son bloques donde se cargan las páginas desde la memoria virtual, por medio de estos es más fácil hacer uso eficiente de la memoria pues el sistema no necesita acceder al disco y la memoria física es más rápida que el disco.

Otros resultados:

- *Tamaño de páginas:* Pudimos observar que cuando se tenía un mayor tamaño para las páginas se lograban reducir los fallos. Esto corresponde a los resultados esperados ya que al utilizar un tamaño de páginas más pequeño, se requieren más páginas, esto implica que habrá una mayor probabilidad de fallos si no hay suficientes marcos para mantener las páginas en la memoria principal.

Otras configuraciones que le permitan entender cómo afecta la memoria virtual el desempeño del programa:

- *Algoritmos diferentes de reemplazo de página:* para la solución presentada se hace uso del algoritmo least recently used (LRU), se podría intentar cambiar por least frequently used (LFU) o first in first out (FIFO). Estas dos configuraciones nos podrían dejar ver la importancia de LRU y cómo este reduce la cantidad de misses que tiene el sistema.
- *Carga del Sistema:* se podría simular cómo afecta el rendimiento de la memoria virtual que haya diferentes cargas en el sistema, esto se puede hacer al simular la ejecución de diferentes programas simultáneamente.

¿Si la localidad del problema manejado fuera diferente cómo variarían los resultados? Explique su respuesta.(considere una localidad mayor y una localidad menor).

Debido a que la localidad afecta directamente el número de fallos de página, si esta fuera manejada diferente el rendimiento se vería afectado. Esto se debe a que si se tiene una localidad mayor se logra que el número de fallos de la página disminuya, mientras que con una localidad menor estos aumentarían.

Localidad mayor: Lo que se logra con esto es que el programa acceda a un conjunto pequeño de páginas en un periodo corto de tiempo muchas veces. Esto significa que las páginas que son cargadas a la memoria física permanecerán allí. El resultado es

que ocurrirán menos fallos de página ya que las páginas necesarias ya se encuentran en la memoria, adicionalmente se hace un uso más eficiente de la memoria ya que el sistema operativo puede predecir cuales son las páginas que debe utilizar frecuentemente y que se queden en los marcos de la memoria.

Localidad menor: Esta logra el efecto contrario debido a que con una localidad menor el programa está accediendo a una mayor cantidad de páginas con mayor aleatoriedad. Esto causa que el sistema operativo no pueda predecir cuales son las páginas que debe utilizar y significa que muchas páginas deben estar cargadas a la memoria física. El resultado es que se generen más fallos de página pues al probabilidad que una página necesaria no esté en la memoria incrementa.