

Caso 2 – Informe

Andres Botero Ruiz – 202223503

Sergio Alejandro Castaño Arcila – 202310390

Manolo Hernandez Rojas - 202224469

Tabla de Contenido

Contrato de Equipo.....	2
UML.....	6
Descripción del algoritmo usado para generar las referencias de página	7
Descripción de las estructuras de datos usadas para simular el comportamiento del sistema de paginación	9
Esquema de sincronización usado.....	13
Tabla con los datos recopilados.....	14
Gráficas que ilustren el comportamiento del sistema	22
Gráficas de tiempo	27
¿Corresponden a los resultados que esperaba, con respecto al número de marcos asignados?	31
¿Aplicar el filtro sobel representa un problema de localidad alta, media o baja?	31

Contrato de Equipo

Equipo #5

Miembros del equipo (nombre y correo electrónico):

1. Manolo Hernandez Rojas – m.hernandezr2@uniandes.edu.co
2. Andres Botero Ruiz - a.botelor@uniandes.edu.co
3. Sergio Alejandro Castaño Arcila - sa.castanoa1@uniandes.edu.co

Procedimientos del equipo

1. Día, hora y lugar de las reuniones periódicas del equipo:

R/ Martes, jueves y sábados a las 6 de la tarde de manera virtual hasta la entrega del proyecto.

2. Método de comunicación preferido (p. ej., correo electrónico, teléfono celular, Brightspace/Microsoft Teams, WhatsApp) para informarse mutuamente sobre reuniones de equipo, anuncios, actualizaciones, recordatorios, problemas:

R/ Nos comunicaremos por whatsapp.

3. Política de toma de decisiones (¿por consenso? ¿por mayoría de votos?):

R/ Sera por consenso, todos debemos estar de acuerdo en las decisiones.

4. Método para establecer y seguir las agendas de las reuniones (¿Quién establecerá cada agenda? ¿Cuándo? ¿Cómo se notificará/recordará a los miembros del equipo? ¿Quién será responsable de que el equipo siga la agenda durante una reunión de equipo? ¿Qué se hará para mantener al equipo en el buen camino durante una reunión?):

R/ Manolo establecerá la agenda de cada reunión, 1 hora antes de la reunión. Se notifocara por medio del grupo de whatsapp. El encargado de que se siga la agenda será Manolo. Se tomarán breves descansos para mantener la mente calmada.

5. Método de mantenimiento de registros (¿Quién será responsable de registrar y distribuir las actas? ¿Cómo y cuándo se distribuirán las actas? ¿Dónde se guardarán todas las agendas y actas?):

R/ Andres mantendrá el registro de las actas, se distribuirán a más tardar en la mañana del día siguiente. Se guardarán en un repositorio de github.

6. Cuidado de lo Social: ¿quién trae galletas? ¿cuándo nos podemos reunir que no sea para trabajar? ¿qué nivel de cercanía tendremos?

R/ Vamos a ir a almorzar una vez a la semana para charlar de otros temas ajenos al proyecto, para mantener un ambiente cálido.

Expectativas del equipo

Calidad de trabajo

1. Estándares del proyecto (¿Cuál es un nivel realista de calidad para presentaciones en equipo, escritura colaborativa, investigación individual, preparación de borradores, revisiones por pares, etc.):

R/ La calidad va a ser representativa por nuestro máximo esfuerzo. Siempre vamos a revisar los aportes de cada integrante del grupo, para así buscar mejoras.

2. Estrategias para cumplir con estos estándares:

R/ Revisión de pares después de cada aporte al repositorio, revisión periódica de la totalidad del código, entre otros.

Participación del equipo

1. Estrategias para asegurar la cooperación y la distribución equitativa de tareas:

R/ Cada uno se va a desempeñar en el campo que mejor se le da, de esta manera se repartirá de manera consciente el trabajo.

2. Estrategias para fomentar/incluir ideas de todos los miembros del equipo (mantenimiento del equipo):

R/ Cuando empiece la reunión y surjan problemas relacionados al caso, haremos una lluvia de ideas, para que cada uno aporte al trabajo. También debe haber al menos una idea implementada de cada integrante.

3. Estrategias para mantenerse en la tarea (mantenimiento de la tarea):

R/ Descansos paulatinos, hablar en el medio de como fue el día y ice breakers de tal tipo.

4. Preferencias de liderazgo (informal, formal, individual, compartido):

R/ Compartido: Cada uno tiene voz.

Responsabilidad personal

1. Asistencia, puntualidad y participación individual esperada en todas las reuniones del equipo:

R/ Se debe ser puntual para todas las reuniones, y en caso de no poder asistir o entrada tarde, se debe avisar.

2. Nivel esperado de responsabilidad para cumplir con las tareas del equipo, los plazos y los plazos:

R/ Se espera que si se pacta una fecha, se entregue ahí lo planeado. Sin embargo si se presentan problemas se debe comunicar al grupo para ayudar a resolverlo entre todos.

3. Nivel esperado de comunicación con otros miembros del equipo:

R/ Comunicación diaria.

4. Nivel esperado de compromiso con las decisiones y tareas del equipo.

R/ Un nivel alto de compromiso.

Consecuencias por no seguir los procedimientos y cumplir con las expectativas

1. Describa, como grupo, manejaría infracciones de cualquiera de las obligaciones de este acuerdo de equipo:

R/ El grupo hablará de porque se infringió, y se buscaran respuestas y soluciones.

2. Describa lo que hará su equipo si las infracciones continúan:

R/ El grupo se comunicará con la profesora o el monitor.

Acciones para agradecer y fortalecer al equipo

1. Describa, como grupo, que acciones realizaran para reconocer la calidad de los aportes y logros tanto individuales como del equipo:

R/ Se darán dulces cuando un integrante haga un aporte valioso.

2. Describa una forma de celebrar lograr las metas del equipo:

R/ También con dulces.

a. Participé 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.

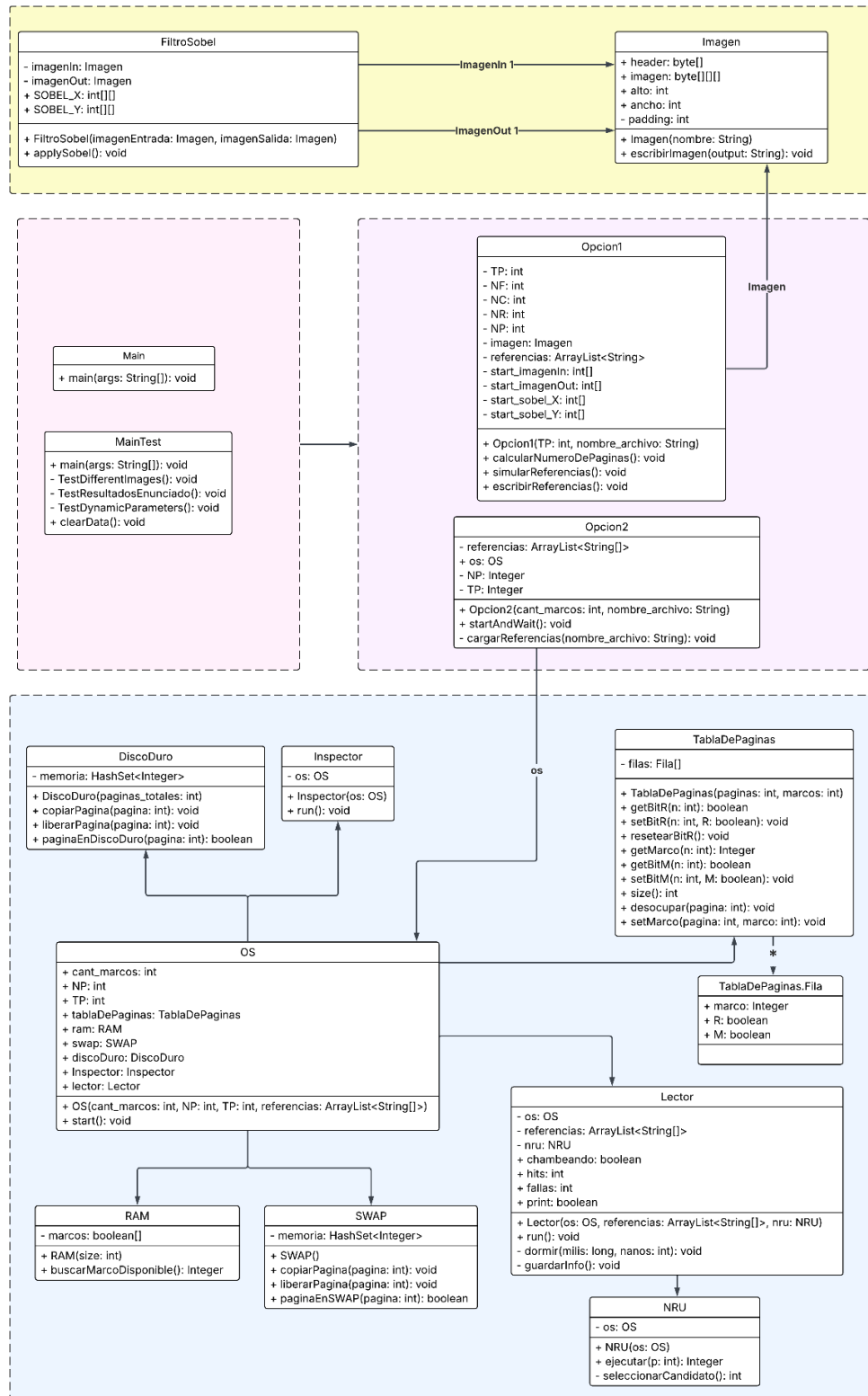
C. Entiendo que si no acato estos términos y condiciones, sufriré las consecuencias como se indica en este acuerdo.

Nombre, fecha: Manolo Hernandez Rojas - 26/03/25

Nombre, fecha: Andres Botero Ruiz - 26/03/25

Nombre, fecha: Sergio Alejandro Castaño Arcila - 26/03/25

UML



Descripción del algoritmo usado para generar las referencias de página

El algoritmo implementado en la clase Opcion1 genera referencias de página simulando el acceso a diferentes matrices en memoria durante la aplicación del filtro Sobel. A continuación, se describe el proceso paso a paso:

1. Cálculo del número de páginas virtuales (calcularNumeroDePaginas)

- Se calcula el tamaño en bytes requerido por las matrices involucradas:
 - imagenIn y imagenOut (matrices de la imagen de entrada y salida): $NF * NC * 3$ bytes (cada píxel tiene 3 componentes: rojo, verde y azul).
 - sobel_X y sobel_Y (matrices del filtro Sobel): 36 bytes cada una (9 elementos de 4 bytes).
 - Se suman los tamaños de todas las matrices para obtener los bytes totales.
 - Se calcula el número de páginas virtuales necesarias (NP) dividiendo los bytes totales entre el tamaño de página (TP) y redondeando hacia arriba.
 - Se determina el inicio de cada matriz en términos de página y desplazamiento dentro de la página, siguiendo el orden: imagenIn, sobel_X, sobel_Y, imagenOut.
-

2. Simulación de referencias de página (simularReferencias)

Este método simula el acceso a las matrices durante la aplicación del filtro Sobel. El algoritmo sigue estos pasos:

- **Iteración sobre la imagen de entrada (imagenIn):**
 - Se recorren los píxeles de la imagen (excluyendo los bordes) con dos bucles anidados para las filas (i) y columnas (j).
 - Para cada píxel, se aplica el filtro Sobel, lo que implica acceder a:
 1. **Matriz imagenIn:**
 - Se accede a los valores de los píxeles vecinos (en un área de 3x3 alrededor del píxel actual).
 - Por cada componente de color (rojo, verde, azul), se calcula la página y el desplazamiento correspondiente y se registra una referencia de lectura (R).

2. Matrices sobel_X y sobel_Y:

- Se accede a los valores de las máscaras Sobel para calcular los gradientes en las direcciones X e Y.
- Cada posición de las matrices Sobel se consulta tres veces (una por cada componente de color), y se registran referencias de lectura (R).

3. Matriz imagenOut:

- Se escriben los valores resultantes del filtro Sobel en la matriz de salida.
- Por cada componente de color (rojo, verde, azul), se calcula la página y el desplazamiento correspondiente y se registra una referencia de escritura (W).

- Cada referencia se almacena en el formato:

<NombreMatriz>[fila][columna].<componente>,<número de página>,<desplazamiento>,[R/W]

Donde R (lectura) o W (escritura).

3. Escritura de las referencias en un archivo (escribirReferencias)

- Se escriben las referencias generadas en un archivo de texto (referencias.txt), junto con información adicional como el tamaño de página (TP), dimensiones de la imagen (NF, NC), número de referencias (NR), y número de páginas virtuales (NP).
-

Resumen del algoritmo

El algoritmo simula el acceso a memoria durante la aplicación del filtro Sobel, generando referencias de página para cada operación de lectura/escritura en las matrices involucradas.

Descripción de las estructuras de datos usadas para simular el comportamiento del sistema de paginación

El sistema de paginación simulado utiliza varias estructuras de datos para modelar el comportamiento de la memoria virtual y su interacción con la memoria física. A continuación, se describen las principales estructuras de datos y cómo se usan:

1. Tabla de Páginas (TablaDePaginas)

- **Estructura:**

- Es un arreglo de objetos Fila, donde cada fila representa una página.
- Cada Fila contiene:
 - Integer marco: Indica el marco de memoria física asignado a la página (o null si no está en RAM).
 - boolean R: Bit de referencia, indica si la página ha sido referenciada recientemente.
 - boolean M: Bit de modificación, indica si la página ha sido modificada.

- **Uso:**

- La tabla de páginas se utiliza para rastrear el estado de cada página en el sistema.

- **Actualización:**

- **Bit R:** Se actualiza a true cuando una página es leída o escrita. Es reiniciado periódicamente por el hilo Inspector.
- **Bit M:** Se actualiza a true cuando una página es escrita.
- **Marco:** Se asigna o libera cuando una página es cargada o removida de la RAM.

- **Cuándo:**

- Durante las operaciones de lectura/escritura en la RAM.
- Cuando se ejecuta el algoritmo de reemplazo de páginas (NRU).

2. RAM (RAM)

- **Estructura:**
 - Un arreglo de booleanos (boolean[] marcos), donde cada índice representa un marco de memoria física.
 - true indica que el marco está ocupado, false que está libre.
- **Uso:**
 - Simula la memoria física disponible para almacenar páginas.
- **Actualización:**
 - Cuando se asigna un marco a una página, el marco correspondiente se marca como ocupado (true).
 - Cuando se libera un marco, se marca como libre (false).
- **Cuándo:**
 - Durante la asignación de un marco a una página o cuando se libera un marco por el algoritmo NRU.

3. Disco Duro (DiscoDuro)

- **Estructura:**
 - Un HashSet<Integer> que almacena los índices de las páginas que están en el disco duro.
- **Uso:**
 - Representa las páginas que no están en RAM ni en SWAP, pero están disponibles en el disco duro.
- **Actualización:**
 - Se agrega una página al disco duro cuando es cargada inicialmente.
 - Se elimina una página del disco duro cuando es cargada en RAM.
- **Cuándo:**

- Durante la carga inicial de las páginas o cuando una página es cargada en RAM.
-

4. SWAP (SWAP)

- **Estructura:**
 - Un HashSet<Integer> que almacena los índices de las páginas que han sido modificadas y están en la memoria SWAP.
 - **Uso:**
 - Representa las páginas que han sido removidas de la RAM, pero necesitan ser preservadas debido a modificaciones.
 - **Actualización:**
 - Se agrega una página al SWAP cuando es removida de la RAM y tiene el bit M activado.
 - Se elimina una página del SWAP cuando es cargada nuevamente en RAM.
 - **Cuándo:**
 - Durante el reemplazo de páginas por el algoritmo NRU.
-

5. Referencias (ArrayList<String[]>)

- **Estructura:**
 - Una lista de arreglos de cadenas (String[]), donde cada entrada representa una referencia a una página.
 - Cada referencia contiene:
 - Índice de la página.
 - Desplazamiento dentro de la página.
 - Acción (R para lectura o W para escritura).
- **Uso:**
 - Simula las referencias realizadas por los procesos.

- **Actualización:**
 - No se actualiza directamente, pero se procesa secuencialmente por el hilo Lector.
 - **Cuándo:**
 - Durante la simulación de las referencias.
-

6. Algoritmo NRU (NRU)

- **Estructura:**
 - Clasifica las páginas en cuatro clases basadas en los bits R y M:
 - Clase 0: R = false, M = false.
 - Clase 1: R = false, M = true.
 - Clase 2: R = true, M = false.
 - Clase 3: R = true, M = true.
 - **Uso:**
 - Selecciona una página candidata para ser removida de la RAM.
 - **Actualización:**
 - Libera el marco de la página seleccionada y, si es necesario, la copia al SWAP.
 - **Cuándo:**
 - Cuando la RAM está llena y se necesita espacio para cargar una nueva página.
-

7. Hilos (Lector y Inspector)

- **Lector:**
 - Procesa las referencias y actualiza los bits R y M en la tabla de páginas.
 - Invoca el algoritmo NRU cuando es necesario liberar espacio en la RAM.
- **Inspector:**

- Resetea periódicamente los bits R de todas las páginas en la tabla de páginas.
-

Resumen del Flujo de Actualización:

1. Lectura/Escritura:

- El hilo Lector procesa una referencia.
- Si la página no está en RAM, se verifica si está en SWAP o disco duro.
- Si la RAM está llena, se ejecuta el algoritmo NRU para liberar espacio.
- Se actualizan los bits R y M según la acción (R o W).

2. Reemplazo de Páginas:

- El algoritmo NRU selecciona una página para ser removida de la RAM.
- Si la página fue modificada, se copia al SWAP.

3. Reseteo de Bits R:

- El hilo Inspector resetea periódicamente los bits R de todas las páginas.

Estas estructuras y su interacción permiten simular el comportamiento de un sistema de paginación, incluyendo la gestión de memoria física, el reemplazo de páginas y el manejo de referencias.

Esquema de sincronización usado

Los únicos métodos que necesitan sincronización son en los que se consultan o escriben los bits R, ya que queremos evitar que el algoritmo NRU clasifique las páginas incorrectamente al consultar los bits R justo cuando el segundo thread (el inspector) está reiniciando los bits. Es por eso que en la clase `TablaDePaginas.java` los únicos métodos sincronizados son: `getBitR()`, `setBitR()`, `resetearBitR()`, `getMarco()`

Así aseguramos consistencia en las consultas.

Tabla con los datos recopilados

caso2-parrotspeq.bmp

Página de 256B, caso2-parrotspeq.bmp		
marcos_asignados	numero_hits	numero_fallas
1	584962	171794
2	598904	157852
3	722456	34300
4	735421	21335
5	749114	7642
6	750738	6018
7	755791	965
8	756536	220



Página de 512B, caso2-parrotspeq.bmp		
marcos_asignados	numero_hits	numero_fallas
1	585326	171430
2	599135	157621
3	727734	29022
4	745708	11048
5	756346	410
6	756646	110
7	756646	110
8	756646	110

Pagina de 1024B, caso2-parrotspeq.bmp		
marcos_asignados	numero_hits	numero_fallas
1	586901	169855
2	598161	158595
3	738213	18543
4	756591	165
5	756701	55
6	756701	55
7	756701	55
8	756701	55

Pagina de 2048B, caso2-parrotspeq.bmp		
marcos_asignados	numero_hits	numero_fallas
1	592436	164320
2	603553	153203
3	740892	15864
4	756728	28
5	756728	28
6	756728	28
7	756728	28
8	756728	28

Pagina de 4096B, caso2-parrotspeq.bmp		
marcos_asignados	numero_hits	numero_fallas
1	603883	152873
2	615049	141707
3	751913	4843
4	756742	14
5	756742	14
6	756742	14
7	756742	14
8	756742	14

cute_cat.bmp

Pagina de 256B, cute_cat.bmp		
marcos_asignados	numero_hits	numero_fallas
1	654940	218996
2	670745	203191
3	691160	182776
4	834337	39599
5	849497	24439
6	863833	10103
7	870759	3177
8	873361	575



Pagina de 512B, cute_cat.bmp		
marcos_asignados	numero_hits	numero_fallas
1	676222	197714
2	691382	182554
3	844823	29113
4	864718	9218
5	871552	2384
6	873808	128
7	873809	127
8	873809	127

Pagina de 1024B, cute_cat.bmp		
marcos_asignados	numero_hits	numero_fallas
1	678322	195614
2	691716	182220
3	855584	18352
4	872708	1228
5	873872	64
6	873872	64
7	873872	64
8	873872	64

Pagina de 2048B, cute_cat.bmp		
marcos_asignados	numero_hits	numero_fallas
1	684134	189802
2	696932	177004
3	864069	9867
4	873356	580
5	873904	32
6	873904	32
7	873904	32
8	873904	32

Pagina de 4096B, cute_cat.bmp		
marcos_asignados	numero_hits	numero_fallas
1	695550	178386
2	708532	165404
3	867982	5954
4	873660	276
5	873920	16
6	873920	16
7	873920	16
8	873920	16

kirby.bmp

Pagina de 256B, kirby.bmp		
marcos_asignados	numero_hits	numero_fallas
1	413807	121273
2	423112	111968
3	508193	26887
4	520067	15013
5	528128	6952
6	533426	1654
7	534915	165
8	534923	157



Pagina de 512B, kirby.bmp		
marcos_asignados	numero_hits	numero_fallas
1	414056	121024
2	422429	112651
3	516200	18880
4	530025	5055
5	535000	80
6	535001	79
7	535001	79
8	535001	79

Pagina de 1024B, kirby.bmp		
marcos_asignados	numero_hits	numero_fallas
1	415917	119163
2	423708	111372
3	521790	13290
4	535040	40
5	535040	40
6	535040	40
7	535040	40
8	535040	40

Pagina de 2048B, kirby.bmp		
marcos_asignados	numero_hits	numero_fallas
1	421689	113391
2	429222	105858
3	528199	6881
4	535060	20
5	535060	20
6	535060	20
7	535060	20
8	535060	20

Pagina de 4096B, kirby.bmp		
marcos_asignados	numero_hits	numero_fallas
1	433085	101995
2	441095	93985
3	531855	3225
4	535070	10
5	535070	10
6	535070	10
7	535070	10
8	535070	10

pikachu.bmp

Pagina de 256B, pikachu.bmp		
marcos_asignados	numero_hits	numero_fallas
1	549776	161200
2	563614	147362
3	678127	32849
4	691127	19849
5	703617	7359
6	709524	1452
7	710769	207
8	710769	207



Pagina de 512B, pikachu.bmp		
marcos_asignados	numero_hits	numero_fallas
1	550826	160150
2	562559	148417
3	686211	24765
4	704989	5987
5	710158	818
6	710872	104
7	710872	104
8	710872	104

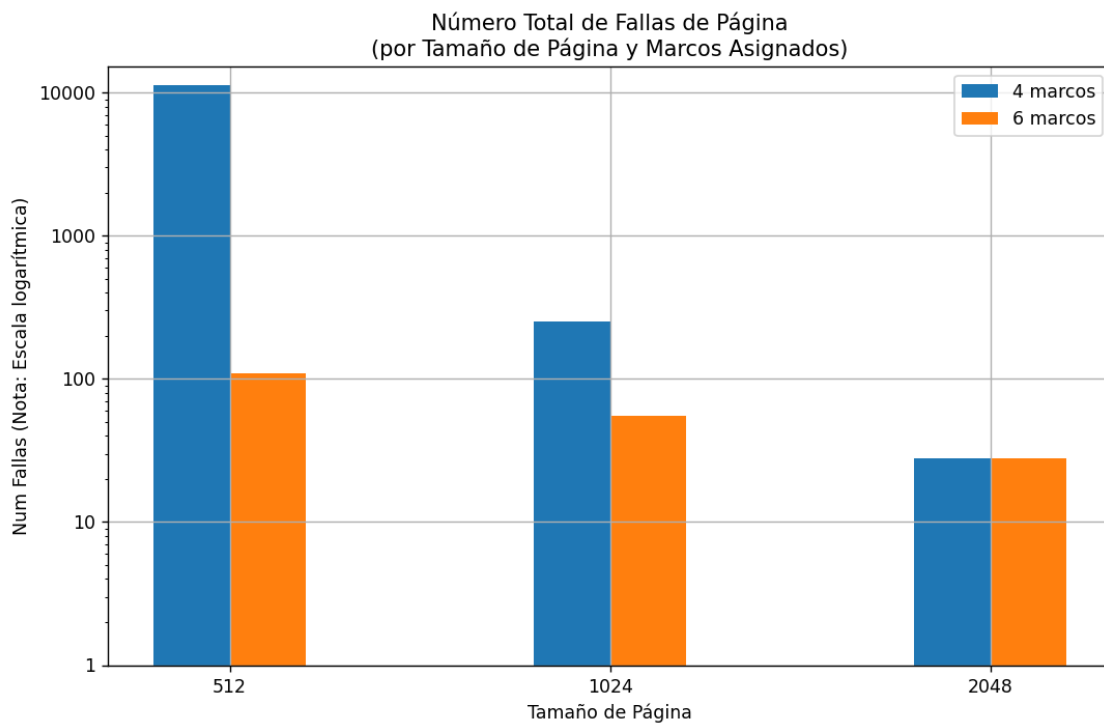
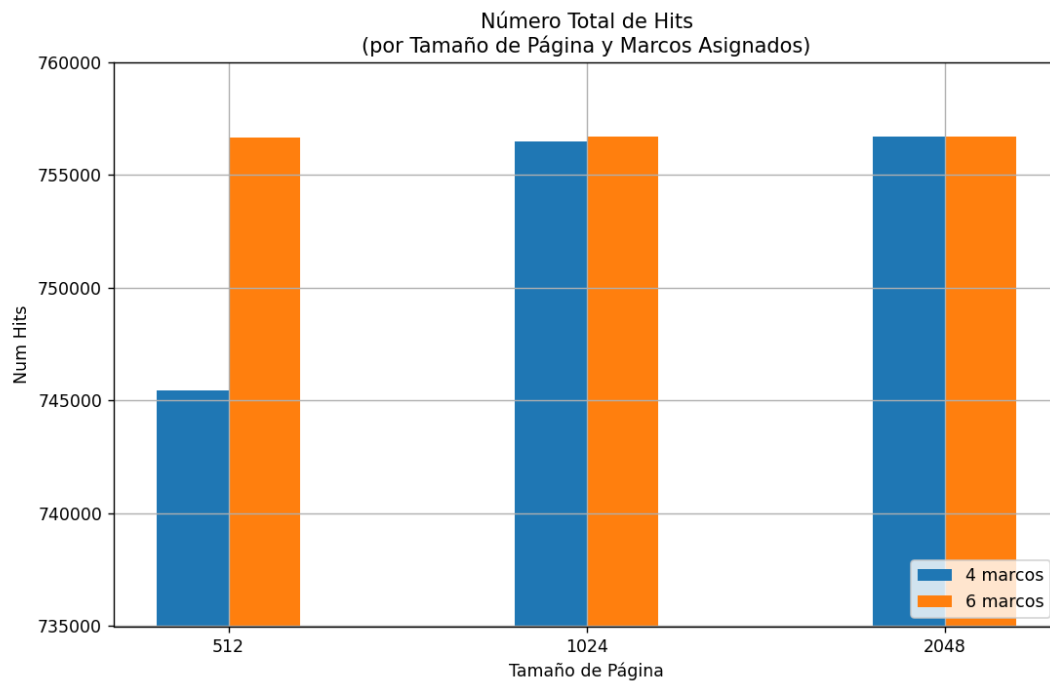
Pagina de 1024B, pikachu.bmp		
marcos_asignados	numero_hits	numero_fallas
1	553551	157425
2	564132	146844
3	696863	14113
4	708934	2042
5	710922	54
6	710922	54
7	710922	54
8	710922	54

Pagina de 2048B, pikachu.bmp		
marcos_asignados	numero_hits	numero_fallas
1	559337	151639
2	569525	141451
3	702447	8529
4	709980	996
5	710950	26
6	710950	26
7	710950	26
8	710950	26

Pagina de 4096B, pikachu.bmp		
marcos_asignados	numero_hits	numero_fallas
1	559968	151008
2	579790	131186
3	704113	6863
4	710963	13
5	710963	13
6	710963	13
7	710963	13
8	710963	13

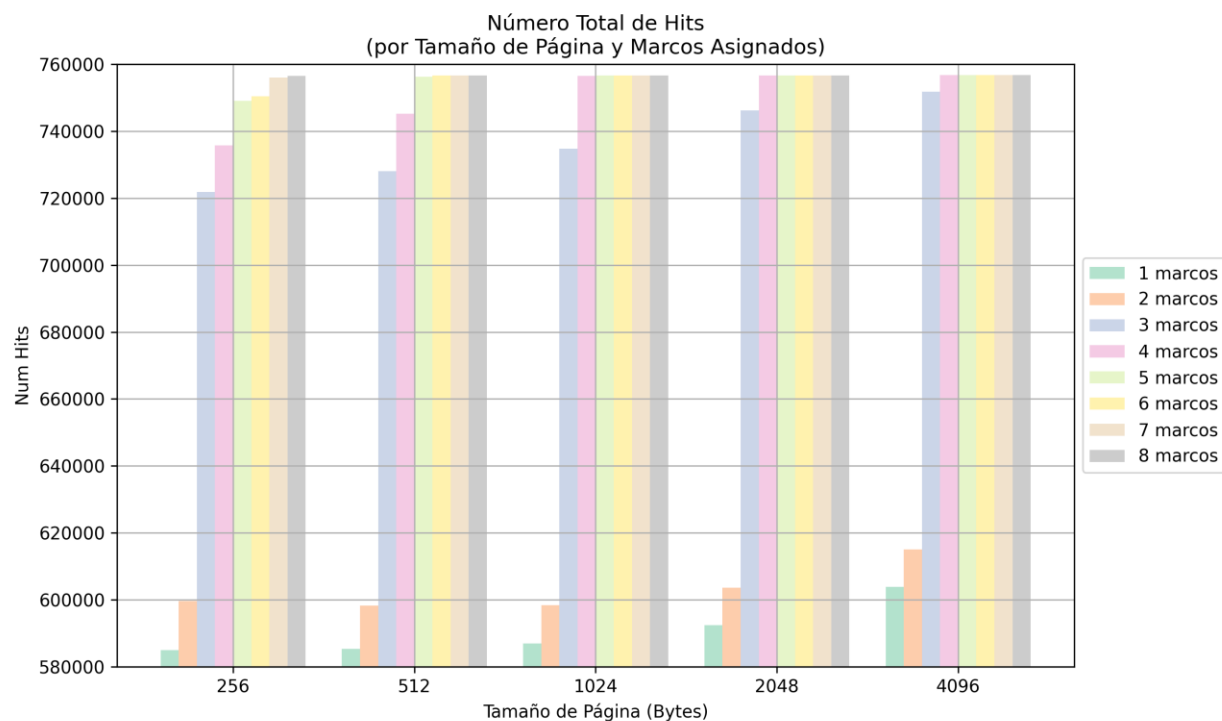
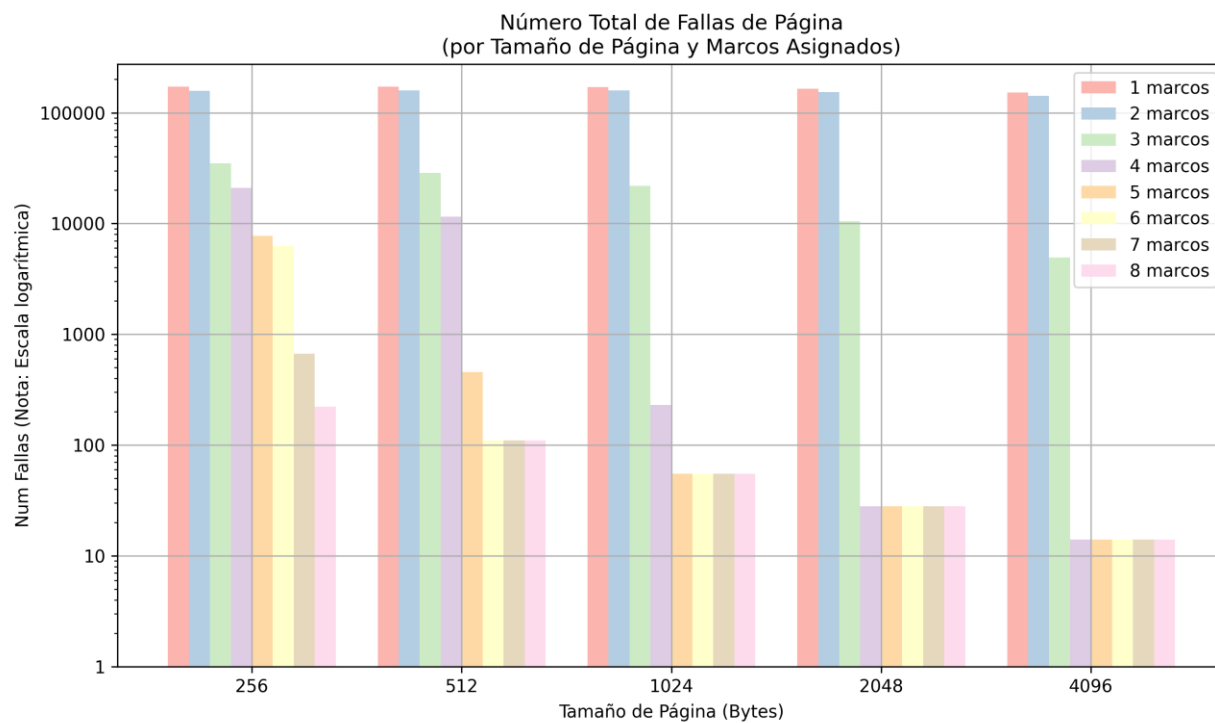
Gráficas que ilustren el comportamiento del sistema

Replicando las gráficas del enunciado con la imagen **caso2-parrotspeq.bmp**



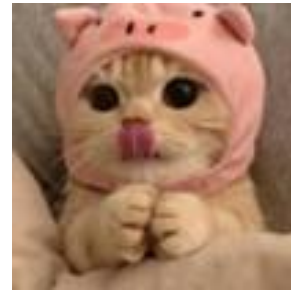
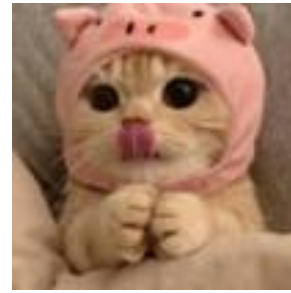
Considerando más escenarios para la imagen **caso2-parrotspeq.bmp**

Imagen de 119 x 79 pixeles



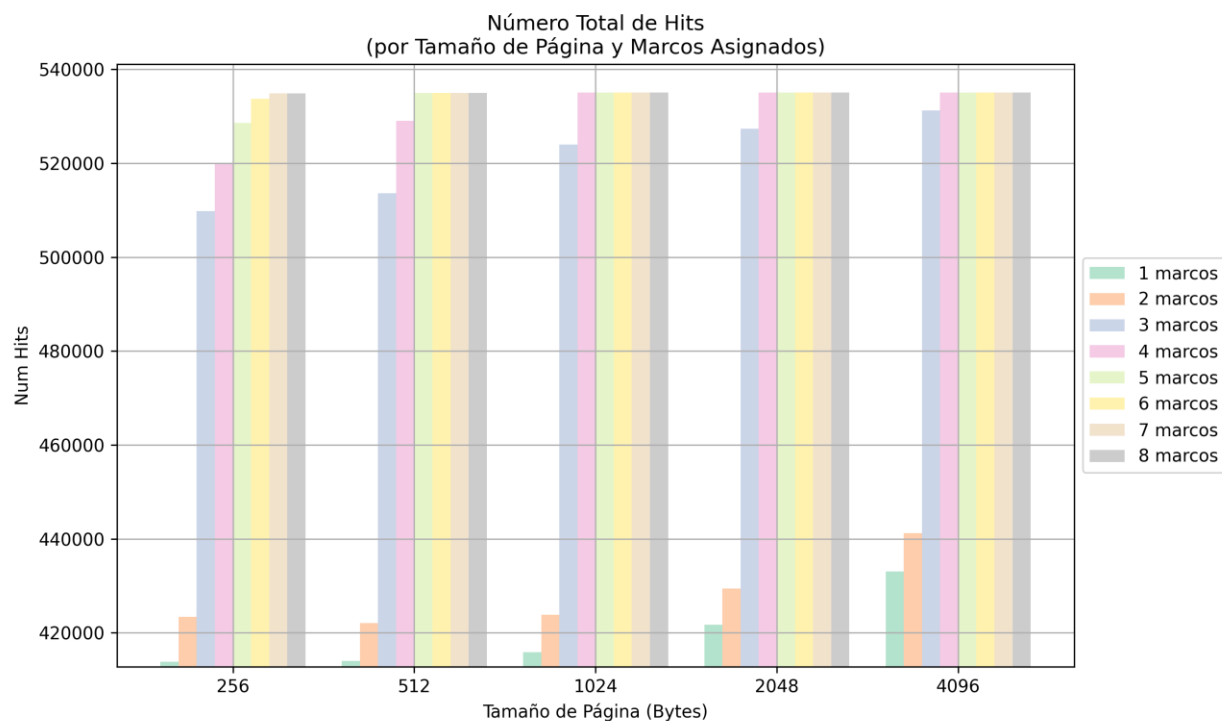
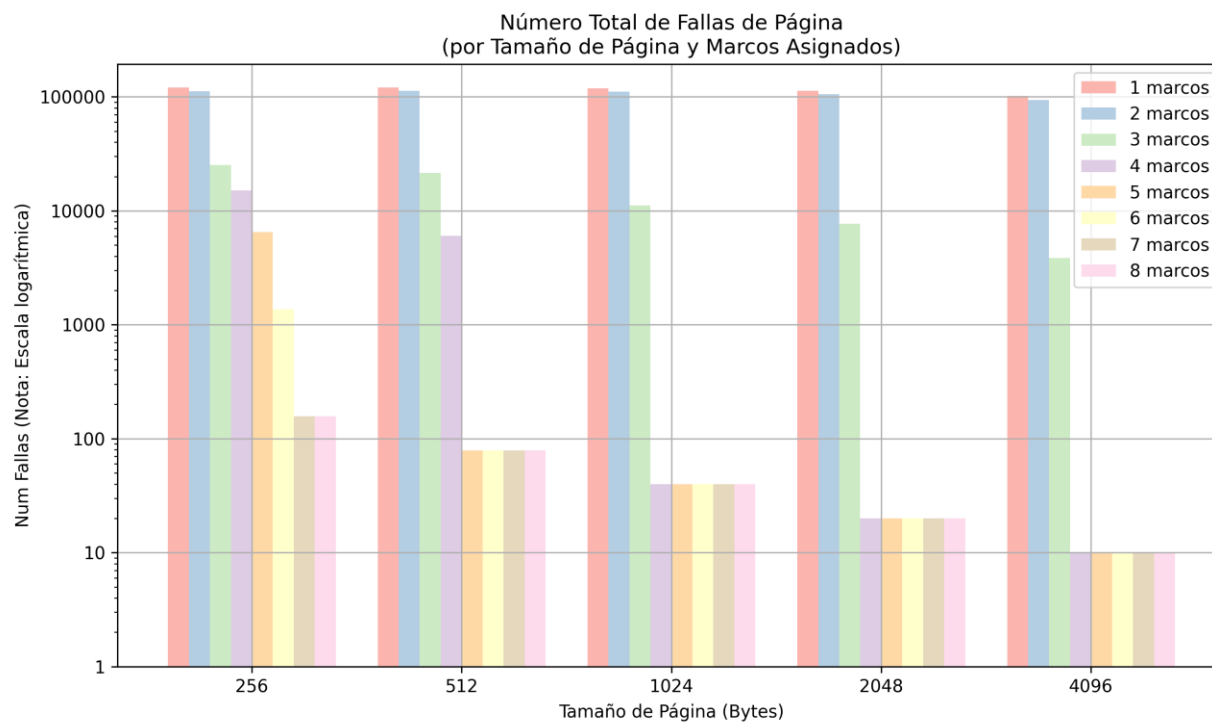
A small orange cat is wearing a pink pig mask. The cat is looking directly at the camera with its tongue sticking out. The mask has small ears and a snout. The cat is sitting on a light-colored surface.

A small orange cat is wearing a pink pig mask. The cat is looking directly at the camera with its tongue sticking out. The mask has small ears and a snout. The cat is sitting on a light-colored surface.



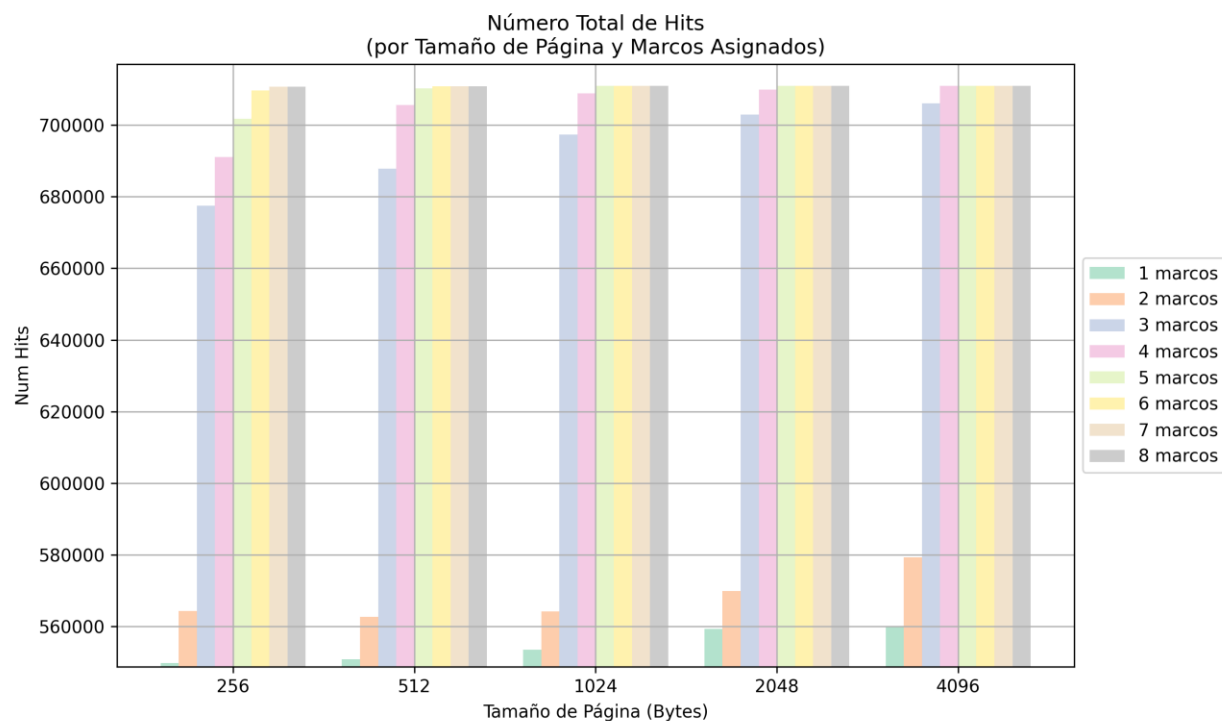
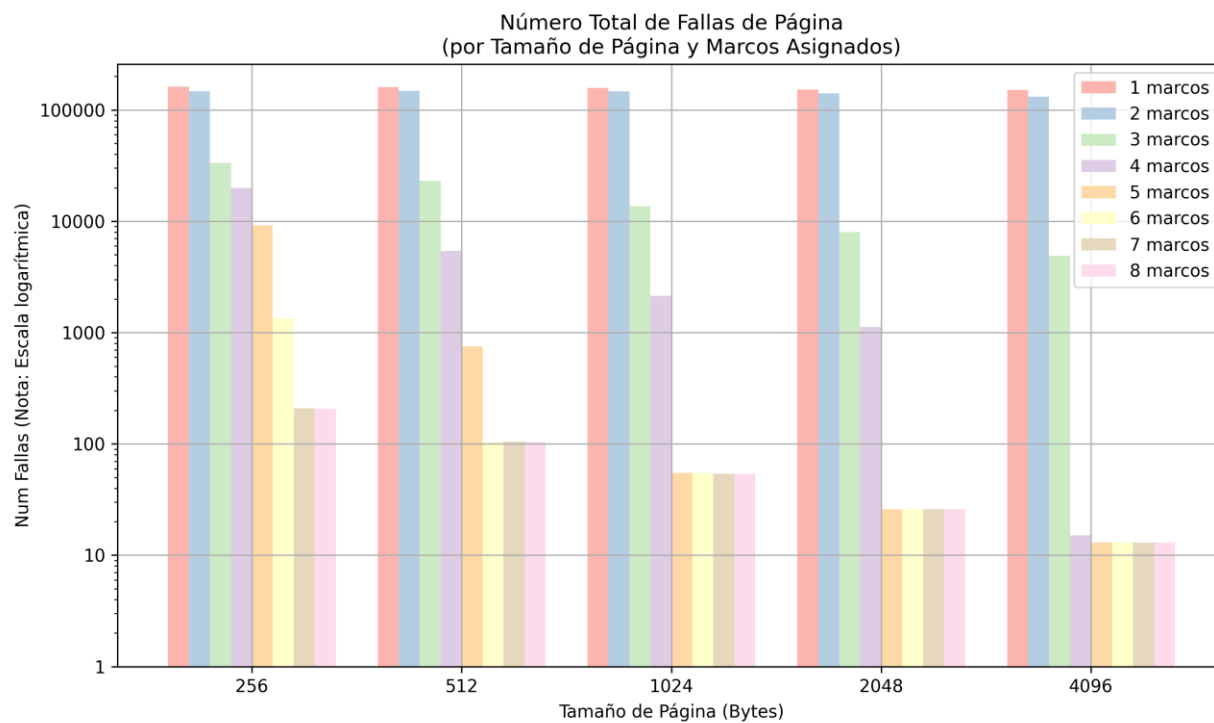
Probando más escenarios para la imagen **kirby.bmp**

Imagen de 100 x 67 pixeles



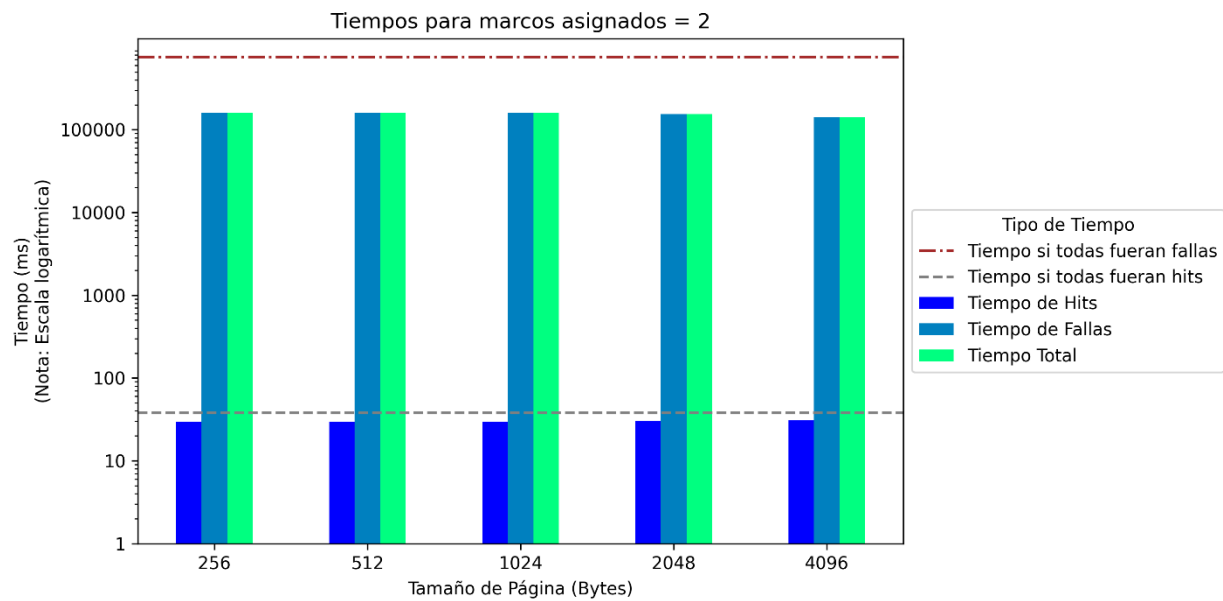
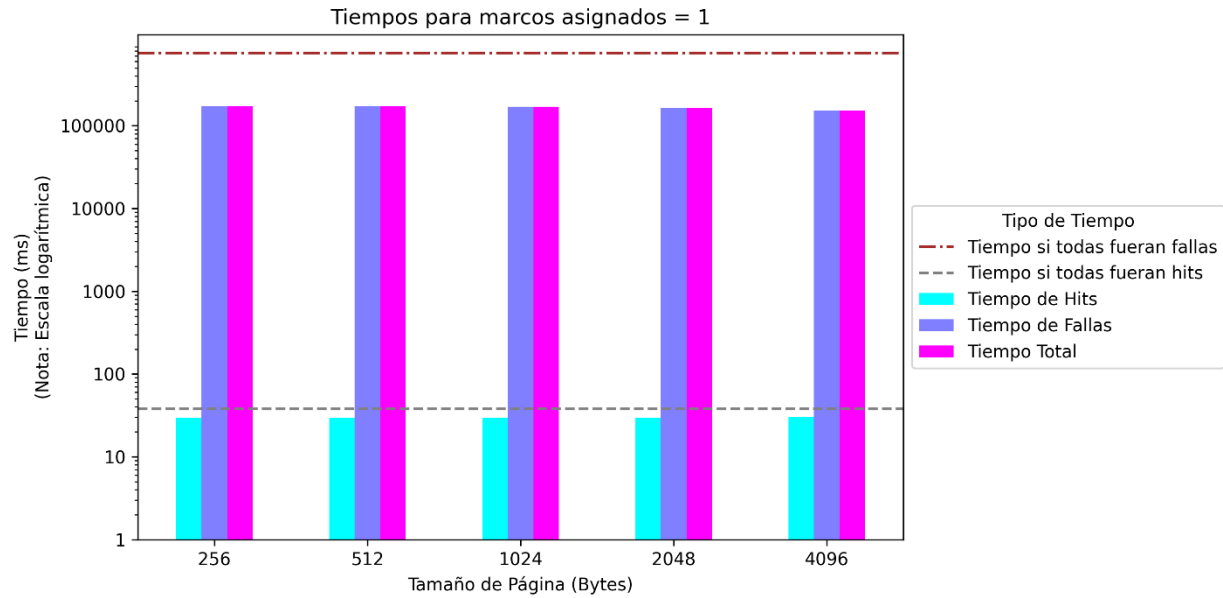
Probando más escenarios para la imagen **pikachu.bmp**

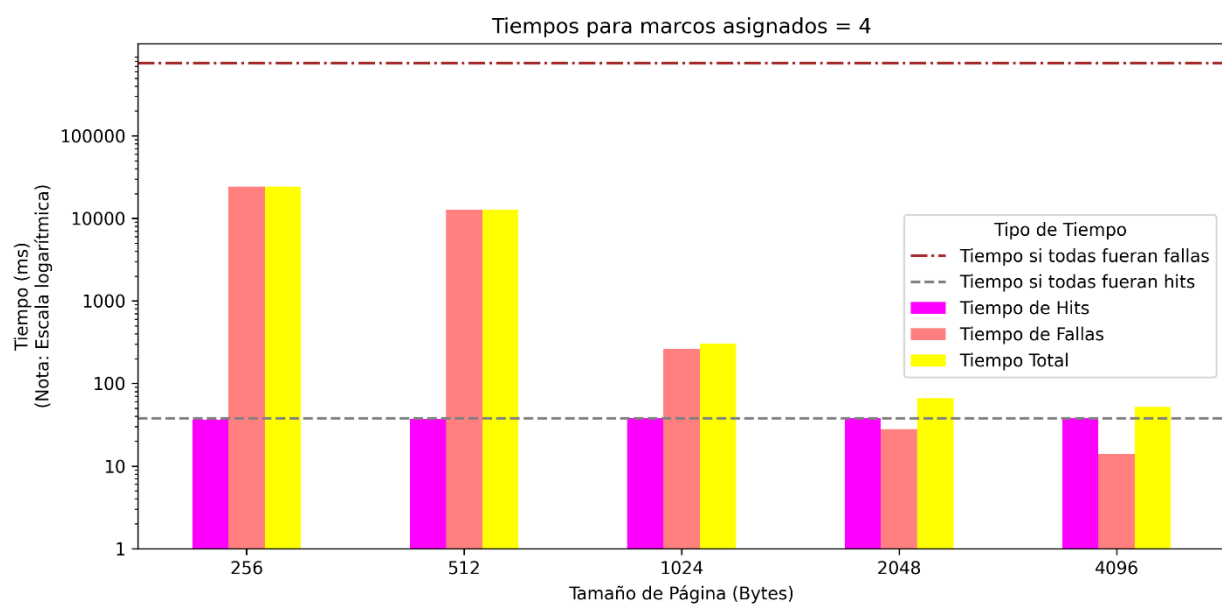
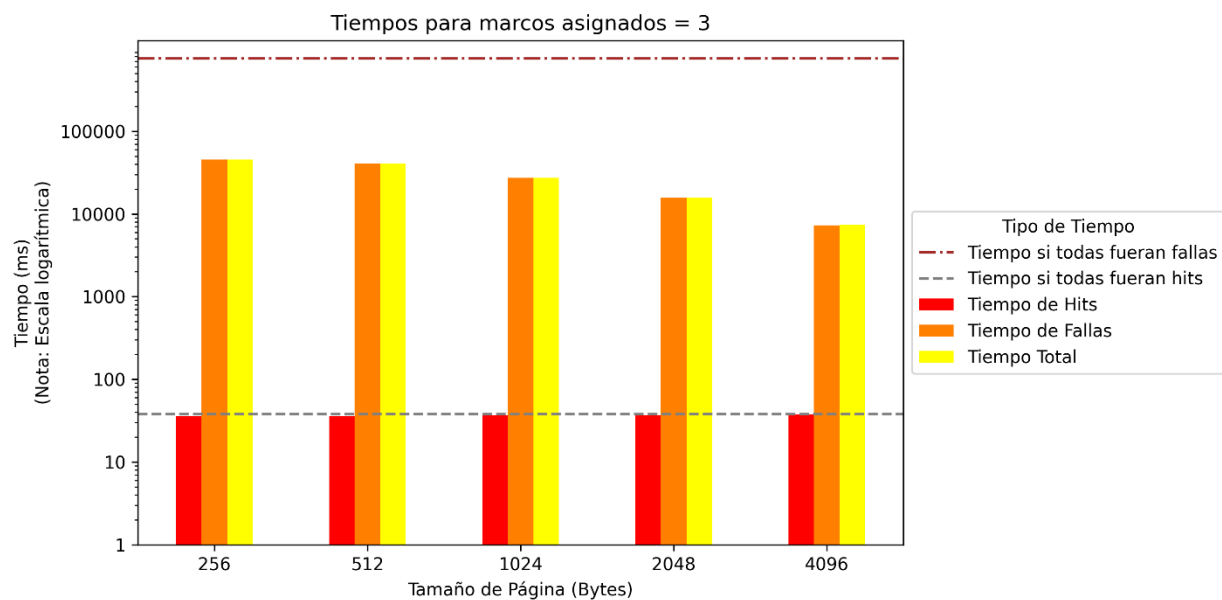
Imagen de 94 x 94 pixeles

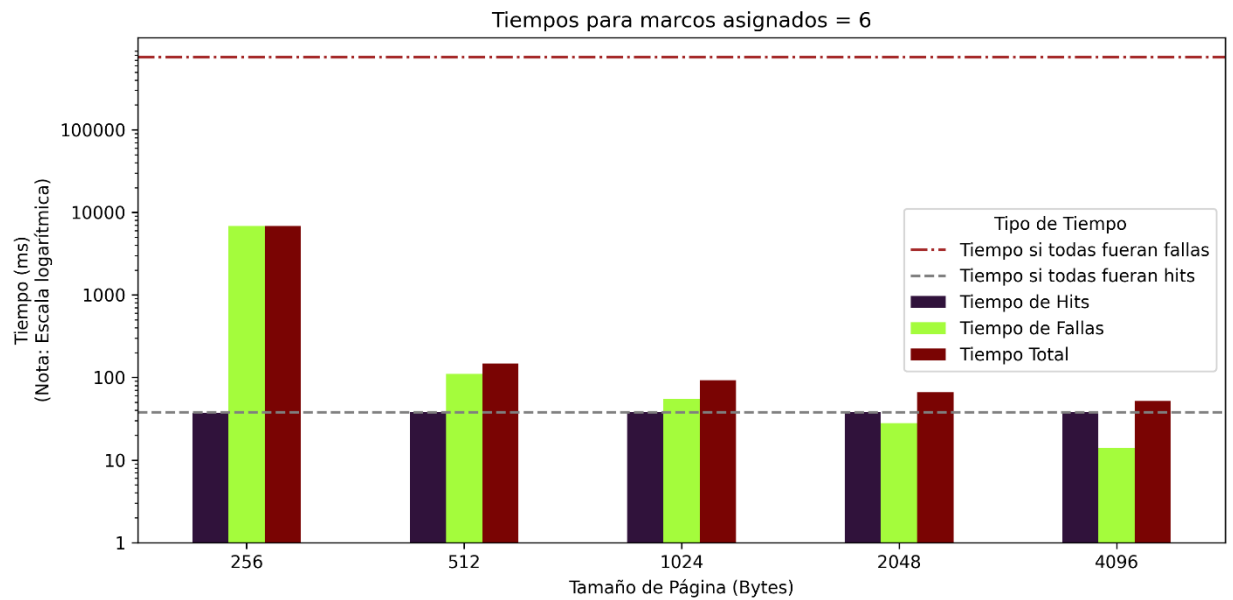
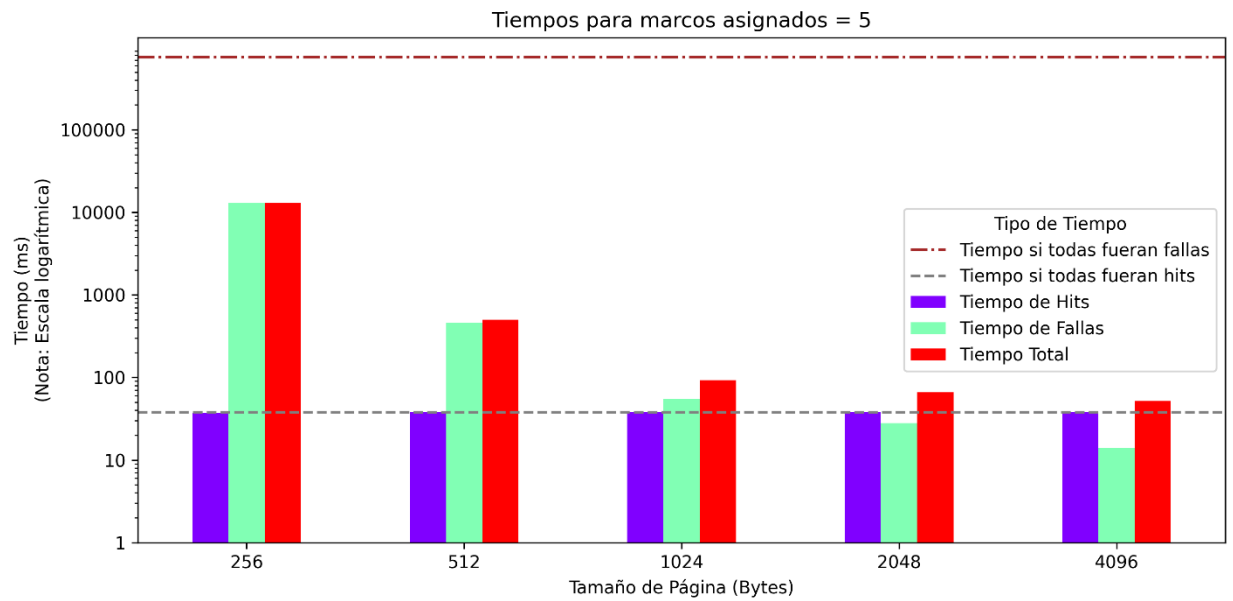


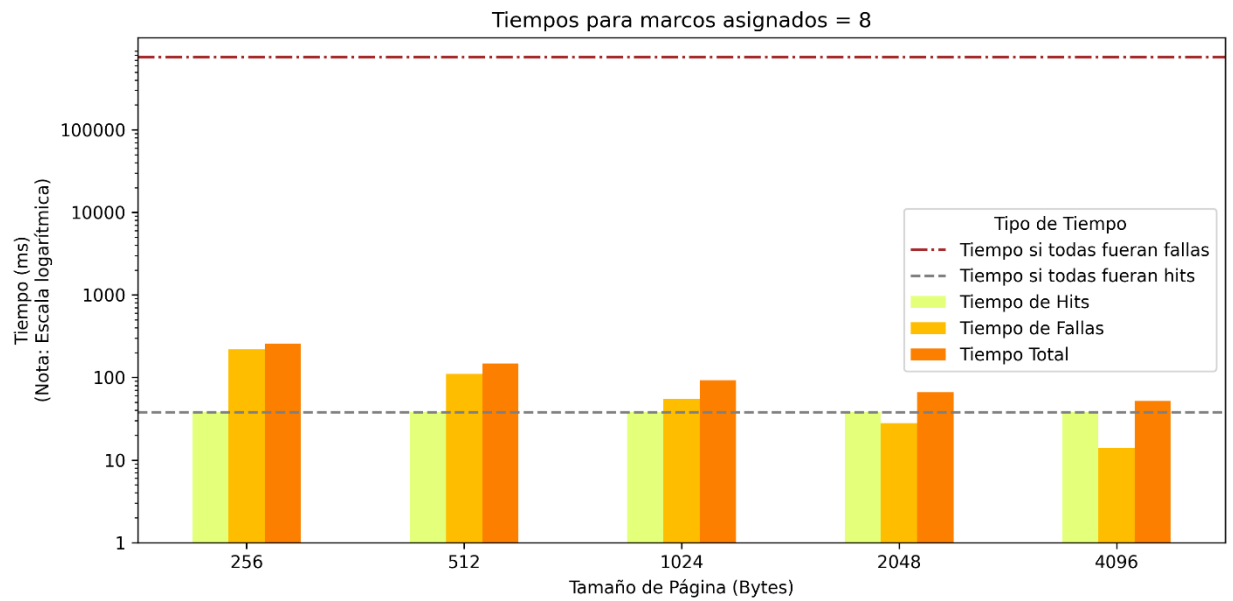
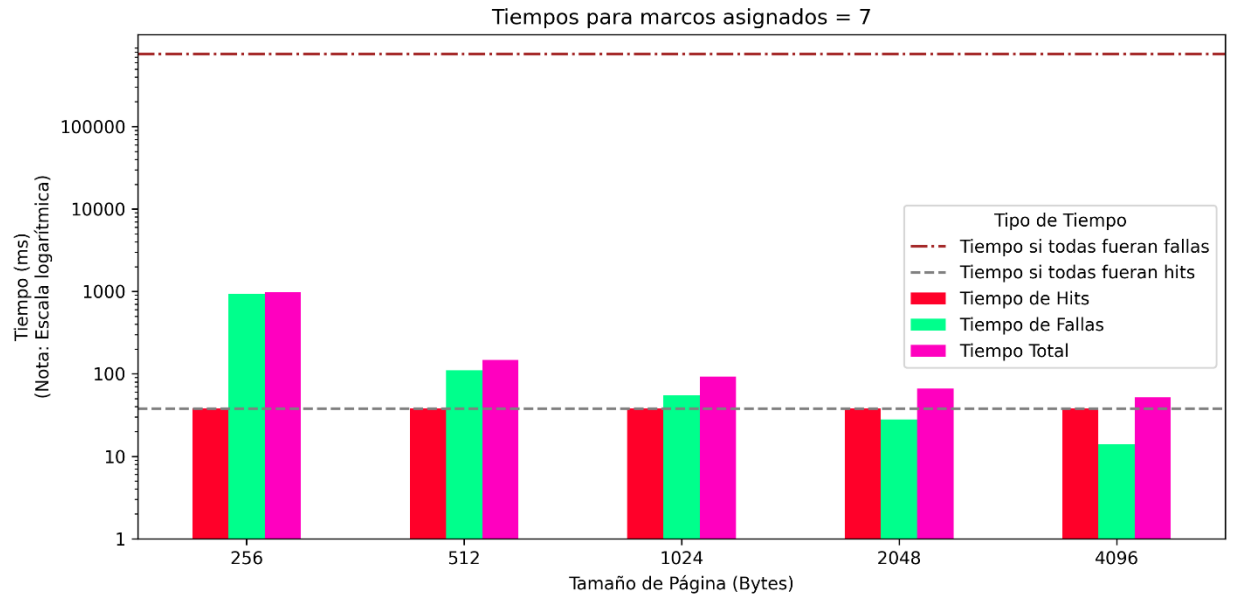
Gráficas de tiempo

Para la imagen **caso2-parrotspeq.bmp** mostramos las gráficas de tiempo para cada marco:









¿Corresponden a los resultados que esperaba, con respecto al número de marcos asignados?

Los resultados obtenidos muestran que a medida que se asignan más marcos de página, la cantidad de fallos de página disminuye. Esto era lo esperado, ya que con más marcos disponibles, es posible almacenar una mayor cantidad de páginas en memoria, reduciendo la necesidad de intercambiarlas constantemente debido a restricciones de espacio. Este comportamiento se alinea con la teoría de administración de memoria en sistemas operativos, donde una mayor asignación de marcos mejora el rendimiento al minimizar los accesos a disco, que son costosos en términos de tiempo.

¿Aplicar el filtro sobel representa un problema de localidad alta, media o baja?

El filtro Sobel al operar a través de una convolución, implica que cada píxel es evaluado en función de una ventana local de vecinos. Este mecanismo determina que el acceso a la memoria siga un patrón estructurado y relativamente secuencial dentro de regiones específicas de la imagen. No obstante, debido a que la convolución exige acceder de manera recurrente a áreas próximas, la localidad espacial es considerablemente alta, mientras que la localidad temporal tiende a ser moderada o baja, dado que cada píxel se utiliza por un breve instante y luego deja de ser requerido. En términos generales, la aplicación del filtro Sobel configura un problema de **localidad intermedia**, pues aunque los datos se reutilizan dentro de sectores delimitados, la necesidad de acceder a distintas secciones de la imagen limita la eficacia del caché.