

Carlos Eduardo Figueredo 201813445

Erik Nielsen 201913493

## Informe Caso 2

Departamento de Ingeniería de sistemas y computación

Universidad de Los Andes, Bogotá

{ea.nielsen, ce.figuero} @uniandes.edu.co

### Tabla de contenido

<b><u>1 ESTRUCTURAS DE DATOS</u></b> .....	<b><u>1</u></b>
<b><u>2 ESQUEMA DE SINCRONIZACIÓN</u></b> .....	<b><u>2</u></b>
<b><u>3 TABLA DE DATOS</u></b> .....	<b><u>2</u></b>
<b><u>4 VARIACIÓN EN FALLAS</u></b> .....	<b><u>3</u></b>
<b><u>5 GRÁFICAS</u></b> .....	<b><u>3</u></b>
<b><u>6 ANÁLISIS DE RESULTADOS</u></b> .....	<b><u>4</u></b>

## 1. Estructuras de datos

Las estructuras de datos están declaradas en la clase Main y están manipuladas por las clases ActualizadorTPyMP y ActualizadorRM.

Estructuras	Descripción
Tabla de hash de solo llaves	Dicha tabla de hash se utiliza para saber cuáles referencias están dentro de la cola y se utiliza cuando para omitir unas instrucciones cuando la referencia está en la cola y se vuelve a indicar. Esta estructura se actualiza con cada referencia nueva y va mirando que paginas virtuales están dentro de la cola.
Lista de enteros posiciones	Esta estructura de datos se utiliza para determinar en qué página de memoria RAM se encuentra una referencia en memoria virtual, la lista se actualiza cada vez que se modifica la cola, ya que, si se realiza el algoritmo de remplazo, otra página de memoria virtual entrara en dicho marco de página en memoria RAM.
Cola	Esta estructura de datos se utiliza para conocer las páginas que han sido utilizadas recientemente y poder utilizar el algoritmo de LRU. La cola se actualiza cada vez que hay una referencia de modo que se agrega o elimina un elemento por cada iteración. Adicionalmente, cuando el tamaño de la cola es igual al número

	de páginas en memoria RAM, esta saca la última, la cual es la página menos recientemente usada. La actualización depende de que página de la memoria RAM está vacía, o si se necesita usar el algoritmo de remplazo, a su vez, depende del si la referencia prende o el bit de lectura o de escritura.
Tabla de Hash con llave valor	Esta es la estructura de datos que representa la tabla de páginas completa, es decir, con los bits de lectura y escritura. Esta página se actualiza cada vez que hay una referencia, donde lo único que se modifica es el valor, el cual es un arreglo de 3 enteros, donde la primera posición representa la página en memoria real que esta dicha información, la segunda el bit de lectura y la tercera el bit de escritura.
Matriz de String	Es una matriz que guarda los valores recibidos por la lectura del archivo, cada fila es una instrucción, en la primera columna tiene la referencia al marco de página y en la segunda tiene el tipo de referencia que ocurre en esa instrucción.
Fallos	Actualiza el número de fallas de página cada vez que no encuentra una página que necesita y se tiene que llevar a cabo el algoritmo de envejecimiento.

## 2. Esquema de sincronización

Los datos que se comparten son los que están en la tabla Hash llamada “valoresTablas”, esta tabla es actualizada por el thread 1 llamado ActualizadorTPyMP y por el thread 2 llamado ActualizadorRM. El ActualizadorTPyMP va recorriendo las instrucciones y actualizando cada 1 ms los estados de valoresTablas y el marco de páginas. El ActualizadorRM hace limpieza de bits de referencia que están dentro de valoresTablas y lo hace cada 20 ms.

## 3. Tabla de datos

Paginas virtuales	Paginas en memoria RAM	#Fallos
16	4	87
	6	61
	8	32
	10	16
	12	16
	14	15
32	4	323
	6	208
	8	101
	10	90
	12	78
	14	67
64	4	401

	6	351
	8	292
	10	245
	12	197
	14	151
128	4	467
	6	439
	8	420
	10	402
	12	379
	14	330

## 4. Variación en fallas

La variación en el número de fallas ocurre debido a que el hilo que maneja algoritmo de envejecimiento a veces puede modificar bits de referencia justo antes (o después) de que el algoritmo NRU retire las páginas del marco de páginas lo cual hará que a veces realice más o menos fallos de página.

## 5. Gráfica

Para la gráfica se comparó el número de fallas por página con la cantidad de marcos y esto por cada número de páginas virtuales.

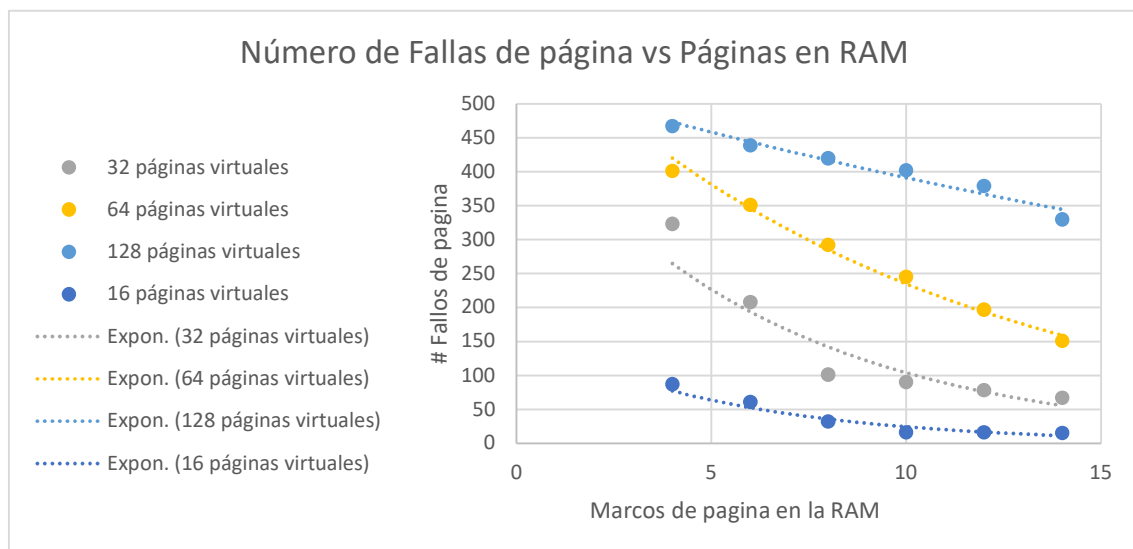


Ilustración 1. Número de Fallas de página vs Páginas en RAM

## **6. Análisis de resultados**

Como se puede evidenciar en la ilustración 1, al aumentar el número de marcos de página asignados al proceso en la memoria RAM, disminuye el número de fallos de página hasta que este valor se estabiliza. Este comportamiento es similar a una función exponencial decreciente, por lo cual los fallos nunca llegarían a cero. La ilustración 1 tiene sentido, ya que si hay una mayor cantidad de marcos de páginas por un mismo valor de tablas de páginas, la probabilidad de que la página no esté cargada disminuye.