



Universidad de  
**los Andes**



**FACULTAD  
DE INGENIERÍA  
Y CIENCIAS  
APLICADAS**

## **Informe Tarea 2 Sistemas Operativos**

Integrantes: Eduardo Saldivia  
Emilio Arias  
Profesor: Claudio Alvarez

Jueves 17 de Octubre del 2019

## Algoritmos

Algoritmo 1: Patrón lineal que recorre de inicio a fin las páginas, dándoles valores a estas, para luego leer de inicio a fin todos los valores.

Algoritmo 2: Patrón aleatorio que consiste en entrar 5000 veces a una posición aleatoria de página. Luego, en el caso que esta sea par, el algoritmo escribe en ella. Mientras que en el caso que sea impar, el algoritmo lee el bloque elegido y hace escritura en el antecesor.

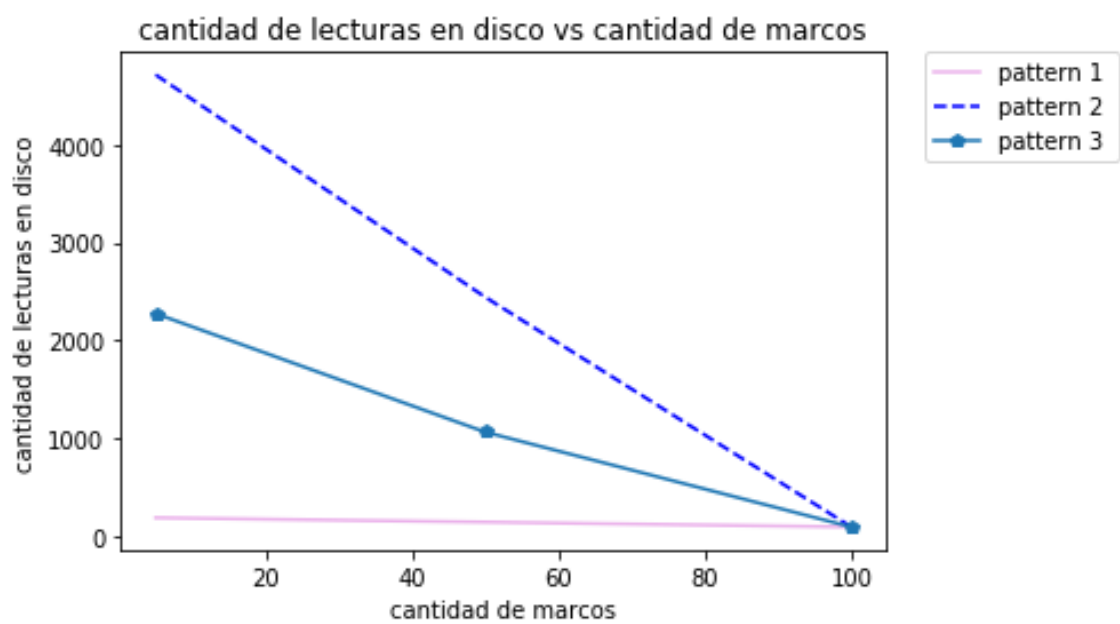
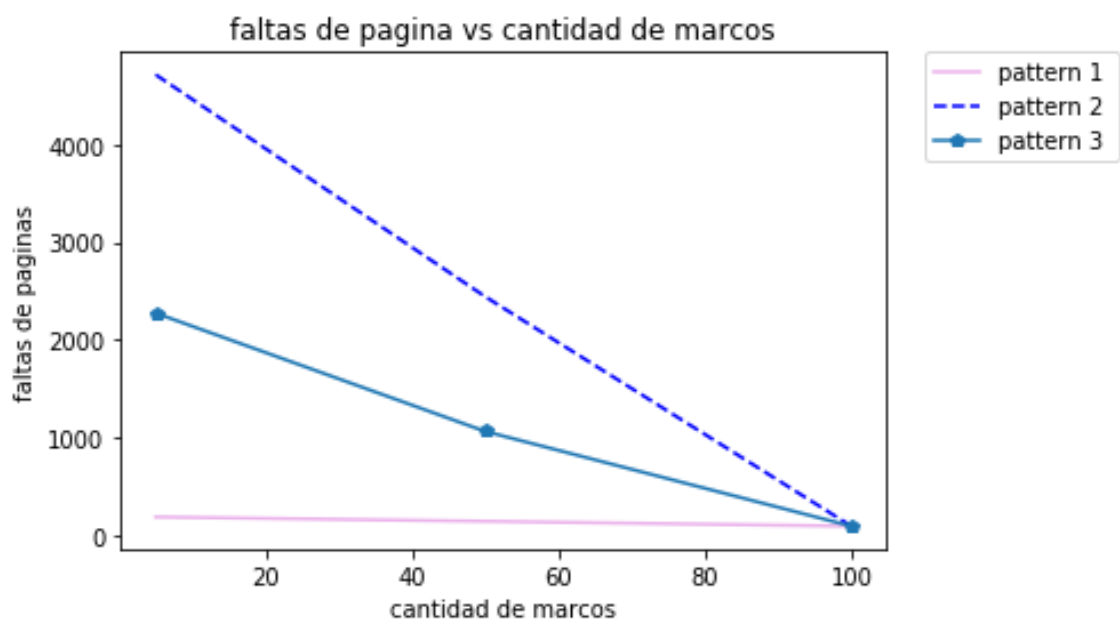
Algoritmo 3: Consiste de una iteración de 100 veces, donde a una variable "a" se le asigna un valor aleatorio, para luego, hacer otra iteración, pero esta vez de 20 veces, donde a una variable "b" se le asigna otro valor aleatorio, y luego, se le actualiza el valor la página en la posición "a" por el valor de la página en la posición "b".

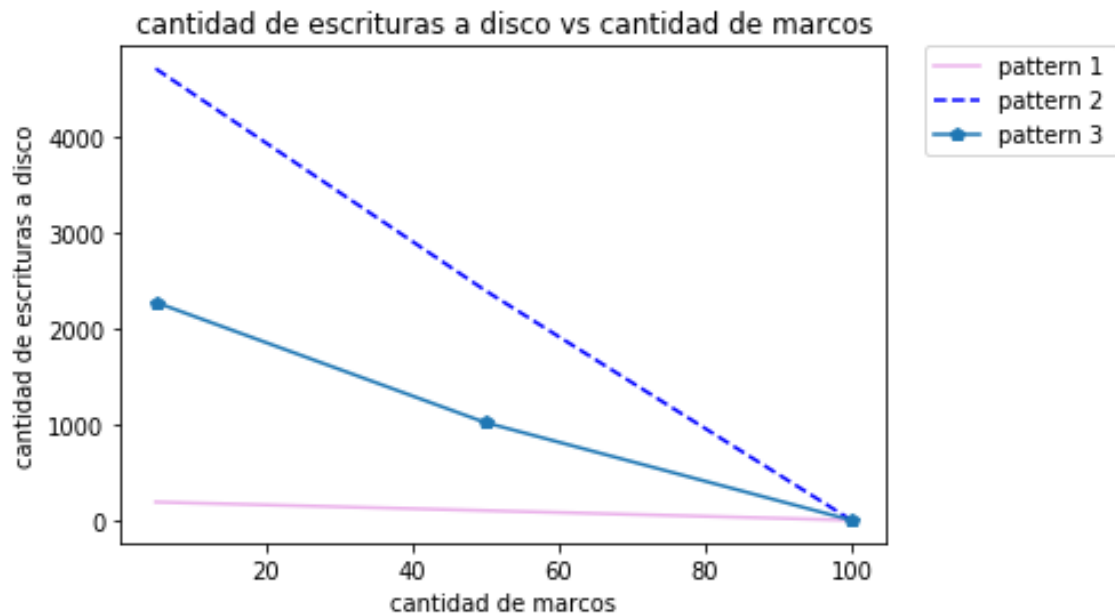
## Hipótesis

Algoritmo 1: Como el algoritmo recorre todas las paginas por orden, debería haber tantas faltas de pagina como llamados a página haga.

Algoritmo 2: El rendimiento debería ser un poco mejor que en el 1, debido a que como entra a las paginas de manera aleatoria, algunas veces podría tener suerte y no tener falta de página, pero a grandes rasgos se comporta parecido al algoritmo 1.

Algoritmo 3: Debería tener tantas faltas de pagina como un poco mas de la mitad o menos, que llamados a páginas. Esto debido a que dentro de los ciclos se actualizan los valores de tal manera que la variable b va a tener faltas de paginas mientras que la variable a va a tener casos que están cargados.





## Conclusión

Concluimos que el 3er algoritmo se comporta mejor que los otros dos respecto a la cantidad de faltas de páginas, debido a su método de actuar, pero esto no es sorpresa, ya que el patrón fue diseñado para simular una situación mas real, en el sentido que existen paginas que son llamadas mas veces que otras para usar un programa, por lo que, al ser cargadas una vez, no incurrir en mas faltas de paginas una vez cargadas.

También, cuando se concluyo que en el caso que se iteraba con 100 números de marcos, se obtenían exactamente 100 faltas de páginas, esto debido a que al tener el mismo numero de marcos que de páginas, estas al ser cargadas incurrían en faltas de paginas una sola vez ya que luego se tenían todas las paginas cargadas en memorias por lo que si se volvían a necesitar, ya estaban listas. Por lo que, si  $nframes = npages = \text{faltas de pagina}$ .