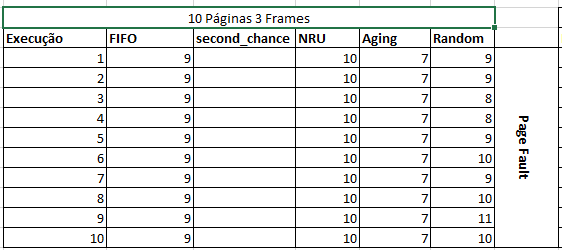
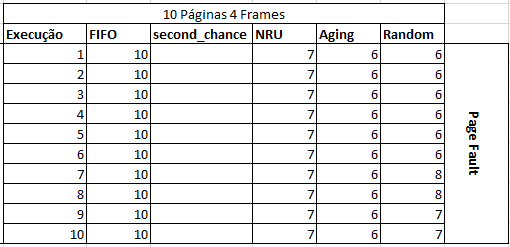
**FIFO** – Substitui a página mais antiga

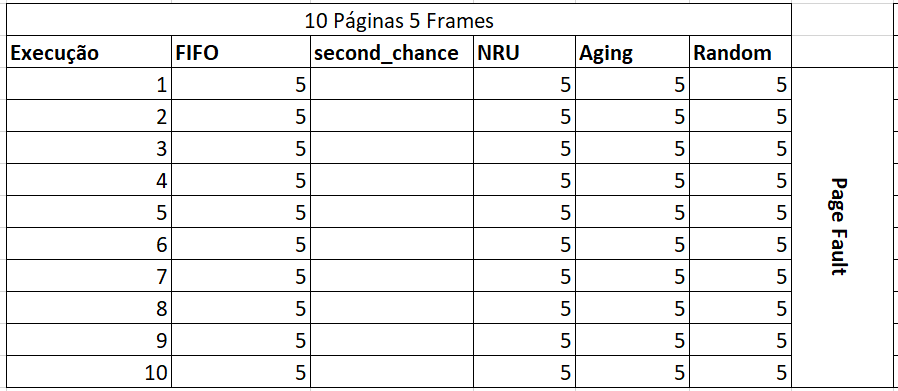
**Second** **Chance** – Substitui a página mais antiga na qual o bit-R seja 0.(Não obtive sucesso ao tentar implementar este algorítimo, a parte implementada está presente no arquivo main.c porém seu funcionamento não está correto.)

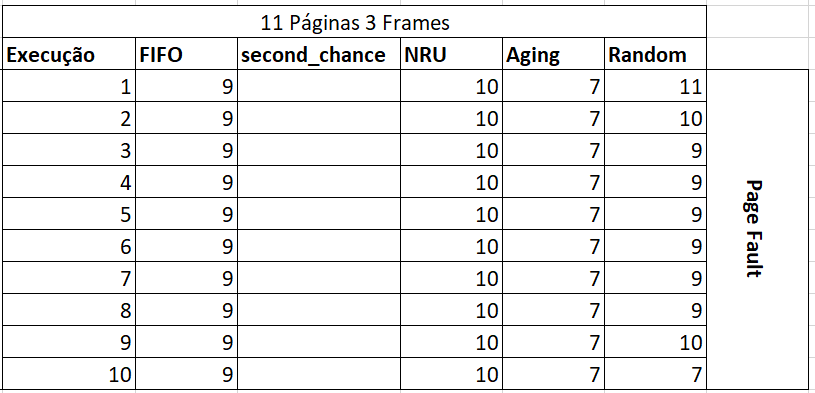
**NRU** – Substitui a página pertencente à menor classe dentre as seguintes:  
 **Classe 0**. Bit R = 0 e Bit M = 0) não referenciada nem modificada.  
 **Classe 1**. Bit R = 0 e Bit M = 1) não referenciada, mas modificada.  
 **Classe 2**. Bit R = 1 e Bit M = 0) referenciada, mas não modificada.  
 **Classe 3.** Bit R = 1 e Bit M = 1) referenciada e modificada.

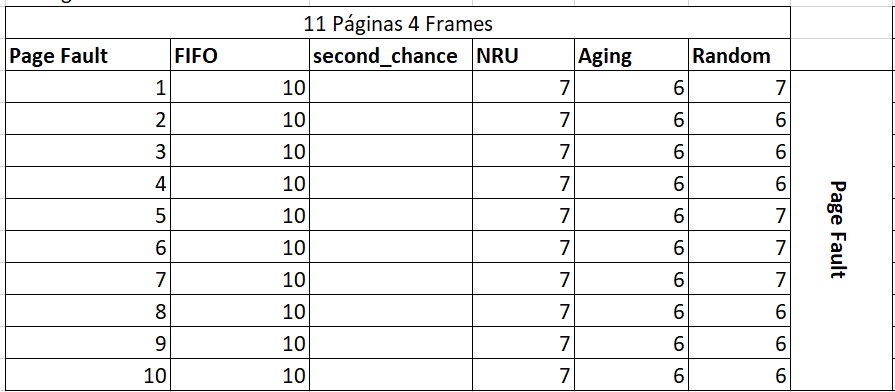
**Aging –** Cria um binário e insere neste o bit R, a cada clock do relógio este é movido 1 bit para a direita.











Visto as execuções verifica-se o motivo do algorítimo fifo ser pouco utilizado, mesmo com o aumento de 3 para frames, suas Page Faults aumentaram, diminuindo apenas com 5 frames que comportam todas as páginas acessadas pelo arquivo DAT, assim sendo em quase todos os casos menos efetivo que o algorítimo random. Os algorítimos NRU e Aging se tornam mais efetivos conforme aumentam os frames disponíveis. Todos os algorítimos igualam sua execução com 5 frames, visto que com 5 frames são comportadas todas as páginas virtuais do arquivo .dat utilizado para testes.