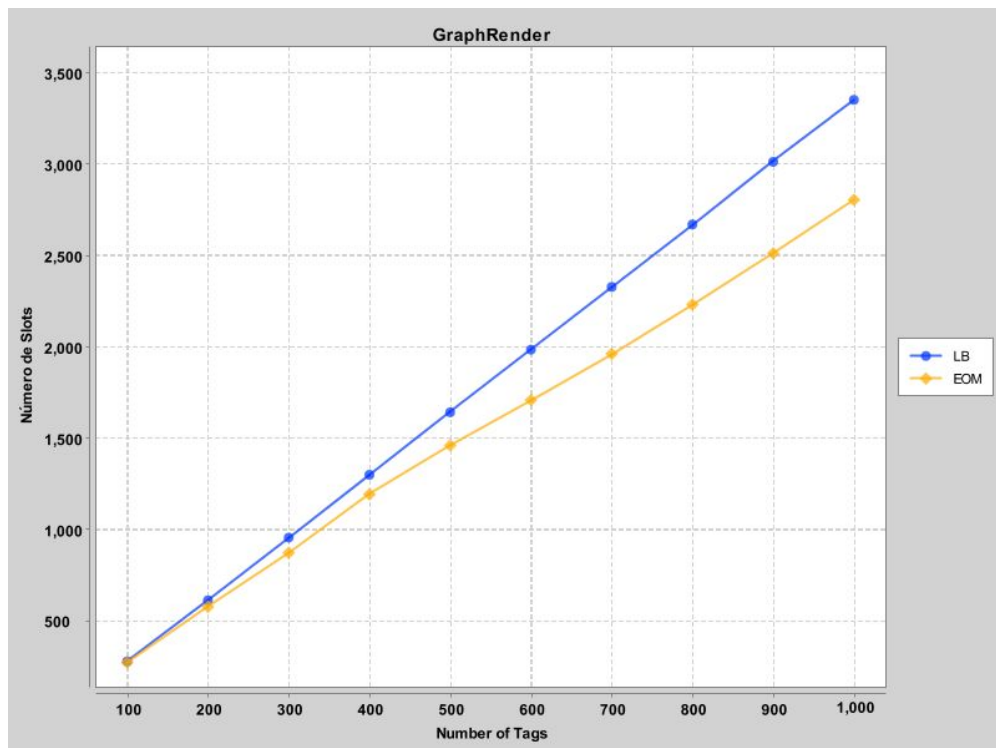


Estimadores para o DFSA

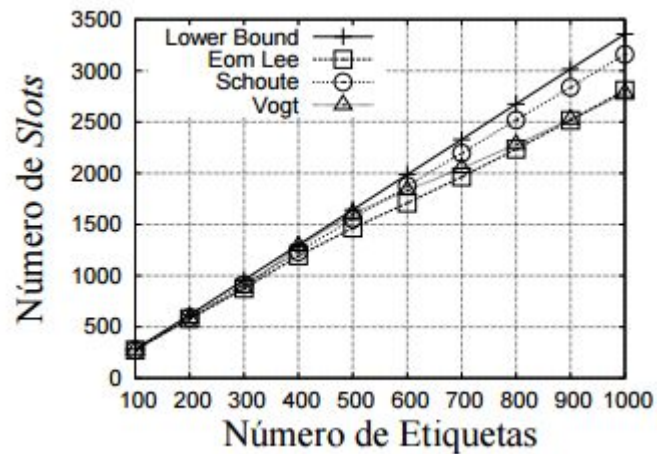
Atividade 5 - Atualizada com Chen, Vahedi, e correções

- Bruno Dutra de Lemos Neto (bdln)
- Danilo Alfredo Marinho de Souza (dams)
- Lucas de Souza Albuquerque (lsa2)
- Victor Nunes de Farias Neves (vnfn)

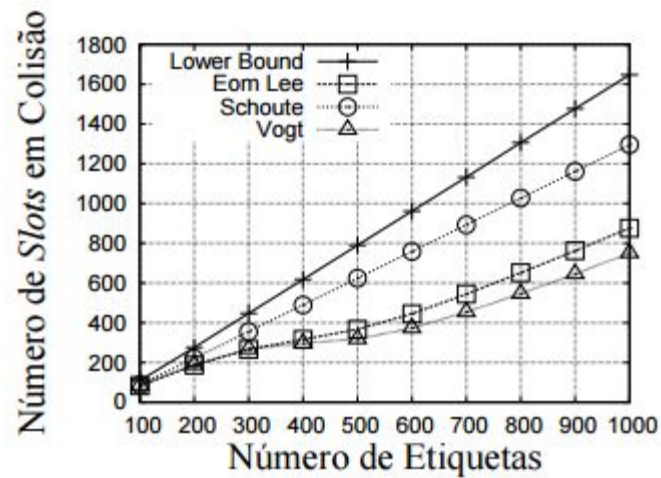
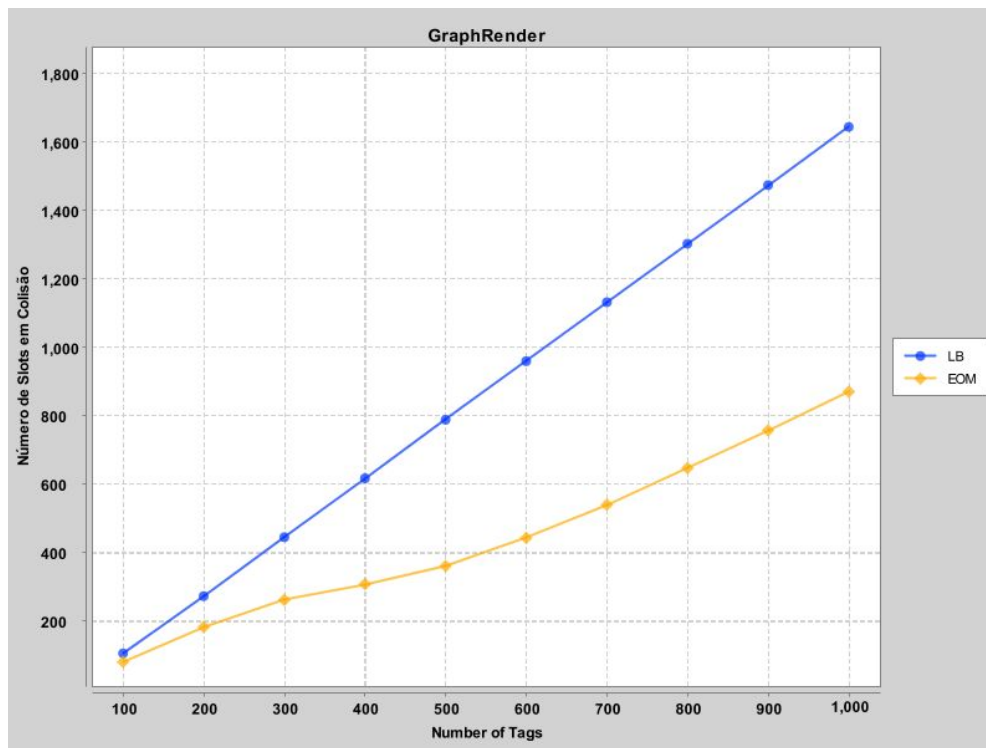
Lower Bound, Eom-Lee (Slots Totais)



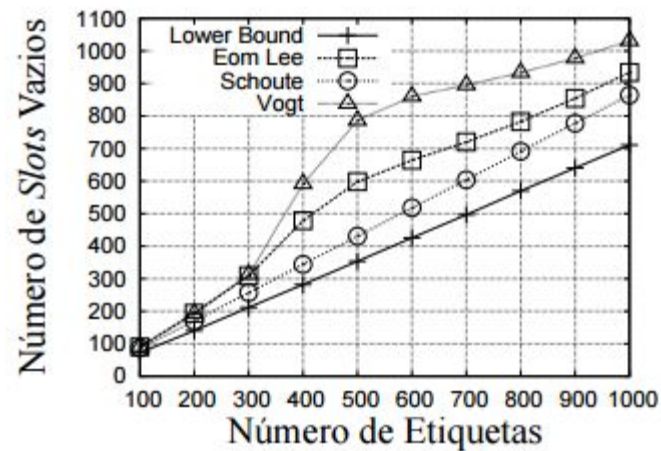
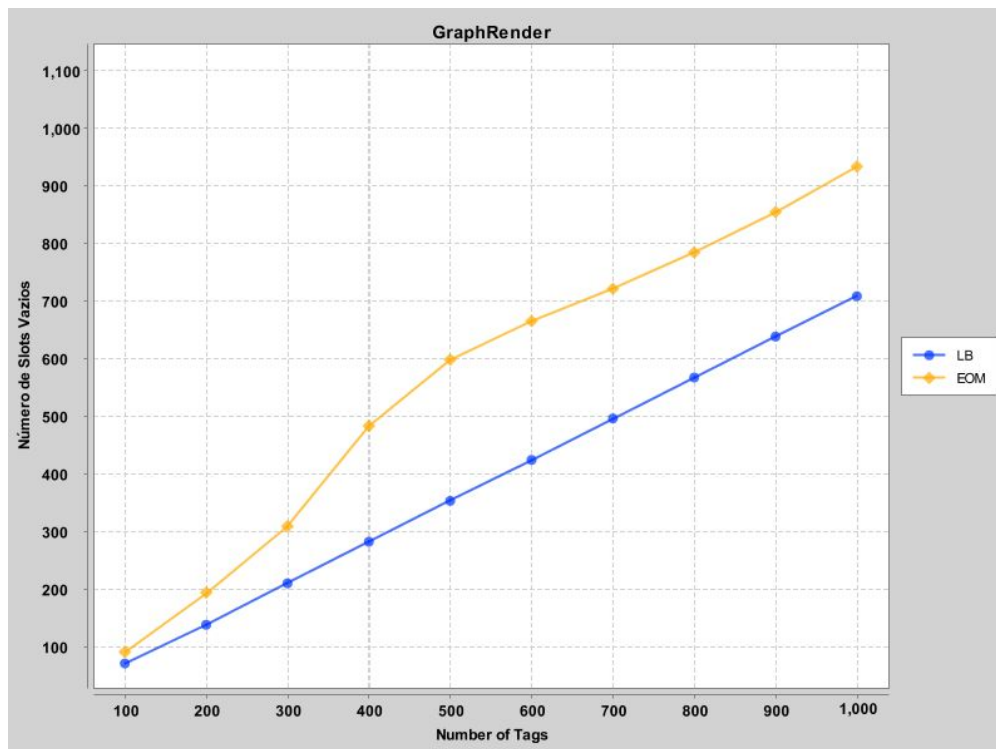
*Escala foram ajustadas para comparação com os gráficos



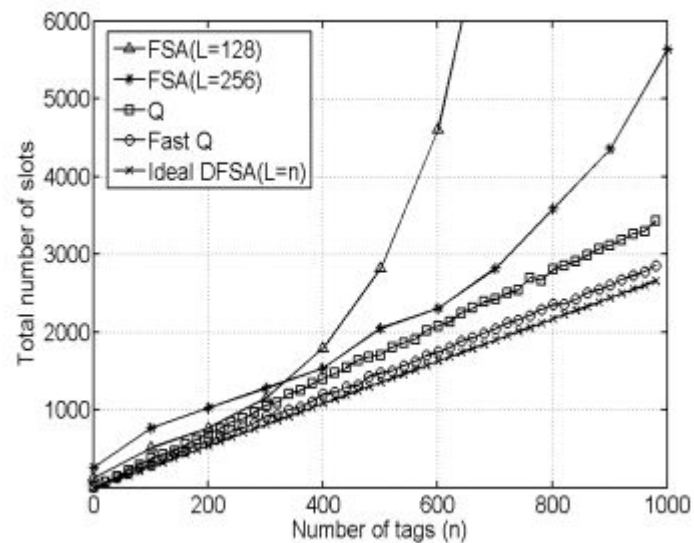
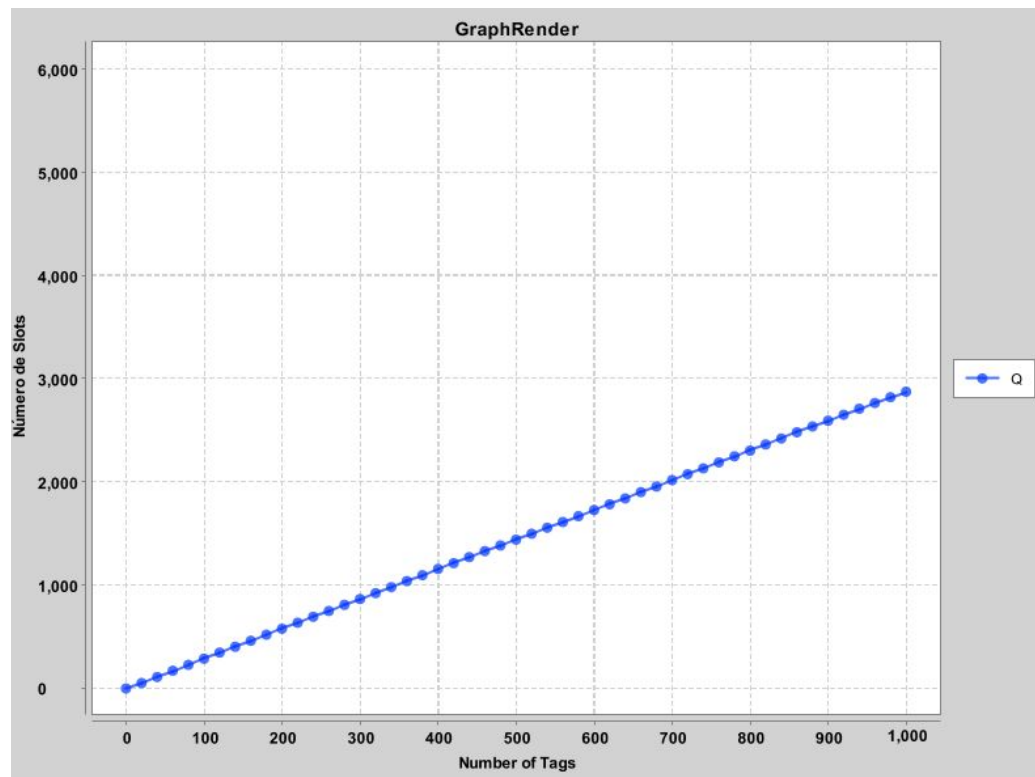
Lower Bound, Eom-Lee (Slots em Colisão)



Lower Bound, Eom-Lee (Slots Vazios)

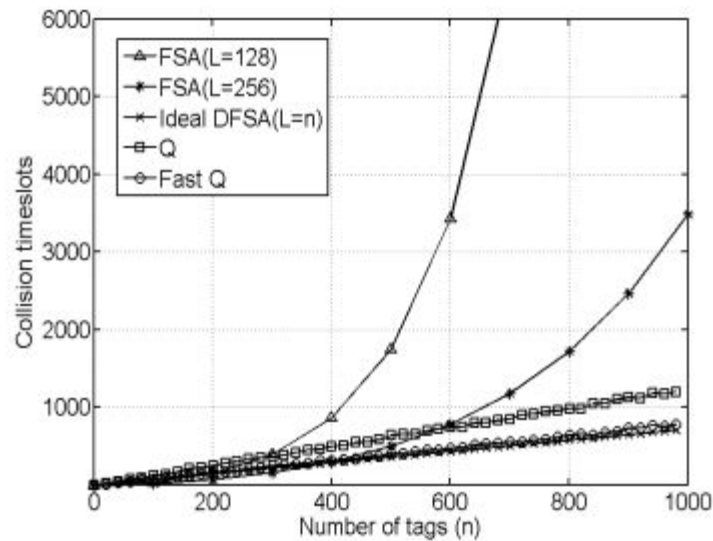
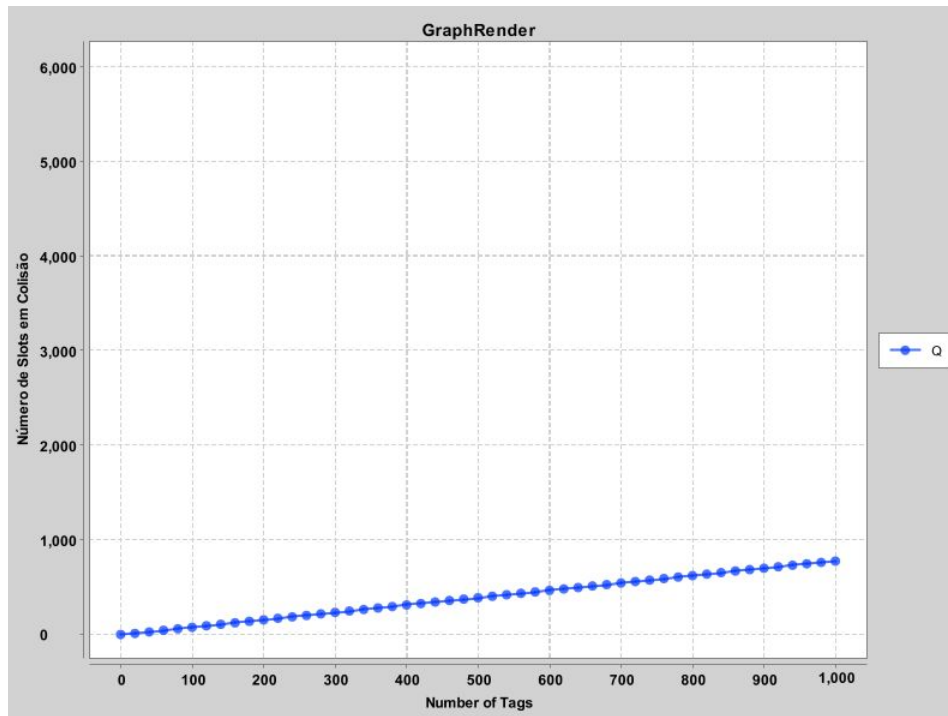


Q (Número de Slots) - FAST-Q

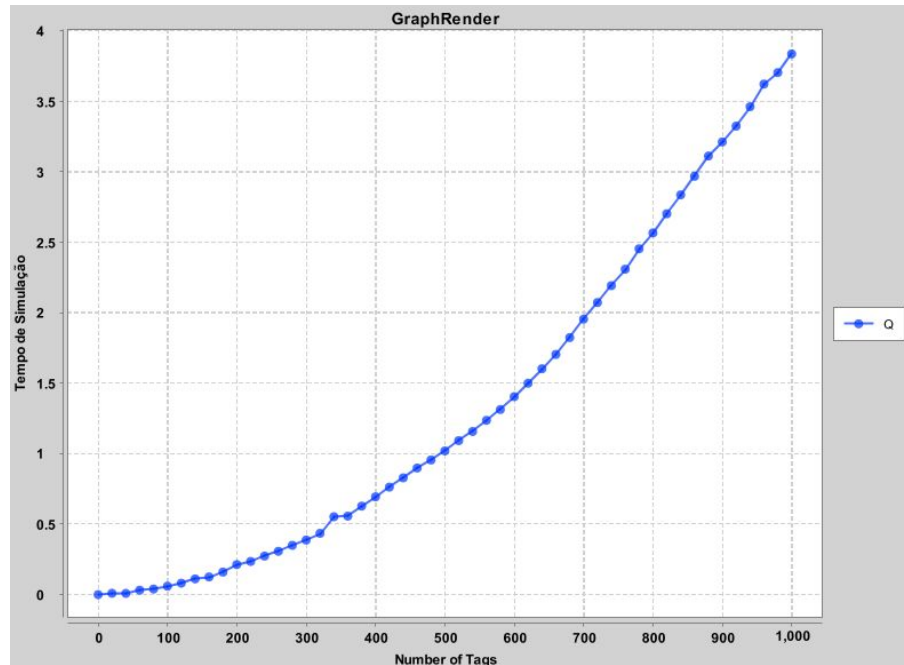
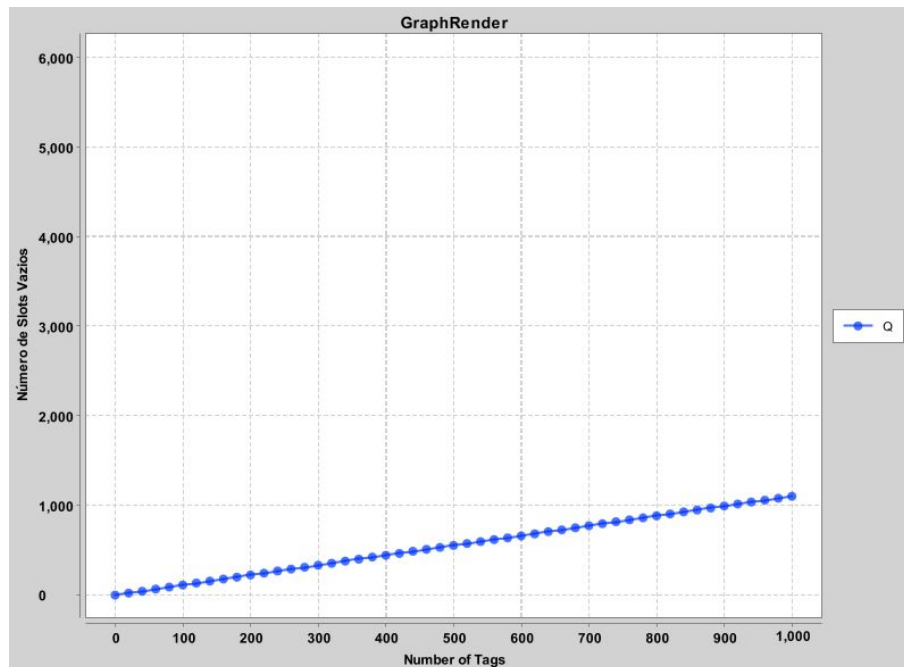


*A implementação do Q foi a mostrada no algoritmo da especificação: o FAST-Q com constantes diferentes para colisão (C_{col}) e slots vazios (C_{idle})

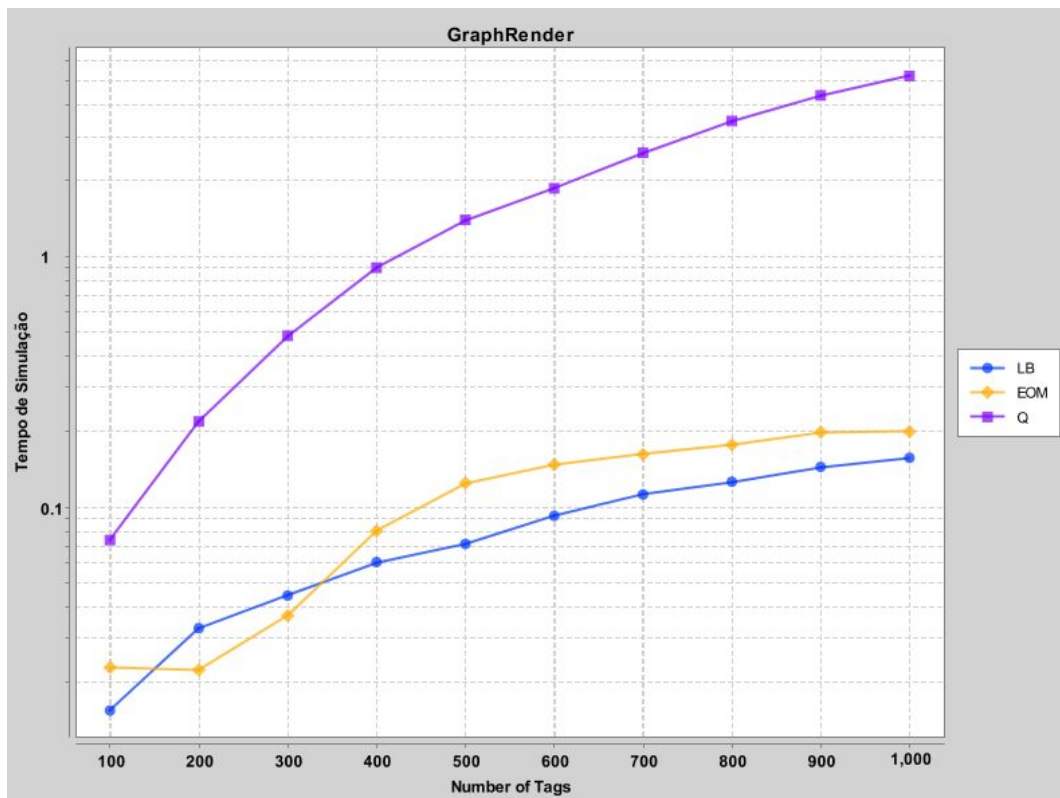
Q (Número de Slots em Colisão)



Q (Número de Slots Vazios e Tempo)



Lower Bound, Eom-Lee e Q (Tempo - Escala Logarítmica)

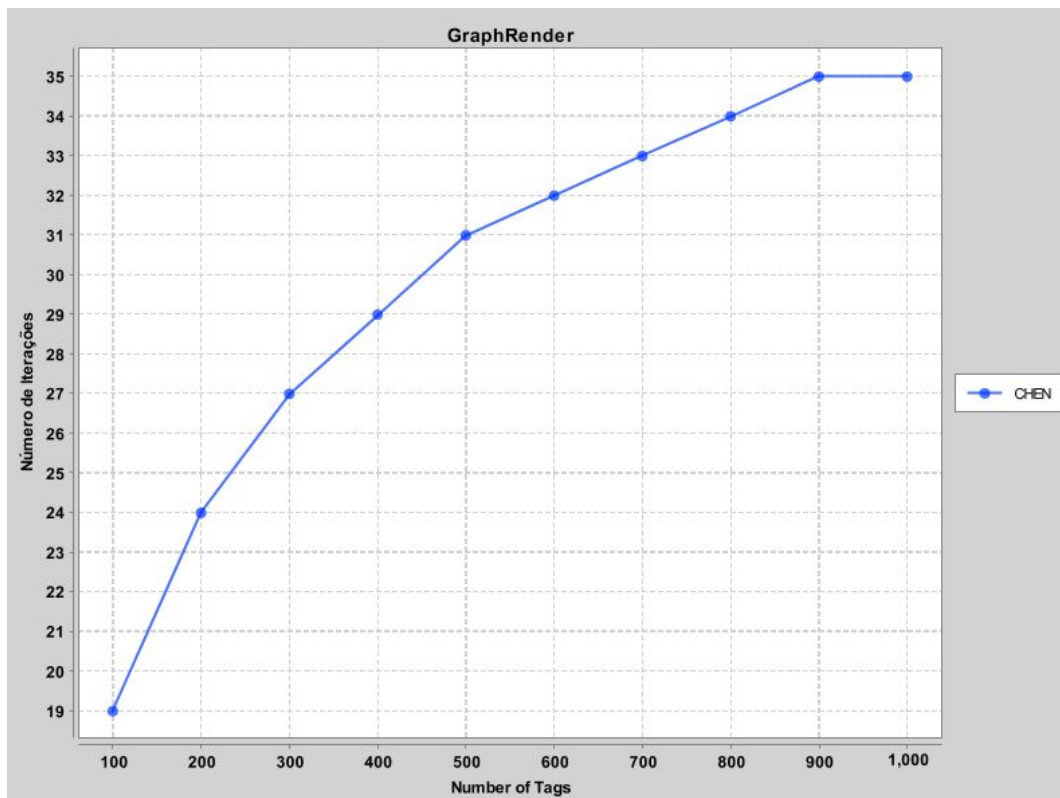


*Tempo agora apresentado em uma escala logarítmica.

*Após a primeira apresentação, foi-se corrigido um erro na implementação que fazia cálculos/loops desnecessários e tornava o tempo de cálculo do Lower Bound e Eom-Lee em um tempo exponencial

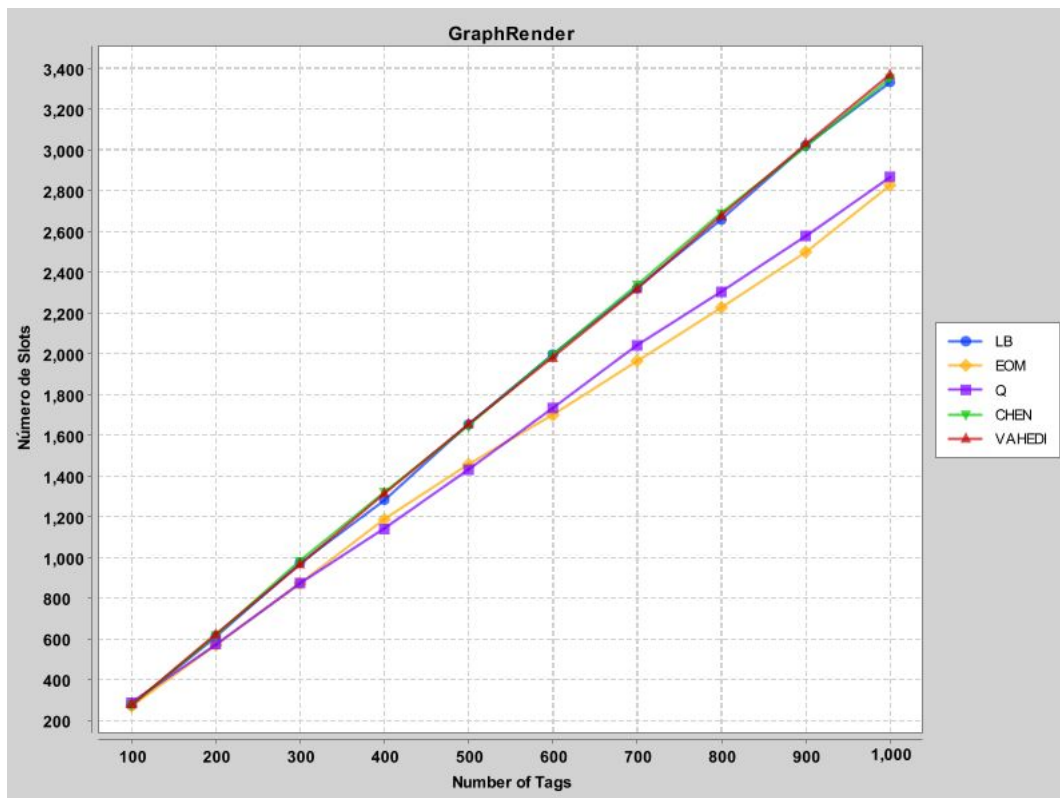
*A mesma mudança melhorou consideravelmente o tempo de Q

Chen (Iterações do algoritmo)



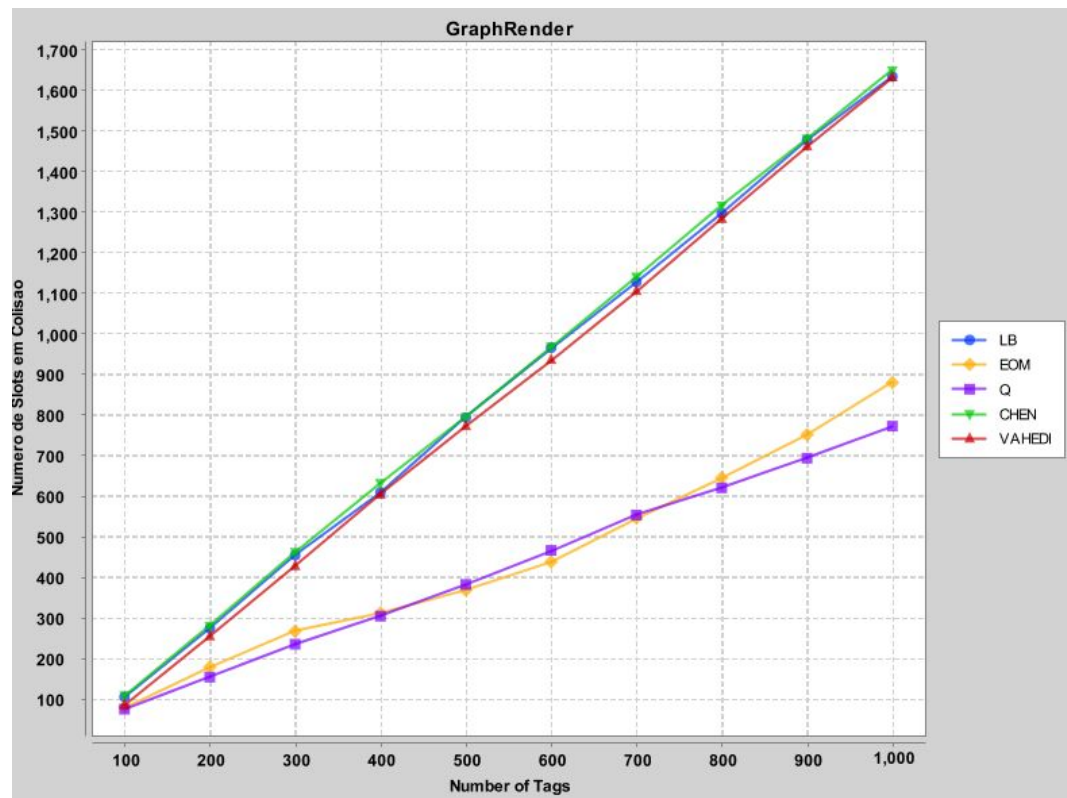
*Iterações dos loops ($prev < next$) para o algoritmo Chen

Lower Bound, Eom-Lee, Q, Chen e Vahedi (Slots)

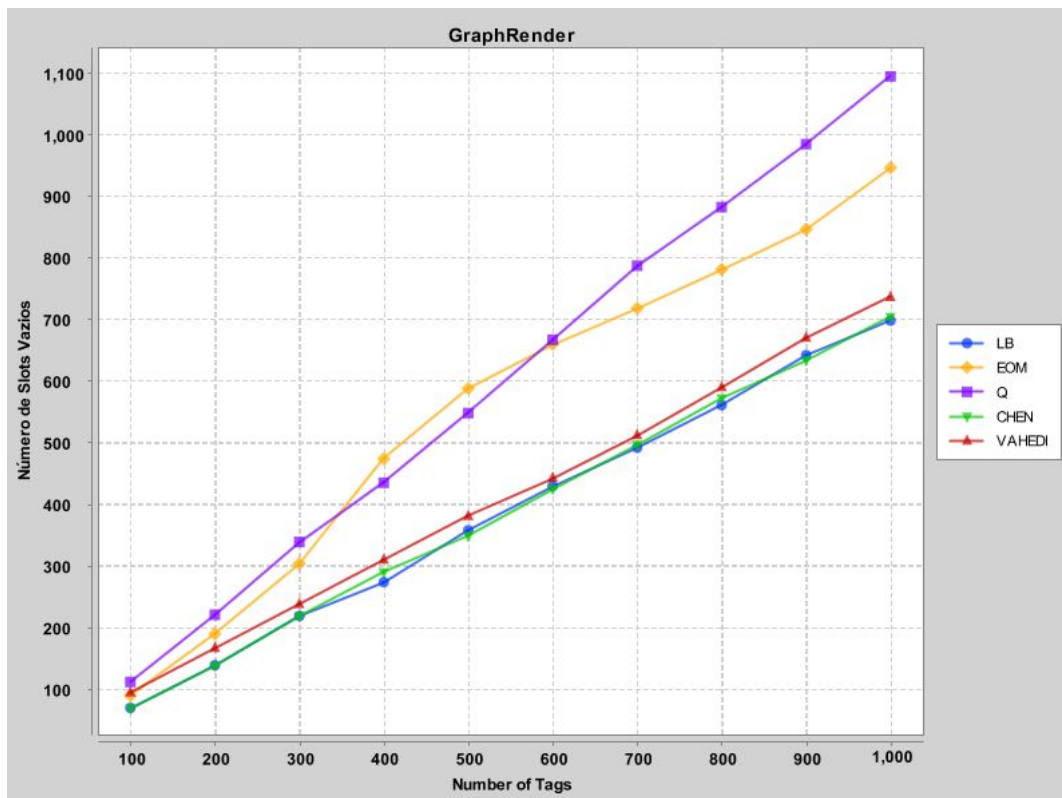


*Devido à demora do Vahedi, estes gráficos foram feitos com 100 repetições cada, em vez de 2000

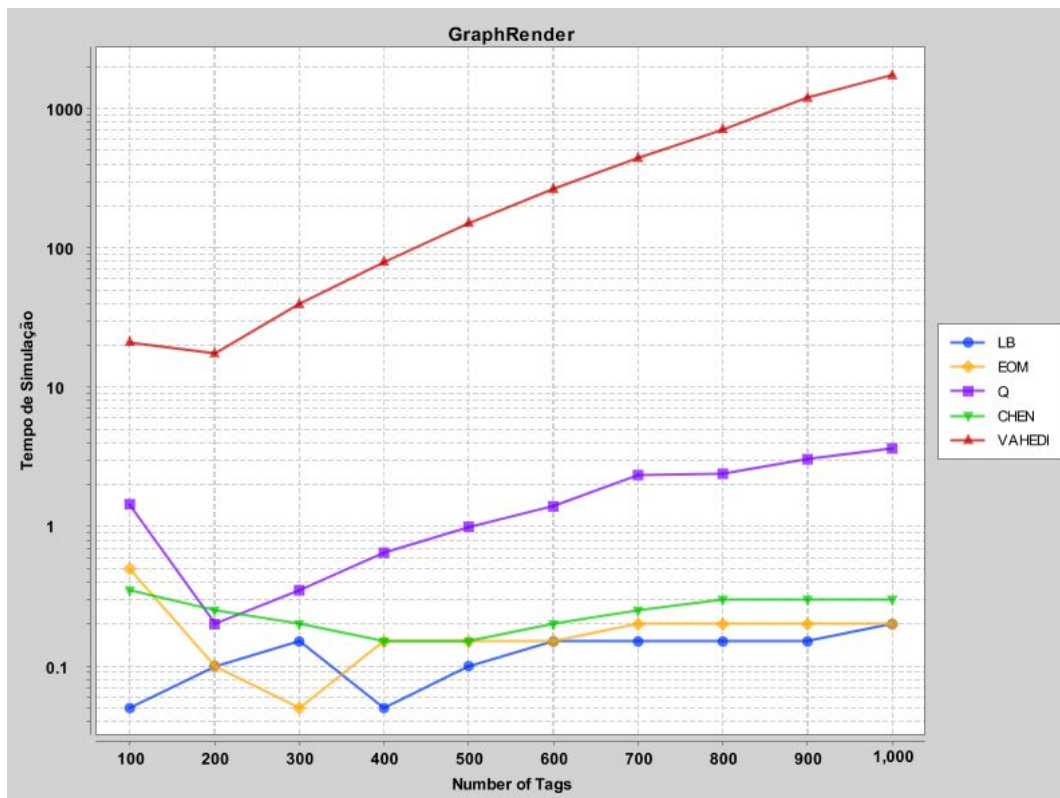
Lower Bound, Eom-Lee, Q, Chen e Vahedi (Slots em Colisão)



Lower Bound, Eom-Lee, Q, Chen e Vahedi (Slots Vazios)



Lower Bound, Eom-Lee, Q, Chen e Vahedi (Tempo)



*Os fatoriais de 0 até 5000 foram calculados e armazenados com antecedência, melhorando consideravelmente o tempo de tanto o algoritmo CHEN quanto o VAHEDI

*O tempo ainda grande do VAHEDI se deve aos dois loops para o cálculo da variável P3, que são baseados nos números de slots de colisão: a complexidade, para cada novo cálculo do tamanho do próximo frame, se aproxima de $O(n^3)$