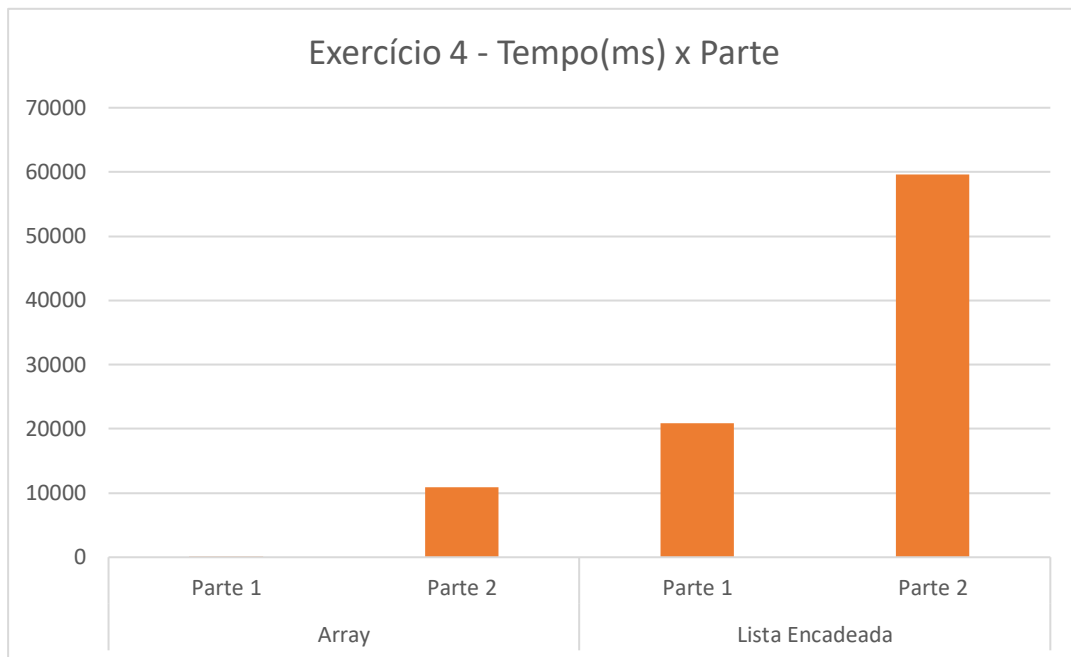


Exercício 4 – Análise dos Resultados



A parte 1 contém um loop para inserção de novos elementos e um loop que pega elementos aleatoriamente armazenados nas estruturas. Diante disso, percebe-se que a velocidade da array(80ms) para essas 2 operações é superior a de uma lista encadeada(20905ms) já que apesar do tempo de inserção ser semelhante nessas estruturas, o acesso direto aos elementos de um array é muito mais rápido que o acesso a elementos aleatoriamente em uma lista encadeada já que é preciso percorrê-la a partir de seu início até encontrar o elemento do índice solicitado, o que faz essa estrutura ser muita mais lenta(261 vezes mais lenta) do que a array.

Na parte 2, temos operações de remoção e inserção de elementos. Em um array, a operação de remoção é muito cara, já que é preciso mover para a esquerda todos os elementos após aquele que foi deletado. Por isso, a parte 2 é 135 vezes mais lenta que a parte 1, enquanto que na lista encadeada, é uma questão de apenas alterar ponteiros e, por isso, essa operação é apenas 2,8 vezes mais cara que a parte 1. Mesmo assim, como a operação de get é lenta em uma lista encadeada, a array continua performando melhor pelo menos para o número de elementos inseridos (10^6). Esse número foi escolhido devido ao longo tempo de execução ao se trabalhar com 10^8 elementos.