

## TC-CAL07

**Alunos: Lucas Meneghelli Pereira e Vinicius Gasparini**

### PILHA

<b>Função</b>	<b>Complexidade</b>
Empilha	$O(1)$ para todos os casos.
Desempilha	$O(1)$ para todos os casos.
Limpar	$O(n)$ para todos os casos.
Print	$O(n)$ para todos os casos.
Tamanho	$O(1)$ para todos os casos.

### LISTA ENCADEADA

<b>Função</b>	<b>Complexidade</b>
Adicionar	$O(1)$ para todos os casos.
Anexar	$O(n)$ para todos os casos.
Remover	$O(1)$ para melhor caso quando valor estiver na primeira posição da lista, e $O(n)$ para pior caso quando valor não existir na lista.
Encontrar	$O(1)$ para melhor caso quando valor estiver na primeira posição da lista, e $O(n)$ para pior caso quando valor não existir na lista.
RemoverDuplicatas	$O(n)$ para todos os casos. Pior caso chega apenas a $2O(n)$ , sendo igual a $O(n)$ .

### ARVORE VP

<b>Função</b>	<b>Complexidade</b>
rotateLeft	$O(1)$ para todos os casos.
Inserir	$O(1)$ para melhor caso quando valor estiver na raiz da árvore, e $O(n \log(n))$ para pior caso quando valor estiver em alguma folha da árvore.
Busca	$O(1)$ para melhor caso quando valor estiver na raiz da árvore, e $O(n \log(n))$ para pior caso quando valor estiver em alguma folha da árvore.
PreOrder	$O(n)$ para todos os casos.
Maximo	$O(n \log(n))$ para todos os casos.