



**Universidade Federal do Ceará**

**Estrutura de Dados Avançada**

**Arthur Antunes Nogueira da Silva**

**Afonso Barbosa de Souza Neto**

**Matrículas: 368334, 369581**

## **Relatório e Análise de Dados**

### **Trabalho Final**

- **Descrição da máquina geradora dos resultados;**

Notebook Asus Z450LA-WX007T

Processador: Intel® Core™ i5-5200U Dual Core 2.20 GHz com Turbo Max até 2.7 GHz

Cache: 3 MB

Sistema operacional: Elementary OS

Memória: RAM 8 GB DDR3L

Tipo de memória: DDR3L

Disco rígido: (HD) 1 TB SATA 5400 RPM

Placa de vídeo: Intel Corporation Broadwell-U Integrated Graphics (rev 09)

**Classe:** Empresa;

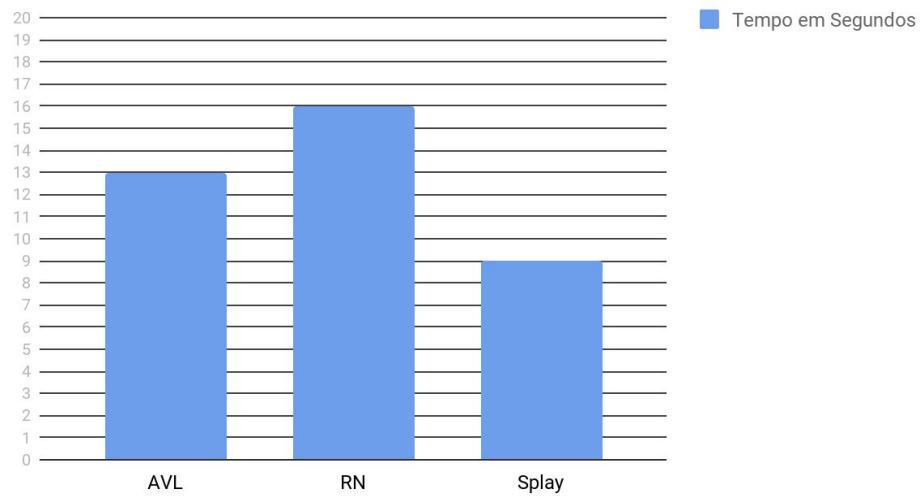
**Chave:** Data;

Para analisarmos melhor os dados para fazer o relatório, primeiro vamos mostrar o gráfico de colunas para cada tipo de operação, e logo após para as instâncias passadas com mais buscas ou mais remoções;

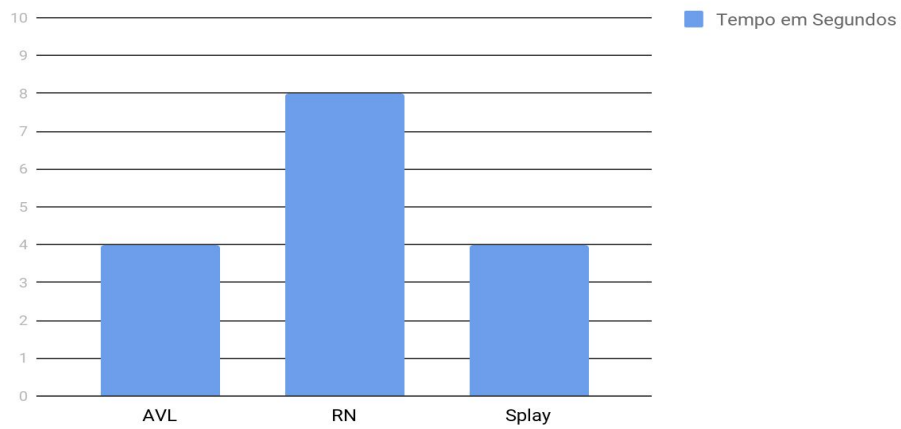
Para complementar melhor o experimento, no último teste da descrição que diz que a cada 20% inserido, fazer busca em 30% dos que foram inseridos e depois remover 10%, experimentamos também o contrário, ou seja, a cada 20% dos registros inseridos, 10% buscados e 30% removidos.

## Gráficos de teste para cada tipo de operação;

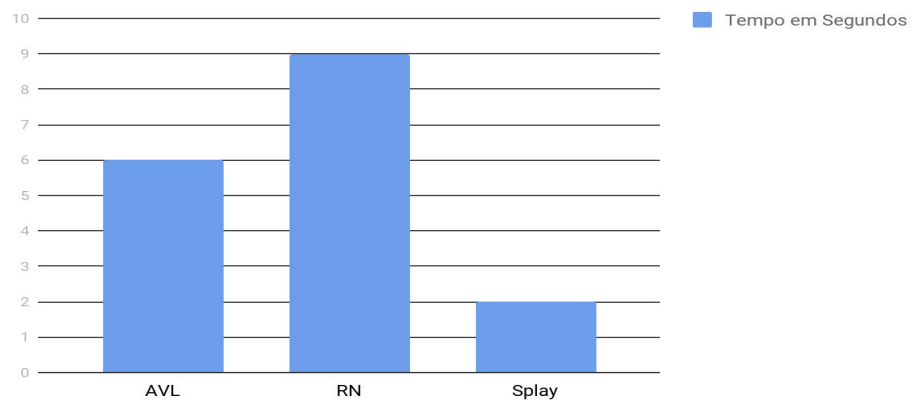
### Inserção



### Busca

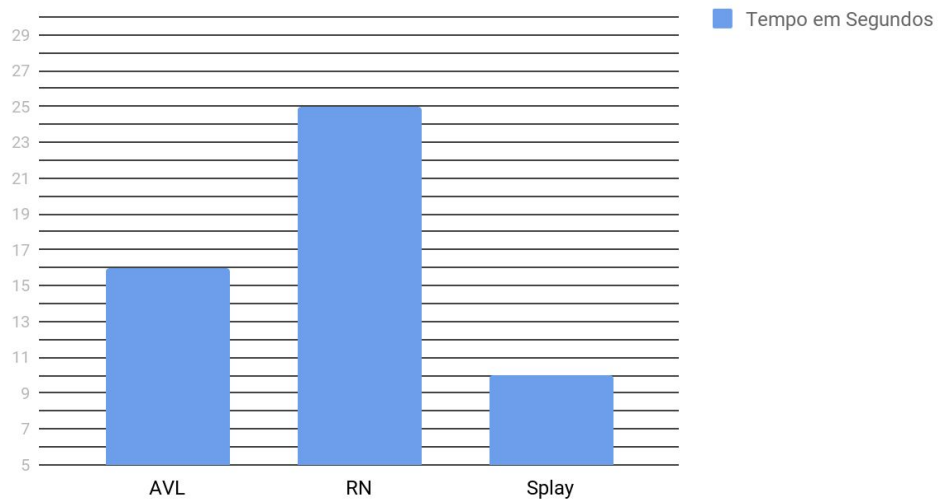


### Remoção

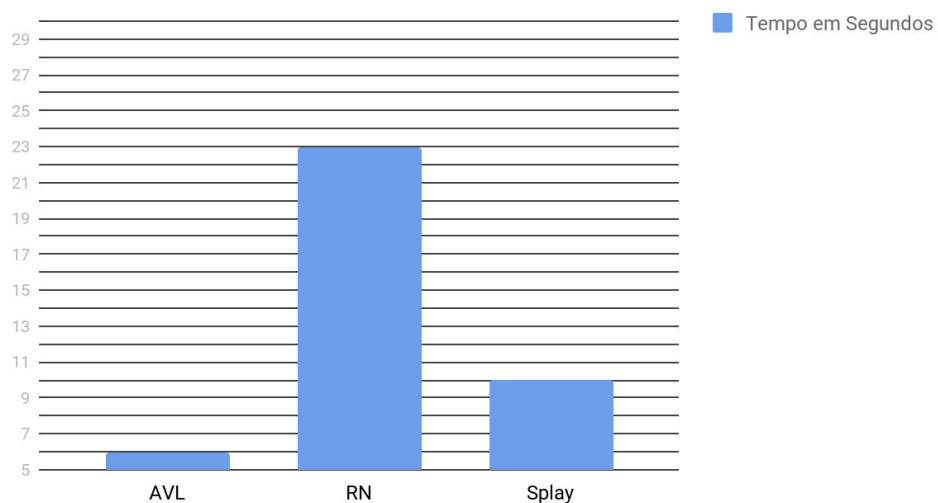


## Gráficos de teste usando as três operações simultaneamente;

A cada 20%, 30% são buscados e 10% removidos



A cada 20%, 30% são removidos e 10% buscados



A princípio, podemos observar que em todos os casos a árvore que apresentou mais tempo de computação em segundos, para todos os testes foi a Rubro Negra, principalmente nos dois últimos testes, que a diferença quando a maioria das operações eram de remoção foi de até 18 segundos, quando comparado a AVL, que já era esperado, pois depois de perceber que nas operações simples o RN demoravam mais tempo, a junção das 3,

independente da porcentagem de tipo de operação ia fazer a RN ter mais tempo de computação.

Outra coisa a se observar é que a Splay foi melhor que a AVL em quase todos os casos, menos no caso quando testamos 30% de remoção após 20% de inserção e busca. O que causa estranheza já que nas operações de remoção a Splay se apresentou 4 segundos melhor que a AVL de tempo de computação, mas é bom ressaltar que como há muitas operações em chaves aleatórias, esse tipo de ocorrido não foge do que é esperado.

**Tabela de tempo de árvore por teste executado**

	Inserção	Busca	Remoção	30% Busca	30%Remoção
<b>AVL</b>	13	4	6	16	6
<b>RN</b>	16	8	9	25	23
<b>Splay</b>	9	4	2	10	10

Essa tabela nos auxilia a ver na maioria dos teste quem foi melhor no geral e para cada caso específico.

No geral, somando todos os tempos computacionais, a Splay se mostrou melhor que as outras, ou seja, num cenário onde todos os nossos testes fossem utilizados, o que faria com eficiência o processo e em menos tempo seria a Splay.

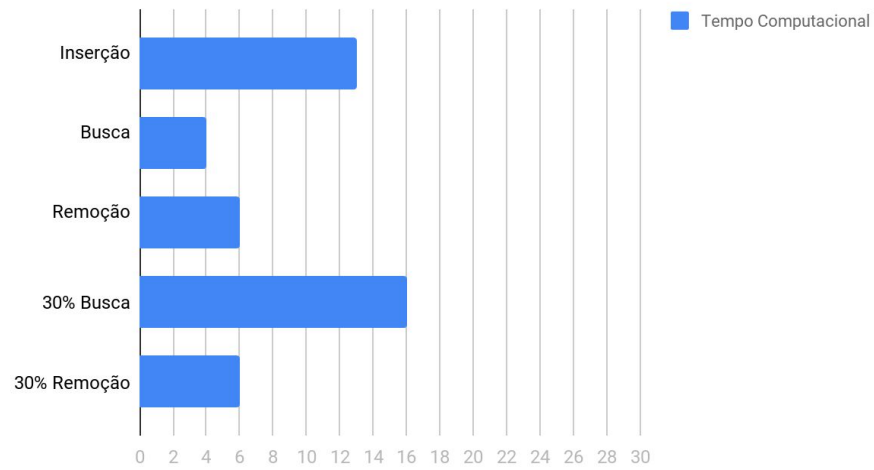
Para uma busca, ou operações com mais remoções, a AVL é tão boa ou melhor que a Splay, pois demonstrou ter um menor tempo de computação. Como a AVL mantém o número de filhos de no máximo 1, ela sempre se apresenta balanceada, e sempre fica perto de  $O(\log n)$  nas operações de busca.

A Splay também é eficiente sempre ela pega o nó mais visitado, e coloca pro topo, ou seja, é bom para operações somente de busca.

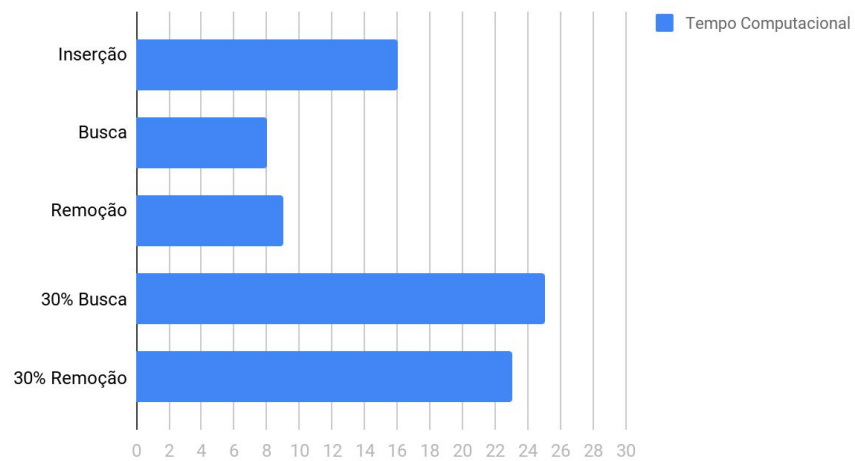
Uma coisa que podemos observar na AVL é que, apesar de as operações de busca ter um menor tempo computacional do que as de remoção, nos últimos testes, o tempo com 30% de remoção foi melhor que o de 30% de busca, também não foge do comum mas é interessante observar.

## Gráfico de cada árvore

### AVL



### RN



### Splay

