

## IMPLEMENTAÇÃO DE ESTRUTURAS DE DADOS

A implementação segue em c/c++ e foi desenvolvida usando a API Dev-C++. Optei pela utilização de duas Estruturas de Dados diferentes, Árvore AVL e Listas.

Foi utilizada na implementação a Estrutura de Árvore AVL para inserção e busca, de forma que a árvore é varrida por completa retornando os valores, não utilizando totalizadores como sugerido. Conforme observado na literatura a complexidade da AVL é  $O(\log n)$ . Através da recursividade foi tratada a Orientação a Objeto.

Foi utilizada também a Estrutura de Listas, com inserção no início, a complexidade no pior caso será de  $O(n)$ , optei por não repetir a mesma estrutura (AVL), por ter percebido, quando conversamos em sua sala o caráter pedagógico da atividade, entendo que a atividade proposta seria também para que pudéssemos observar e ampliar o leque de conhecimento sobre diferentes Estruturas de Dados e a possibilidade de combina las, por isso implementei uma segunda Estrutura, no caso a Lista.

Não utilizei outras Estruturas como Hash, pois apesar das tabelas hash terem em média tempo constantes de busca. Avaliar uma boa função de espalhamento pode levar tempo e como este foi o último conteúdo a ser trabalhado em sala, ficou muito curto o tempo (*deadline*). Na maioria dos casos soluções mais simples como uma lista encadeada podem e devem ser levadas em consideração. Pois como pode ser observado na literatura os dados na memória ficam aleatoriamente distribuídos, o que também causa *overhead* no sistema. Também é relevante o tempo gasto na depuração e remoção de erros é maior do que nas árvores AVL, que também podem ser levadas em conta para a solução de problemas do mesmo tipo.

A ideia da implementação realizada permite acesso eficiente para todos os tipos de expressões através da inserção de conjuntos. Também cada resumo de venda da

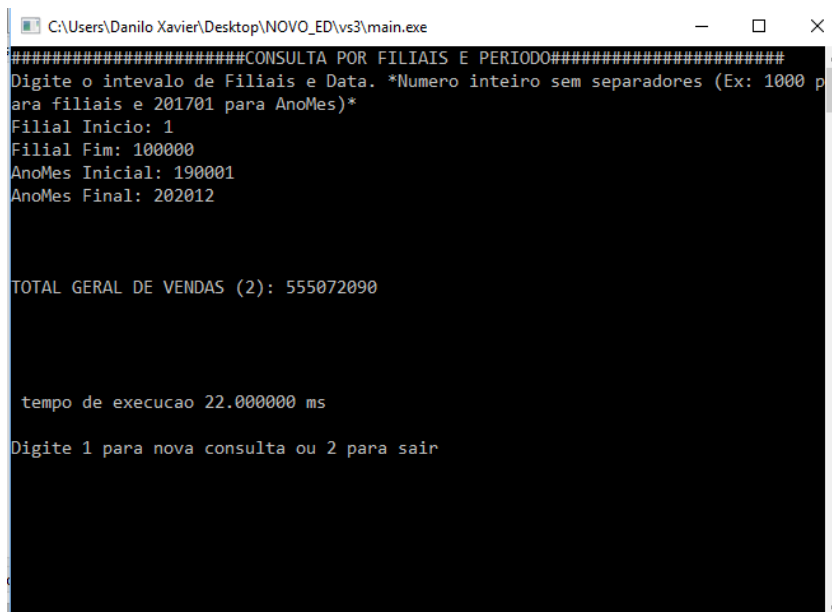
**UFF - Universidade Federal Fluminense**  
**Instituto de Computação**  
**mação do Curso de Pós-Graduação em Computação**  
**Disciplina de Estrutura de Dados (2017.1)**  
**Docente: Prof. Dr. Luiz André Portes Paes Leme**  
**Discente: Danilo Domingos Xavier**

origem a somente uma instância. E ainda atende a grandes intervalos entre Filiais, AnoMes e Total de Venda.

Segue junto a implementação o arquivo .TXT que foi utilizado para testes na implementação, ele obedece ao formato proposto no enunciado da atividade proposta e contém uma quantidade relevante de dados, o que demonstra a real eficiência da implementação.

Em teste realizados em arquivos de texto com mais de 500.000 linhas o tempo de execução foi de pouco mais de 22 milissegundos. Conforme as imagens abaixo.

```
554170 550, 201209, 038, 1000
554171 300, 201408, 054, 1000
554172 800, 199006, 032, 1000
554173 1, 200003, 098, 1000
554174 1000, 202004, 007, 1000
554175 1000, 201809, 076, 1000
554176 200, 201509, 099, 1000
554177 500, 201608, 345, 1000
554178 700, 201703, 021, 1000
554179 100, 198612, 045, 1000
554180 550, 201209, 038, 1000
554181 300, 201408, 054, 1000
554182 800, 199006, 032, 1000
```



```
C:\Users\Danilo Xavier\Desktop\NOVO_ED\vs3\main.exe
#####CONSULTA POR FILIAIS E PERIODO#####
Digite o intervalo de Filiais e Data. *Numero inteiro sem separadores (Ex: 1000 p
ara filiais e 201701 para AnoMes)*
Filial Inicio: 1
Filial Fim: 100000
AnoMes Inicial: 190001
AnoMes Final: 202012

TOTAL GERAL DE VENDAS (2): 555072090

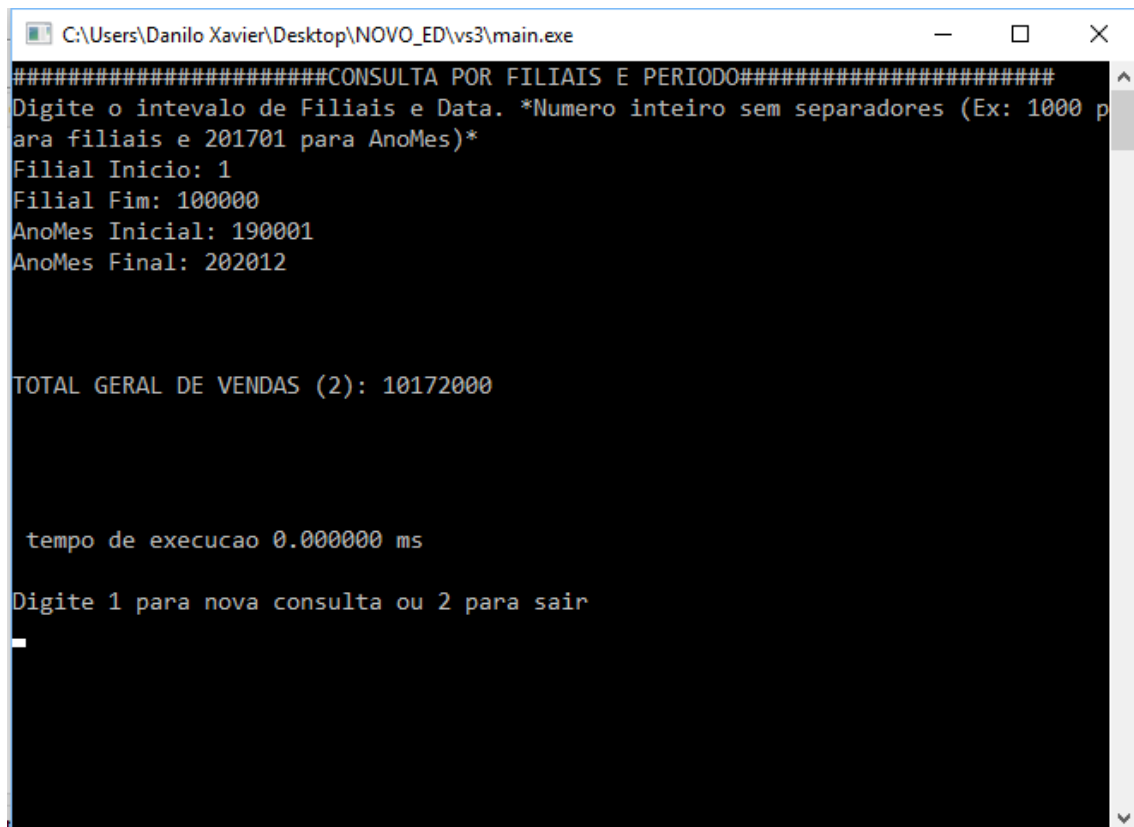
tempo de execucao 22.000000 ms

Digite 1 para nova consulta ou 2 para sair
```

**UFF - Universidade Federal Fluminense**  
**Instituto de Computação**  
**mação do Curso de Pós-Graduação em Computação**  
**Disciplina de Estrutura de Dados (2017.1)**  
**Docente: Prof. Dr. Luiz André Portes Paes Leme**  
**Discente: Danilo Domingos Xavier**

Já em outro teste com pouco mais de 10.000 linhas o tempo de execução foi de menos de 1 milissegundo. Conforme imagens abaixo.

```
10162 1, 200003, 098, 1000
10163 1000, 202004, 007, 1000
10164 1000, 201809, 076, 1000
10165 200, 201509, 099, 1000
10166 500, 201608, 345, 1000
10167 700, 201703, 021, 1000
10168 100, 198612, 045, 1000
10169 550, 201209, 038, 1000
10170 300, 201408, 054, 1000
10171 800, 199006, 032, 1000
10172 1, 200003, 098, 1000
10173 1000, 202004, 007, 1000
```



```
C:\Users\Danilo Xavier\Desktop\NOVO_ED\vs3\main.exe
#####CONSULTA POR FILIAIS E PERIODO#####
Digite o intervalo de Filiais e Data. *Numero inteiro sem separadores (Ex: 1000 p
ara filiais e 201701 para AnoMes)*
Filial Inicio: 1
Filial Fim: 100000
AnoMes Inicial: 190001
AnoMes Final: 202012

TOTAL GERAL DE VENDAS (2): 10172000

tempo de execucao 0.000000 ms

Digite 1 para nova consulta ou 2 para sair
_
```

Assim atestando a eficiência da implementação.