

Aluno: Diego Rockenbach

Os testes realizados variaram o *splitfactor* entre 0.1, 0.5 e 0.8, e o *sortedListSize* e o *shuffle* entre 0 e 1000, alternadamente (como uma balança).

Assim, os cenários testados e seus respectivos resultados foram os seguintes:

Analisando os resultados mostrados ao lado, concluímos que um *splitfactor* mais alto (como 0.8) é ideal pra inserções ordenadas ou parcialmente ordenadas, pois reduz a fragmentação, e seu desempenho cai conforme aumentam as inserções aleatórias. Enquanto isso, um *splitfactor* menor (como 0.1) performa melhor em cenários com muitas inserções aleatórias pois ele evita grandes blocos de estouro.

Assim, para inserções desordenadas um *splitfactor* menor é mais recomendado para inserções predominantemente aleatórias ou desordenadas, enquanto um *splitfactor* mais próximo de 1 tem uma

performance melhor para inserções predominantemente ordenadas. Portanto, se as inserções inicialmente forem ordenadas e depois passarem a ser aleatórias, ou o contrário, um *splitfactor* intermediário (como 0.5) é o ideal para manter a performance nas operações.

```
sorted 0; shuffle 1000;
0.1: blocks write: 2483 - blocks count 381
0.5: blocks write: 1825 - blocks count 210
0.8: blocks write: 1800 - blocks count 202

sorted 200; shuffle 800;
0.1: blocks write: 2215 - blocks count 357
0.5: blocks write: 1769 - blocks count 210
0.8: blocks write: 1768 - blocks count 203

sorted 500; shuffle 500;
0.1: blocks write: 2536 - blocks count 510
0.5: blocks write: 1679 - blocks count 212
0.8: blocks write: 1693 - blocks count 199

sorted 800; shuffle 200;
0.1: blocks write: 3386 - blocks count 796
0.5: blocks write: 1796 - blocks count 266
0.8: blocks write: 1566 - blocks count 182

sorted 1000; shuffle 0;
0.1: blocks write: 3980 - blocks count 994
0.5: blocks write: 1994 - blocks count 332
0.8: blocks write: 1598 - blocks count 200
```