Implementação de Readers e Writers - Relatório de Avaliação do Sistema

Integrantes:
- Henrique Costa Dionísio - Nº USP: 13673106

Metodologia

Base de dados

- Arquivo `bd.txt` carregado em memória (uma palavra por linha).
- Toda a base é tratada como uma única região crítica.

Carga de trabalho por thread

- Em cada execução são criadas **100 threads**, com uma proporção fixa de leitores e escritores.
- **Leitor**: 100 leituras em posições aleatórias + sleep(1ms) ainda dentro da região crítica.
- **Escritor**: 100 escritas da palavra "MODIFICADO" em posições aleatórias + sleep(1 ms) ainda dentro da região crítica.

Implementações avaliadas

- **ReadersWriters**: vários leitores podem acessar simultaneamente. Escritores esperam até não haver leitores nem outro escritor (prioridade para leitores).
- **ExclusivoSimples**: um único lock que bloqueia a base para qualquer thread (leitora ou escritora).

Varredura de proporções e medições

- Para cada implementação, foram testadas todas as proporções de **0 leitores / 100 escritores** até **100 leitores / 0 escritores** (101 proporções).
- Para cada proporção, o sistema foi executado **50 vezes**.
- Em cada repetição, mediu-se o tempo entre o início (start) das threads e o término da última (join).
- Para cada combinação (implementação, leitores, escritores), calculou-se o tempo médio em milissegundos.

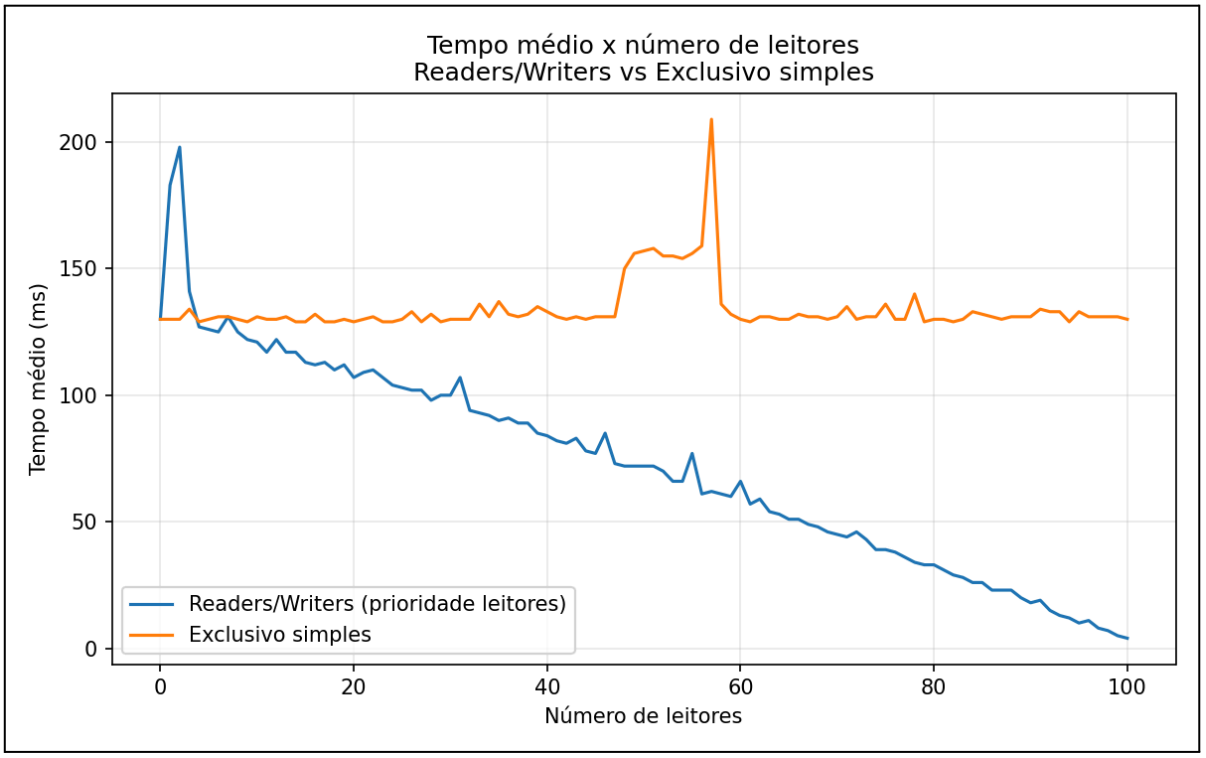
Registro dos resultados

- Os tempos médios foram gravados em “resultados_medias.csv”.

- O gráfico “grafico_comparacao.png” foi gerado a partir desses dados.

Resultados

Gráfico (`grafico_comparacao.png`)



Tempo médio × número de leitores, comparando **ReadersWriters** e **ExclusivoSimples**.

Tabela (`resultados_medias.csv`)

Leitores	Escritores	Tempo médio RW (ms)	Tempo médio Exclusivo (ms)
0	100	130	130
1	99	183	130
2	98	198	130
3	97	141	134

4	96	127	129
5	95	126	130
6	94	125	131
7	93	131	131
8	92	125	130
9	91	122	129
10	90	121	131
11	89	117	130
12	88	122	130
13	87	117	131
14	86	117	129
15	85	113	129
16	84	112	132
17	83	113	129
18	82	110	129
19	81	112	130
20	80	107	129
21	79	109	130
22	78	110	131
23	77	107	129
24	76	104	129
25	75	103	130
26	74	102	133

27	73	102	129
28	72	98	132
29	71	100	129
30	70	100	130
31	69	107	130
32	68	94	130
33	67	93	136
34	66	92	131
35	65	90	137
36	64	91	132
37	63	89	131
38	62	89	132
39	61	85	135
40	60	84	133
41	59	82	131
42	58	81	130
43	57	83	131
44	56	78	130
45	55	77	131
46	54	85	131
47	53	73	131
48	52	72	150
49	51	72	156

50	50	72	157
51	49	72	158
52	48	70	155
53	47	66	155
54	46	66	154
55	45	77	156
56	44	61	159
57	43	62	209
58	42	61	136
59	41	60	132
60	40	66	130
61	39	57	129
62	38	59	131
63	37	54	131
64	36	53	130
65	35	51	130
66	34	51	132
67	33	49	131
68	32	48	131
69	31	46	130
70	30	45	131
71	29	44	135
72	28	46	130

73	27	43	131
74	26	39	131
75	25	39	136
76	24	38	130
77	23	36	130
78	22	34	140
79	21	33	129
80	20	33	130
81	19	31	130
82	18	29	129
83	17	28	130
84	16	26	133
85	15	26	132
86	14	23	131
87	13	23	130
88	12	23	131
89	11	20	131
90	10	18	131
91	9	19	134
92	8	15	133
93	7	13	133
94	6	12	129
95	5	10	133

96	4	11	131
97	3	8	131
98	2	7	131
99	1	5	131
100	0	4	130

Conclusões

Readers/Writers: quando vale a pena

- Poucos leitores (até ~2–3): desempenho igual ou pior que o **ExclusivoSimples** (predomínio de escritores, pouco ganho com o protocolo).
- A partir de ~4 leitores: em geral passa a ser mais rápido, pois vários leitores acessam a região crítica em paralelo.
- Muitos leitores (≥ 50): ganho muito claro. O tempo médio cai para poucas dezenas de ms (chegando a ~4 ms com 100 leitores), enquanto o **ExclusivoSimples** fica em torno de 130 ms.

ExclusivoSimples: quando é suficiente

- Muitos escritores e poucos leitores: desempenho semelhante ao **Readers/Writers**, com implementação mais simples.
- Em cenários dominados por escrita, o **Readers/Writers** não traz ganho relevante e pode até introduzir algum overhead.

Resumo

- Leitura predominante: **Readers/Writers** é a melhor escolha.
- Escrita predominante: as duas implementações se equivalem em desempenho, a escolha pode ser feita pela simplicidade do **ExclusivoSimples**.