TUNING E PERFORMANCE DE BANCOS DE DADOS

Responsável: João Pedro Toledo Gonçalves

Data: 26/01/2026

MANUAL TÉCNICO - TUNING E PERFORMANCE DE BANCOS DE DADOS

Código: ITGENG 0020/26 | **Classificação:** CONFIDENCIAL
Responsável: João Pedro Toledo Gonçalves | **Data:** 26/01/2026

1. HISTÓRICO DE REVISÃO

Data	Versão	Descrição	Autor
26/01/2026	**1.0**	**Criação Inicial**	**João Pedro Toledo Gonçalves**

2. OBJETIVO

Guiar a identificação de gargalos (Slow Queries) e aplicar otimizações seguras (Universais) e agressivas (Tailored) para PostgreSQL, MySQL e Redis.

---

3. PREPARAÇÃO PROATIVA (MONITORAMENTO)

Não espere o problema acontecer. Ative isso AGORA.

1. [x] **MySQL Slow Query Log:**

• No `my.cnf`: `slow_query_log = 1`, `long_query_time = 2` (segundos).

2. [x] **Postgres pg_stat_statements:**

• Instale a extensão: `CREATE EXTENSION pg_stat_statements;`

• Veja as queries mais lentas: `SELECT * FROM pg_stat_statements ORDER BY total_exec_time DESC LIMIT 5;`

3. [x] **Redis Latency Monitor:**

• Config: `CONFIG SET latency-monitor-threshold 100` (ms).

---

4. OTIMIZAÇÕES UNIVERSAIS (SEGURO PARA TODOS)

Alterações de Sistema Operacional (Linux) que beneficiam qualquer DB.

1. Swappiness (Evitar Swap)

Bancos de dados odeiam Swap. O padrão do Linux (60) é ruim.

- **Ação:** Mude para 1 ou 10.
- Arquivo `/etc/sysctl.conf`:

```
vm.swappiness = 10
```

2. Transparent Huge Pages (THP) - DESATIVAR

O THP causa picos de lag no Redis e MongoDB.

- **Ação:** Desative no boot (varie conforme distro, geralmente via GRUB ou Systemd service).
- Verifique: `cat /sys/kernel/mm/transparent_hugepage/enabled` (Deve estar `[never]`).

---

5. TUNING TAILORED (CENÁRIOS ESPECÍFICOS)

Cenário A: Hardware com MUITA RAM (Cache Heavy)

O objetivo é manter o [Working Set] na RAM.

- **Postgres (`shared_buffers`):** Defina como 25% a 40% da RAM Total.
- **MySQL (`innodb_buffer_pool_size`):** Defina como 70-80% da RAM Total (Se o servidor for dedicado apenas ao DB).

Cenário B: Aplicação Write-Heavy (Muita Escrita)

O gargalo é o disco (I/O).

- **Postgres:** Aumente `max_wal_size` (ex: 4GB) para reduzir checkpoints frequentes.
- **Redis:** Evite `AOF everysec` se puder tolerar perda de 1s. Use persistência mista (RDB + AOF).

---

6. DIAGNÓSTICO DE QUERIES (EXPLAIN)

O desenvolvedor diz que "o banco está lento". Prove que é a query.

1. [x] Pegue a query lenta no log.
2. [x] Rode com `EXPLAIN` (MySQL) ou `EXPLAIN ANALYZE` (Postgres).

```
EXPLAIN ANALYZE SELECT * FROM pedidos WHERE data < '2023-01-01';
```

3. [x] **Analise:**

- **Seq Scan:** Leu a tabela inteira do inicio ao fim. **RUIM**. Falta índice.
- **Index Scan:** Usou o índice. **BOM**.

7. VALIDAÇÃO FINAL

[ ] O Swappiness está baixo (1-10)?
[ ] Slow Query Log está capturando consultas lentas?

[ ] Índices foram criados para eliminar "Seq Scans" críticos?