

FIAP

TECNÓLOGO EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

Gabriel Lima Silva - RM 556639

Cauã Marcelo Da Silva Machado - RM 558024

Marcos Ramalho - RM 554611

MASTERING RELATIONAL AND NON-RELATIONAL DATABASE

São Paulo – SP

2025

FIAP

TECNÓLOGO EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

Gabriel Lima Silva - RM 556639

Cauã Marcelo Da Silva Machado - RM 558024

Marcos Ramalho - RM 554611

MASTERING RELATIONAL AND NON-RELATIONAL DATABASE

Trabalho apresentado à disciplina
Masterning Relational And Non-
Relational Database Curso Tecnólogo em
Análise e Desenvolvimento de Sistemas da
FIAP - Campus Paulista, como requisito
parcial para a obtenção de nota.

Professor: Vergílio Valério dos
Santos

São Paulo – SP

2025

SUMARIO

1. Resumo Executivo.....	1
2. Descrição do Projeto.....	1
3. Análise de Mercado.....	1
4. Análise da Concorrência	1
5. Proposta de Valor.....	1
6. Estratégia de Marketing	1
7. Operações	1
8. Plano Financeiro	1
9. Equipe	1
10. Precificação	1
11. Acordo de Nível de Serviço (SLA)	1
12. Funcionalidades	1
13. Tecnologias Utilizadas	1
14. Referências	1
15. Conclusão	1
16. SPRINT 04 – VISÃO GERAL E ARQUITETURA.....	1
17. ORACLE SQL DEVELOPER – BASE CHALLENGER	1
17.1 RESET IDEMPOTENTE DO SCHEMA.....	1
17.2 DDL (TABELAS + ÍNDICES).....	1
17.3 CARGA CONTROLADA.....	1
17.4 PACKAGE PKG_CH (REGRAS).....	1
17.5 WRAPPERS E TRIGGERS	1
17.6 EXPORT JSON MANUAL (SEM FUNÇÕES NATIVAS)	1
18. POWERSHELL – CONVERSÃO DO JSON DO ORACLE	2

PASTA DE TRABALHO.....	3
LÊ E DESSERIALIZA O JSON DO SQL DEVELOPER.....	3
MONTA UM ARRAY JSON PURO E VALIDA.....	3
19. MONGODB – IMPORT, MODELAGEM E VALIDAÇÃO	3
19.1 IMPORT PARA STAGING (45 DOCS).....	3
DESCOBRE MONGOIMPORT	4
IMPORTA ARRAY JSON PURO	4
19.2 MONGO-SETUP.JS – SCHEMA VALIDATOR + AGREGAÇÃO	4
19.3 FIX-ACENTOS.JS – CORREÇÃO DE ACENTUAÇÃO	6
20. MONGODB COMPASS – EVIDÊNCIAS VISUAIS	8
21. AUTOMAÇÃO – EXECUÇÃO FIM-A-FIM (POWERSHELL)	8
RUN-SPRINT4.PS1 (RESUMO COMENTADO).....	9
1) CHECAGENS.....	9
2) CONVERSÃO DO JSON	9
3) IMPORT STAGING	9
4) SETUP + FIX + QUERIES.....	9
22. CONCLUSÃO TÉCNICA DA SPRINT 4	9
0.1) VERIFIQUE SE O SERVIÇO MONGODB ESTÁ ATIVO.....	11
0.2) VERSÕES DO CLIENTE	11
0.3) LOCALIZAR MONGOIMPORT.EXE (DATABASE TOOLS)	11
0.4) PASTA DE TRABALHO	11
1.1) LER O ARQUIVO DO ORACLE E TRANSFORMAR EM OBJETO.....	11
1.2) EXTRAIR APENAS A COLUNA "LINHA" E FILTRAR OS OBJETOS JSON	11
1.3) MONTAR O ARRAY JSON VÁLIDO E SALVAR EM UTF-8.....	11
1.4) SANITY CHECK	11
2.1) IMPORTAR COMO JSON ARRAY	11
2.2) CONFERIR.....	11

3.2) SETUP	11
CONTAGENS	11
EXEMPLO DE OBSERVAÇÕES CORRIGIDAS, MOTO 6 E MOTO 1	11
DENTRO DE C:\USERS\MARCOS RAMALHO\DOCUMENTS\MONGO.....	11

1. Resumo Executivo

Nome da Empresa: LorArch Solutions

Missão:

Fornecer uma plataforma inteligente e integrada para a localização e gestão de motocicletas em armazéns, utilizando tecnologias avançadas como LoRa e RFID, garantindo eficiência operacional, redução de custos e melhoria da experiência logística.

Visão:

Ser referência nacional e internacional em soluções inovadoras para gestão logística de frotas, promovendo agilidade, segurança e sustentabilidade por meio da tecnologia.

Objetivo Geral:

Desenvolver um sistema eficiente para localização, monitoramento e controle de motocicletas em ambientes logísticos, facilitando a tomada de decisões e otimizando processos.

Objetivos Específicos:

- Integrar tecnologias LoRa e RFID para localização precisa.
- Automatizar o registro de movimentações, defeitos e manutenções.
- Fornecer uma interface simples, intuitiva e acessível para usuários de diferentes níveis.
- Garantir segurança, integridade e escalabilidade dos dados.

2. Descrição do Projeto

O Projeto LorArch combina o alcance e baixo consumo da tecnologia LoRa com a precisão da RFID para rastrear motocicletas em armazéns. Equipamentos embarcados em cada motocicleta transmitem sua posição via LoRa para gateways instalados estrategicamente, enquanto leitores RFID capturam a passagem das motos em pontos fixos.

A plataforma central processa os dados e os disponibiliza por meio de uma API RESTful, consumida por uma aplicação web responsiva que permite aos operadores:

- Consultar a localização e o status das motos em tempo real.
- Visualizar históricos detalhados de movimentações e manutenções.
- Acionar alarmes sonoros para facilitar a localização física imediata.

Essa arquitetura promove alta confiabilidade, baixo custo operacional e facilidade de expansão para múltiplos ambientes.

3. Análise de Mercado

O mercado de logística e armazenamento passa por rápida transformação digital, demandando soluções conectadas para otimizar o controle de frotas e reduzir perdas, especialmente em setores com alta mobilidade, como o das motocicletas.

Oportunidades:

- Expansão acelerada da logística urbana e e-commerce.
- Pressão crescente para redução de custos e minimização de erros operacionais.
- Popularização e evolução das tecnologias IoT.

Tendências:

- Crescente adoção de LPWANs como LoRa e NB-IoT para comunicação eficiente em longas distâncias.
- Integração robusta de RFID para rastreamento de alta precisão em ambientes internos.
- Interfaces intuitivas que demandam mínimo treinamento para usuários finais.

4. Análise da Concorrência

Embora existam sistemas que utilizam GPS e RFID separadamente, poucos oferecem uma solução integrada para ambientes internos de grande escala com comunicação eficiente.

Concorrentes Diretos:

- Sistemas GPS voltados para veículos externos e grandes frotas.
- Soluções RFID com leitores isolados e sem integração dinâmica.

Diferenciais do LorArch:

- Integração inédita entre LoRa e RFID para ambientes internos.
- Plataforma web responsiva e API moderna para fácil integração.
- Custos operacionais reduzidos e alto potencial de escalabilidade.

5. Proposta de Valor

- Rastreamento rápido e preciso das motocicletas.
- Registro confiável e automatizado de movimentações, defeitos e manutenções.
- Suporte à tomada de decisão baseada em dados reais e atualizados.
- Redução expressiva no tempo de busca e retrabalho operacional.

6. Estratégia de Marketing

Canais:

- Campanhas digitais nas principais plataformas (Google Ads, LinkedIn, Instagram).
- Parcerias estratégicas com transportadoras e operadores logísticos.
- Participação ativa em eventos, feiras e conferências do setor.

Aquisição de Clientes:

- Oferecimento de pilotos gratuitos para demonstração do sistema.
- Programas de indicação e fidelização.
- Desenvolvimento de materiais educativos e divulgação de cases de sucesso.

7. Operações

Modelo Operacional:

- Plataforma hospedada em nuvem com alta disponibilidade e escalabilidade.
- Suporte técnico remoto qualificado e processos de atendimento ágeis.
- Atualizações contínuas com base em feedback dos usuários e análise de uso.

Manutenção:

- Monitoramento em tempo real da plataforma e dos dispositivos IoT.
- Rotinas periódicas de atualização de software e firmware dos equipamentos.

8. Plano Financeiro

Estimativas de Custos:

- Desenvolvimento do software e infraestrutura em nuvem.
- Aquisição, instalação e manutenção dos equipamentos LoRa e RFID.
- Investimento em marketing, vendas e suporte.

Investimentos Estratégicos:

- Contratação e capacitação de equipe técnica especializada.
- Equipamentos IoT e servidores para garantia de desempenho e disponibilidade.

9. Equipe

- CEO: Liderança estratégica e gestão corporativa.
- CTO: Desenvolvimento tecnológico e inovação.
- Gerente de Operações: Coordenação de implantação e suporte técnico.
- Equipe Técnica: Desenvolvimento, testes, manutenção e atendimento.

10. Precificação

Modelo escalável por assinatura mensal, baseado na quantidade de unidades monitoradas e funcionalidades contratadas:

- Plano Básico: Monitoramento e consultas simples.
- Plano Avançado: Inclusão de relatórios detalhados, alertas e gestão de manutenções.
- Plano Premium: Dashboards avançados, integração com outros sistemas e análises preditivas.

11. Acordo de Nível de Serviço (SLA)

- Garantia de disponibilidade mínima de 99%.
- Atendimento ao cliente com resposta em até 24 horas úteis.
- Resolução de incidentes críticos em até 48 horas.

12. Funcionalidades

- Rastreamento em tempo real via LoRa e RFID com alta precisão.
- Histórico completo e auditável das movimentações e manutenções.
- Alarmes sonoros para localização rápida das motos.
- Interface web amigável e API RESTful para integração.
- Relatórios e dashboards personalizáveis conforme necessidade do cliente.

13. Tecnologias Utilizadas

- Dispositivos móveis com Arduino e módulos LoRa.
- Leitores RFID instalados em pontos fixos estratégicos.
- Gateways LoRa baseados em Raspberry Pi.
- Backend desenvolvido em Java com Spring Boot.
- Banco de Dados Oracle para alta confiabilidade e performance.
- Frontend responsivo usando React ou Angular.
- Hospedagem em nuvem (Azure ou AWS) para escalabilidade e segurança.

14. Referências

- Relatórios recentes do setor de logística e cadeia de suprimentos.
- Pesquisas acadêmicas e de mercado sobre IoT e LPWAN.
- Casos práticos de implantação em transportadoras e centros de distribuição.

15. Conclusão

O Projeto LorArch oferece uma solução tecnológica avançada, alinhada com as tendências de mercado, capaz de transformar a gestão logística de motocicletas. Ao integrar LoRa e RFID em uma plataforma eficiente, proporciona agilidade, segurança e redução de custos, atendendo às demandas atuais e futuras das empresas.

16. SPRINT 04 – VISÃO GERAL E ARQUITETURA

Objetivo. Construir um pipeline reproduzível Oracle → JSON → MongoDB, mantendo regras de negócio (auditoria, validações, somas) e gerando evidências no MongoDB Compass.

Fluxo resumido.

1. Oracle SQL Developer: cria schema Challenger (tabelas CH_*), carrega 15 setores, 15 motos, 45 movimentações, aplica packages/triggers, e gera JSON manual em CH_JSON_OUT → ch_movimentos.json.
2. PowerShell: reconstrói o JSON do SQL Developer no formato array JSON puro → ch_movimentos_corrigido.json.
3. MongoDB: mongoimport → coleção staging (45 docs); script mongo-setup.js agrupa por id_moto e cria a coleção motos (3 docs: 1, 6, 11) com schema validator e índices; script fix-acentos.js corrige acentuação.
4. MongoDB Compass: validação visual (docs, índices, pipelines) e prints.

Critérios de aceite.

- staging.countDocuments() = 45
- motos.countDocuments() = 3 (motos 1, 6, 11)
- Totais esperados (amostra): id_moto 6 = 1740, id_moto 1 = 870, id_moto 11 = 600
- Índices criados: id_moto_sql (unique), "movimentacoes.setor", total_custos

17. ORACLE SQL DEVELOPER – BASE CHALLENGER

17.1 RESET IDEMPOTENTE DO SCHEMA

PL/SQL que desabilita/dropa triggers, views, procs, funcs, packages, types, sequences e tables em qualquer ordem, evitando erros e permitindo reexecução limpa. Benefícios: elimina resíduos de execuções anteriores e garante reproduzibilidade.

Trecho ilustrativo – [CÓDIGO 1: bloco DECLARE...BEGIN...END; de reset]

17.2 DDL (TABELAS + ÍNDICES)

Tabelas criadas:

- CH_SETOR (id, nome, ativo, unique por nome, check de ativo)
- CH_MOTOCICLETA (id, placa, modelo, status, data; unique placa; check de status)
- CH_MOVIMENTACAO (id, id_moto, id_setor, tipo, dt, custo, obs; FKs; checks)
- CH_AUDITORIA (id, tabela, operação, id_registro, detalhe, data, usuário_bd)

Índices: IX_CH_MOV_MOTO, IX_CH_MOV_SETOR, IX_CH_MOV_DATA.

Trecho ilustrativo – [CÓDIGO 2: DDL das 4 tabelas + 3 índices]

17.3 CARGA CONTROLADA

- 15 setores (Recepção... RH)
- 15 motos (placas e modelos realistas; status variados)
- 45 movimentações em 3 blocos de 15 cada:
 - Bloco A (motos 1..5; setores 1..3; ENTRADA; custo = 50+N)
 - Bloco B (motos 6..10; setores 4..7; TRANSF; custo = 100 + 2*N)
 - Bloco C (motos 11..15; setores 8..10; SAIDA; custo = 5*N)

- Uso de CONNECT BY e ROWNUM <= 15 para travar em exatos 15 registros por bloco.

17.4 PACKAGE PKG_CH (REGRAS)

- fun_valida_tipo e fun_valida_status (ENTRADA/SAIDA/TRANSF; ATIVA/MANUTENCAO/INATIVA)
- fun_soma_custo_moto(p_id) – soma custos na CH_MOVIMENTACAO
- prc_registrar_auditoria – grava na CH_AUDITORIA
- prc_registrar_mov – valida FKs, valida tipo/custo, insere mov e audita
- prc_atualiza_status_moto – valida status e audita
- prc_rel_somas – imprime qtd e soma geral + TOP 5 por custo

Trecho ilustrativo – [CÓDIGO 3: assinatura do package e highlights de erros - 20061/-20091/-20041]

17.5 WRAPPERS E TRIGGERS

- Wrappers: PRC_CH_REGISTRAR_MOV, PRC_CH_ATUALIZA_STATUS_MOTO, PRC_CH_REL_SOMAS
- Triggers:
 - TRG_CH_MOTO_PLACA_UP: PLACA sempre maiúscula
 - TRG_CH_MOV_AUD_INS: auditoria após INSERT
 - TRG_CH_MOV_AUD_UPDDEL: auditoria UPDATE/DELETE
 - TRG_CH_MOTO_STATUS_AUD: auditoria ao mudar STATUS

17.6 EXPORT JSON MANUAL (SEM FUNÇÕES NATIVAS)

- Tabela CH_JSON_OUT(ORD, LINHA) para armazenar cada linha do JSON
- Package PKG_CH_JSON.prc_gerar_json_mov_det:
 - Abre com [, emite , entre objetos, fecha com]
 - Campos: id_moto, setor, tipo, dt_mov (ISO), obs, custo (ponto decimal via NLS_NUMERIC_CHARACTERS=.,)

Trecho ilustrativo – [CÓDIGO 4: loop que escreve cada objeto JSON em CH_JSON_OUT]

17.7 Spool do arquivo

- Opção A

```
SET PAGES 0 LINESIZE 4000 FEEDBACK OFF TRIMSPOOL ON SPOOL 'C:\Users\Marcos  
Ramalho\Documents\Mongo\ch_movimentos.json'  
SELECT LINHA FROM  
CH_JSON_OUT ORDER BY ORD; SPOOL OFF
```

- Opção B (alternativa) com SET SQLFORMAT JSON (mantendo UTF-8).

18. POWERSHELL – CONVERSÃO DO JSON DO ORACLE

O SQL Developer exporta um “wrapper” com results/items. A conversão abaixo reconstrói um array JSON puro válido para o mongoimport.

Passos (PowerShell):

PASTA DE TRABALHO

```
New-Item -ItemType Directory -Force -Path "$env:USERPROFILE\Documents\Mongo" | Out-Null Set-Location "$env:USERPROFILE\Documents\Mongo"
```

LÊ E DESSERIALIZA O JSON DO SQL DEVELOPER

```
$orig = Get-Content .\ch_movimentos.json -Raw | ConvertFrom-Json $linhas = $orig.results[0].items | ForEach-Object { $.linha } $objetos = $linhas | Where-Object { $ -match '^`s*{.}`s$' } if (-not $objetos -or $objetos.Count -eq 0) { throw "Não encontrei objetos JSON nas linhas." }
```

MONTA UM ARRAY JSON PURO E VALIDA

```
$conteudoFinal = "[`n" + ($objetos -join ",`n") + "`n]" Set-Content .\ch_movimentos_corrigido.json $conteudoFinal -Encoding utf8 $null = Get-Content .\ch_movimentos_corrigido.json -Raw | ConvertFrom-Json "OK: ch_movimentos_corrigido.json reconstruído e válido." | Write-Host
```

19. MONGODB – IMPORT, MODELAGEM E VALIDAÇÃO

19.1 IMPORT PARA STAGING (45 DOCS)

DESCOBRE MONGOIMPORT

```
$mongoimportExe = (Get-ChildItem -Path "C:\Program Files","C:\Program Files (x86)", "$env:LOCALAPPDATA","$env:LOCALAPPDATA\Programs" -Recurse -Filter mongoimport.exe -ErrorAction SilentlyContinue | Select-Object -First 1).FullName
```

IMPORTA ARRAY JSON PURO

```
& $mongoimportExe --db lorarch --collection staging --drop --file ".\ch_movimentos_corrigido.json" --jsonArray
```

Esperado: 45 document(s) imported successfully.

19.2 MONGO-SETUP.JS – SCHEMA VALIDATOR + AGREGAÇÃO

Cria motos com validator e popula via aggregate a partir de staging, agrupando por id_moto e somando custos.

```
// mongo-setup.js use("lorarch");

db.motos.drop();

db.createCollection("motos", { validator: { $jsonSchema: { bsonType: "object", required: ["id_moto_sql","movimentacoes","total_custos"], properties: { id_moto_sql: { bsonType: "int" }, placa:{ bsonType: ["string","null"] }, modelo:{ bsonType: ["string","null"] }, status:{ bsonType: ["string","null"] }, movimentacoes: { bsonType: "array", items: { bsonType: "object", required: ["setor","tipo","dt_mov","obs","custo"], properties: { setor:{ bsonType:"string" }, tipo: { bsonType:"string", enum:["ENTRADA","TRANSF","SAIDA"] }, dt_mov:{ bsonType:"string" }, obs: { { bsonType:"string" } }, custo:{ bsonType:"string" } } } } } } } })
```

```

bsonType:["double","int","decimal"]      }      }      },      total_custos:      {
bsonType:["double","int","decimal"] } } });

db.motos.insertMany( db.staging.aggregate([ { $group: { _id: "$id_moto",
movimentacoes: { $push: { setor: "$setor", tipo: "$tipo", dt_mov: "$dt_mov", obs: {
$ifNull: ["$obs","", ] }, custo: { $toDouble: "$custo" } }}, total_custos: { $sum: { $toDouble:
"$custo" } } }}, { $project: { _id: 0, id_moto_sql: { $toInt: "$_id" }, placa: null, modelo: null,
status: null, movimentacoes: 1, total_custos: 1 } } ]).toArray() );

db.motos.createIndex({ id_moto_sql: 1 }, { unique: true });
db.motos.createIndex({ "movimentacoes.setor": 1 });
db.motos.createIndex({ total_custos: -1 });

```

Execução (mongosh):

```
mongosh "mongodb://localhost:27017/lorarch" --file ".\mongo-setup.js"
```

Esperado: Docs em staging: 45 e Docs em motos: 3.

19.3 FIX-ACENTOS.JS – CORREÇÃO DE ACENTUAÇÃO

Atualiza cada item de movimentacoes com \$map e \$replaceAll (ex.: “padrÃ£o” → “padrão”; “TransferÃªncia” → “Transferência”).

```

// fix-acentos.js

use("lorarch");

db.motos.updateMany()

```

```
{},  
[  
{  
$set:{  
movimentacoes:{  
$map:{  
input: "$movimentacoes",  
as: "m",  
in:{  
$mergeObjects:[  
"$$m",  
{  
obs:{  
$replaceAll:{  
input:{  
$replaceAll:{  
input: " $$m.obs",  
find: "padrÃ£o", replacement: "padrão"
```

```
        }

    },
    find: "TransferÃ¡ncia", replacement: "Transferênci"

    }
}

}

]

}

}

}

}

}

}

]

);

};
```

Execução:

```
mongosh "mongodb://localhost:27017/lorarch" --file ".\fix-acentos.js"
```

19.4 mongo-queries.js – consultas de validação

```
// mongo-queries.js use("lorarch");
```

```

print("TOP      5      por      total_custos:");    printjson(      db.motos.find({},
{id_moto_sql:1,total_custos:1,_id:0}) .sort({total_custos:-1}).limit(5).toArray() );

      print("\nUm exemplo com setor = 'Entrega':"); printjson( db.motos.find({
'movimentacoes.setor': 'Entrega' }, { id_moto_sql:1, 'movimentacoes.$':1, _id:0 })
.limit(5).toArray() );

      print("\nFiltro   por   período   2025-10-01   ..   2025-10-10:"); printjson(
db.motos.aggregate([ { $unwind: '$movimentacoes' }, { $addFields: { mData: {
$dateFromString: { dateString: '$movimentacoes.dt_mov' } } } }, { $match: { mData: {
$gte: ISODate('2025-10-01T00:00:00Z'), $lte: ISODate('2025-10-10T23:59:59Z') } } }, { $project: { _id:0, id_moto_sql:1, setor:'$movimentacoes.setor',
tipo:'$movimentacoes.tipo', dt:'$movimentacoes.dt_mov', custo:'$movimentacoes.custo' } }, { $limit: 10 } ]).toArray() );

```

Execução:

```
mongosh "mongodb://localhost:27017/lorarch" --file ".\mongo-queries.js"
```

Esperado (amostras):

- TOP 5: {6: 1740}, {1: 870}, {11: 600}
- Setor 'Entrega': um doc da moto 6 com uma movimentação TRANSF
- Período 01–10/out/2025: registros da moto 11 (SAIDA)

20. MONGODB COMPASS – EVIDÊNCIAS VISUAIS

Conexão: mongodb://localhost:27017, DB lorarch.

Coleções: staging, motos.

21. AUTOMAÇÃO – EXECUÇÃO FIM-A-FIM (POWERSHELL)

- Script consolidado que:
- Checa serviço do MongoDB e versões do mongosh/mongoimport.
- Converte ch_movimentos.json → ch_movimentos_corrigido.json.
- Importa em staging e monta motos (setup + fix acentos).
- Executa queries e salva log.

Arquivo: run-sprint4.ps1

Execução: no PowerShell, dentro da pasta:

.\run-sprint4.ps1

RUN-SPRINT4.PS1 (RESUMO COMENTADO)

```
$ErrorActionPreference = "Stop" $work = "$env:USERPROFILE\Documents\Mongo" $log = Join-Path $work ("sprint4_log_{0}.txt") -f (Get-Date -Format "yyyyMMdd_HHmmss")  
"== SPRINT 4 RUN $(Get-Date) ==" | Tee-Object -FilePath $log
```

1) CHECAGENS

```
(Get-Service MongoDB).Status | Tee-Object -FilePath $log -Append &  
"$env:LOCALAPPDATA\Programs\mongosh\mongosh.exe" --version | Tee-Object -  
FilePath $log -Append $mongoimportExe = (Get-ChildItem -Path "C:\Program  
Files","C:\Program Files (x86)",  
"$env:LOCALAPPDATA","$env:LOCALAPPDATA\Programs" -Recurse -Filter  
mongoimport.exe -ErrorAction SilentlyContinue | Select-Object -First 1).FullName  
"mongoimport: $mongoimportExe" | Tee-Object -FilePath $log -Append
```

2) CONVERSÃO DO JSON

```
$orig = Get-Content "$work\ch_movimentos.json" -Raw | ConvertFrom-Json  
$linhas = $orig.results[0].items | ForEach-Object { $.linha } $objetos = $linhas | Where-  
Object { $ -match '^\\s*{.}\\s*$' } $conteudoFinal = "[rn" + ($objetos -join ",rn") + "rn]" Set-  
Content "$work\ch_movimentos_corrigido.json" $conteudoFinal -Encoding utf8 $null  
= Get-Content "$work\ch_movimentos_corrigido.json" -Raw | ConvertFrom-Json "OK:  
$work\ch_movimentos_corrigido.json válido." | Tee-Object -FilePath $log -Append
```

3) IMPORT STAGING

```
& $mongoimportExe --db lorarch --collection staging --drop ` --file  
"$work\ch_movimentos_corrigido.json" --jsonArray | Tee-Object -FilePath $log -  
Append & "$env:LOCALAPPDATA\Programs\mongosh\mongosh.exe"
```

```
"mongodb://localhost:27017/lorarch" --eval "db.staging.countDocuments()" | Tee-Object -FilePath $log -Append
```

4) SETUP + FIX + QUERIES

```
& "$env:LOCALAPPDATA\Programs\mongosh\mongosh.exe"  
"mongodb://localhost:27017/lorarch" --file "$work\mongo-setup.js" | Tee-Object -FilePath $log -Append & "$env:LOCALAPPDATA\Programs\mongosh\mongosh.exe"  
"mongodb://localhost:27017/lorarch" --file "$work\fix-acentos.js" | Tee-Object -FilePath $log -Append & "$env:LOCALAPPDATA\Programs\mongosh\mongosh.exe"  
"mongodb://localhost:27017/lorarch" --file "$work\mongo-queries.js" | Tee-Object -FilePath $log -Append  
  
"Script completo finalizado." | Tee-Object -FilePath $log -Append
```

22. CONCLUSÃO TÉCNICA DA SPRINT 4

A Sprint 4 entregou um pipeline funcional, idempotente e auditável, do Oracle ao MongoDB, com schema validator, agregações, índices, evidências no Compass e automação PowerShell.

O processo é reexecutável em ambiente limpo, com saídas determinísticas (45 → 3 docs) e consultas que comprovam regras e métricas de negócio (somas, filtros por setor/periódico).

Próximos passos (opcional):

- Tratar datas como Date no Mongo (converter dt_mov para ISO real em insertMany).
- Popular campos placa/modelo/status com integração futura.
- Publicar o run-sprint4.ps1 com assinatura do PowerShell/ExecutionPolicy para uso no time.

23. Script completo .SQL, Mongo (Sprint 4)

- .SQL

```
/*
=====
===== CHALLENGER 2025 – SPRINT 04

• RESET completo do schema (drop seguro de tudo
do usuário)
• DDL (tabelas, constraints, índices)
• PACKAGE de regras + WRAPPERS + TRIGGERS de
auditoria
• CARGA determinística (15 setores, 15 motos, 45
movs com datas/custos fixos)
• EXPORT JSON "linha a linha" em CH_JSON_OUT
```

• Blocos	SPOOL	prontos
=====		
===== */		
<pre> SET SERVEROUTPUT ON SET DEFINE OFF ALTER SESSION SET nls_date_format = 'YYYY-MM-DD"T"HH24:MI:SS'; PROMPT == INICIO SPRINT 04 == /* ===== [01] RESET ===== Limpa o schema do usuário ===== */ ===== */ DECLARE PROCEDURE try_exec(p_sql VARCHAR2) IS BEGIN EXECUTE IMMEDIATE p_sql; EXCEPTION WHEN OTHERS THEN NULL; END; BEGIN -- desabilita triggers FOR r IN (SELECT trigger_name FROM user_triggers) LOOP try_exec('ALTER TRIGGER' r.trigger_name ' DISABLE'); END LOOP; -- triggers FOR r IN (SELECT trigger_name FROM user_triggers) LOOP try_exec('DROP TRIGGER' r.trigger_name); END LOOP; -- views FOR r IN (SELECT view_name FROM user_views) LOOP try_exec('DROP VIEW' r.view_name); END LOOP; -- procs / funcs / packages FOR r IN (SELECT object_name FROM user_objects WHERE object_type='PROCEDURE') LOOP try_exec('DROP PROCEDURE' r.object_name); END LOOP; FOR r IN (SELECT object_name FROM user_objects WHERE object_type='FUNCTION') LOOP try_exec('DROP FUNCTION ' r.object_name); END LOOP; FOR r IN (SELECT object_name FROM user_objects WHERE object_type='PACKAGE BODY') LOOP try_exec('DROP PACKAGE BODY' r.object_name); END LOOP; FOR r </pre>		

```

IN  (SELECT object_name FROM user_objects WHERE
object_type='PACKAGE') LOOP try_exec('DROP PACKAGE
'||r.object_name); END LOOP;

-- types FOR r IN (SELECT object_name FROM user_objects
WHERE object_type='TYPE BODY') LOOP try_exec('DROP TYPE BODY
'||r.object_name); END LOOP; FOR r IN (SELECT object_name FROM
user_objects WHERE object_type='TYPE') LOOP try_exec('DROP TYPE
'||r.object_name||' FORCE'); END LOOP;

-- sequences criadas pelo usuário (mantém as ISEQ$$_ das
identidades) FOR r IN (SELECT sequence_name FROM
user_sequences WHERE sequence_name NOT LIKE 'ISEQ$$_%')
LOOP try_exec('DROP SEQUENCE'||r.sequence_name); END LOOP;

-- tables (tento com e sem aspas) FOR r IN (SELECT
table_name FROM user_tables) LOOP try_exec('DROP TABLE
'||r.table_name||'' CASCADE CONSTRAINTS PURGE');
try_exec('DROP TABLE'||r.table_name||' CASCADE CONSTRAINTS
PURGE'); END LOOP;

DBMS_OUTPUT.PUT_LINE(..reset ok'); END; / PROMPT ..reset
concluido

/*
=====
[02] DDL
=====
===== Tabelas do domínio, constraints
e
=====
===== índices
=====
===== */
CREATE TABLE CH_SETOR ( ID_SETOR NUMBER
GENERATED BY DEFAULT AS IDENTITY, NOME_SETOR VARCHAR2(60)
NOT NULL, ATIVO CHAR(1) DEFAULT 'S' NOT NULL, CONSTRAINT
PK_CH_SETOR PRIMARY KEY (ID_SETOR), CONSTRAINT

```

```

UQ_CH_SETOR_NOME UNIQUE (NOME_SETOR), CONSTRAINT
CK_CH_SETOR_ATIVO CHECK (ATIVO IN ('S','N')) );

CREATE TABLE CH_MOTOCICLETA ( ID_MOTO NUMBER
GENERATED BY DEFAULT AS IDENTITY, PLACA VARCHAR2(8) NOT
NULL, MODELO VARCHAR2(60) NOT NULL, STATUS VARCHAR2(15)
DEFAULT 'ATIVA' NOT NULL, DT_CADASTRO DATE DEFAULT SYSDATE
NOT NULL, CONSTRAINT PK_CH_MOTO PRIMARY KEY (ID_MOTO),
CONSTRAINT UQ_CH_MOTO_PLACA UNIQUE (PLACA), CONSTRAINT
CK_CH_MOTO_STATUS CHECK (STATUS IN
('ATIVA','MANUTENCAO','INATIVA')) );

CREATE TABLE CH_MOVIMENTACAO ( ID_MOV NUMBER
GENERATED BY DEFAULT AS IDENTITY, ID_MOTO NUMBER NOT
NULL, ID_SETOR NUMBER NOT NULL, TIPO_MOV VARCHAR2(10)
NOT NULL,
DT_MOV DATE NOT NULL, CUSTO NUMBER(10,2) DEFAULT 0
NOT NULL, OBSERVACAO VARCHAR2(400), CONSTRAINT
PK_CH_MOV PRIMARY KEY (ID_MOV), CONSTRAINT
FK_CH_MOV_MOTO FOREIGN KEY (ID_MOTO) REFERENCES
CH_MOTOCICLETA(ID_MOTO), CONSTRAINT FK_CH_MOV_SETOR
FOREIGN KEY (ID_SETOR) REFERENCES CH_SETOR(ID_SETOR),
CONSTRAINT CK_CH_MOV_TIPO CHECK (TIPO_MOV IN
('ENTRADA','SAIDA','TRANSF')), CONSTRAINT CK_CH_MOV_CUSTO
CHECK (CUSTO >= 0) );

CREATE TABLE CH_AUDITORIA ( ID_AUDITORIA NUMBER
GENERATED BY DEFAULT AS IDENTITY, TABELA VARCHAR2(30) NOT
NULL, OPERACAO VARCHAR2(10) NOT NULL, ID_REGISTRO
NUMBER, DETALHE VARCHAR2(4000), DATA_AUD DATE DEFAULT
SYSDATE NOT NULL, USUARIO_BD VARCHAR2(128) DEFAULT USER

```

```

NOT NULL, CONSTRAINT PK_CH_AUD PRIMARY KEY (ID_AUDITORIA)
);

CREATE INDEX IX_CH_MOV_MOTO ON
CH_MOVIMENTACAO(ID_MOTO); CREATE INDEX IX_CH_MOV_SETOR
ON CH_MOVIMENTACAO(ID_SETOR); CREATE INDEX
IX_CH_MOV_DATA ON CH_MOVIMENTACAO(DT_MOV);

PROMPT ..ddl criado

/*
=====
[03] CARGA
=====
15 setores, 15 motos, 45
movimentações (datas/custos determinísticos) Datas base: 2025-09-
29 .. 2025-10-13 (15 dias) Custos: - Moto 1 (Recepcao, ENTRADA) -
Moto 6 (Entrega, TRANSF) - Moto 11 (Comercial, SAIDA)
=====
*/
-- zera (idempotente) DELETE FROM
CH_MOVIMENTACAO; DELETE FROM CH_MOTOCICLETA; DELETE
FROM CH_SETOR; COMMIT;

-- 03.1 Setores INSERT INTO CH_SETOR (NOME_SETOR,
ATIVO) VALUES ('Recepcao''S'); INSERT INTO CH_SETOR
(NOME_SETOR, ATIVO) VALUES ('Entrega''S'); INSERT INTO
CH_SETOR (NOME_SETOR, ATIVO) VALUES ('Comercial''S'); INSERT
INTO CH_SETOR (NOME_SETOR, ATIVO) VALUES ('Financeiro''S');
INSERT INTO CH_SETOR (NOME_SETOR, ATIVO) VALUES ('Oficina''S');
INSERT INTO CH_SETOR (NOME_SETOR, ATIVO) VALUES
('Estoque''S'); INSERT INTO CH_SETOR (NOME_SETOR, ATIVO)
VALUES ('RH''S'); INSERT INTO CH_SETOR (NOME_SETOR, ATIVO)
VALUES ('Compras''S'); INSERT INTO CH_SETOR (NOME_SETOR,
ATIVO) VALUES ('TI''S'); INSERT INTO CH_SETOR (NOME_SETOR,
ATIVO) VALUES ('Juridico''S'); INSERT INTO CH_SETOR
(NOME_SETOR, ATIVO) VALUES ('Diretoria''S'); INSERT INTO

```

```

CH_SETOR (NOME_SETOR, ATIVO) VALUES ('Marketing','S'); INSERT
INTO CH_SETOR (NOME_SETOR, ATIVO) VALUES ('Qualidade','S');
INSERT INTO CH_SETOR (NOME_SETOR, ATIVO) VALUES
('Seguranca','S'); INSERT INTO CH_SETOR (NOME_SETOR, ATIVO)
VALUES ('Logistica','S');

-- 03.2 Motos INSERT INTO CH_MOTOCICLETA (PLACA,
MODELO, STATUS) VALUES ('AAA1A11','CG 160 Fan','ATIVA'); INSERT
INTO CH_MOTOCICLETA (PLACA, MODELO, STATUS) VALUES
('BBB2B22','NMax 160','ATIVA'); INSERT INTO CH_MOTOCICLETA
(PLACA, MODELO, STATUS) VALUES ('CCC3C33','Biz
125','MANUTENCAO'); INSERT INTO CH_MOTOCICLETA (PLACA,
MODELO, STATUS) VALUES ('DDD4D44','CB 300F','ATIVA'); INSERT
INTO CH_MOTOCICLETA (PLACA, MODELO, STATUS) VALUES
('EEE5E55','Fazer 250','INATIVA'); INSERT INTO CH_MOTOCICLETA
(PLACA, MODELO, STATUS) VALUES ('FFF6F66','YBR 150','ATIVA'); -- ID
6 INSERT INTO CH_MOTOCICLETA (PLACA, MODELO, STATUS)
VALUES ('GGG7G77','PCX 160','ATIVA'); INSERT INTO
CH_MOTOCICLETA (PLACA, MODELO, STATUS) VALUES
('HHH8H88','Hornet 600','MANUTENCAO'); INSERT INTO
CH_MOTOCICLETA (PLACA, MODELO, STATUS) VALUES ('III9I99','CB
Twister','ATIVA'); INSERT INTO CH_MOTOCICLETA (PLACA, MODELO,
STATUS) VALUES ('JJJ0J00','Bros 160','ATIVA'); INSERT INTO
CH_MOTOCICLETA (PLACA, MODELO, STATUS) VALUES
('KKK1K12','MT 03','ATIVA'); -- ID 11 INSERT INTO CH_MOTOCICLETA
(PLACA, MODELO, STATUS) VALUES ('LLL2L23','Factor 150','INATIVA');
INSERT INTO CH_MOTOCICLETA (PLACA, MODELO, STATUS) VALUES
('MMM3M34','Pop 110','ATIVA'); INSERT INTO CH_MOTOCICLETA
(PLACA, MODELO, STATUS) VALUES ('NNN4N45','CBR 650R','ATIVA');
INSERT INTO CH_MOTOCICLETA (PLACA, MODELO, STATUS) VALUES
('OOO5O56','XRE 300','MANUTENCAO');

```

```

-- 03.3 Movimentações determinísticas para 3 motos (1, 6, 11)

DECLARE v_base DATE := DATE '2025-09-29'; v_setor_recepcao
CH_SETOR.ID_SETOR%TYPE; v_setor_entrega
CH_SETOR.ID_SETOR%TYPE; v_setor_comercial
CH_SETOR.ID_SETOR%TYPE; BEGIN SELECT id_setor INTO
v_setor_recepcao FROM ch_setor WHERE nome_setor = 'Recepcao';
SELECT id_setor INTO v_setor_entrega FROM ch_setor WHERE
nome_setor = 'Entrega'; SELECT id_setor INTO v_setor_comercial
FROM ch_setor WHERE nome_setor = 'Comercial';

-- Moto 1 – Recepcao / ENTRADA FOR k IN 0..14 LOOP INSERT
INTO CH_MOVIMENTACAO (ID_MOTO, ID_SETOR, TIPO_MOV,
DT_MOV, CUSTO, OBSERVACAO) VALUES (1, v_setor_recepcao,
'ENTRADA', v_base + k, 65 - k, 'Entrada padrão'); END LOOP;

-- Moto 6 – Entrega / TRANSF FOR k IN 0..14 LOOP INSERT INTO
CH_MOVIMENTACAO (ID_MOTO, ID_SETOR, TIPO_MOV, DT_MOV,
CUSTO, OBSERVACAO) VALUES (6, v_setor_entrega, 'TRANSF', v_base
+ k, 130 - (2*k), 'Transferência de setor'); END LOOP;

-- Moto 11 – Comercial / SAIDA FOR k IN 0..14 LOOP INSERT
INTO CH_MOVIMENTACAO (ID_MOTO, ID_SETOR, TIPO_MOV,
DT_MOV, CUSTO, OBSERVACAO) VALUES (11, v_setor_comercial,
'SAIDA', v_base + k, 75 - (5*k), 'Saída para cliente'); END LOOP;

COMMIT; END; /

PROMPT === Conferência de carga === COLUMN T FORMAT
A15 SELECT 'CH_SETOR' T, COUNT() QTD FROM CH_SETOR UNION
ALL SELECT 'CH_MOTOCICLETA', COUNT() FROM CH_MOTOCICLETA
UNION ALL SELECT 'CH_MOVIMENTACAO', COUNT(*) FROM
CH_MOVIMENTACAO; PROMPT

```

```

=====
=====

/* ===== [04] PACKAGE
REGRAS ===== */ CREATE OR REPLACE PACKAGE
PKG_CH AS c_err_tipo_invalido CONSTANT PLS_INTEGER := -20061;
c_err_status_invalido CONSTANT PLS_INTEGER := -20091;
c_err_fk_inexistente CONSTANT PLS_INTEGER := -20041;

FUNCTION fun_valida_tipo(p_tipo VARCHAR2) RETURN
NUMBER; FUNCTION fun_valida_status(p_status VARCHAR2)
RETURN NUMBER; FUNCTION fun_soma_custo_moto(p_id_moto
NUMBER) RETURN NUMBER;

PROCEDURE prc_registrar_auditoria( p_tabela IN VARCHAR2,
p_operacao IN VARCHAR2, p_id_registro IN NUMBER, p_detalhe IN
VARCHAR2 );

PROCEDURE prc_registrar_mov( p_id_moto IN NUMBER,
p_id_setor IN NUMBER, p_tipo_mov IN VARCHAR2, p_custo IN
NUMBER DEFAULT 0, p_obs IN VARCHAR2 DEFAULT NULL );

PROCEDURE prc_atualiza_status_moto( p_id_moto IN
NUMBER, p_novo_status IN VARCHAR2 );

PROCEDURE prc_rel_somas; END PKG_CH; /

CREATE OR REPLACE PACKAGE BODY PKG_CH AS FUNCTION
fun_valida_tipo(p_tipo VARCHAR2) RETURN NUMBER IS BEGIN IF
UPPER(p_tipo) IN ('ENTRADA','SAIDA','TRANSF') THEN RETURN 1; ELSE
RETURN 0; END IF; END;

FUNCTION fun_valida_status(p_status VARCHAR2) RETURN
NUMBER IS BEGIN IF UPPER(p_status) IN

```

```

('ATIVA','MANUTENCAO','INATIVA') THEN RETURN 1; ELSE RETURN 0;
END IF; END;

FUNCTION fun_soma_custo_moto(p_id_moto NUMBER)
RETURN NUMBER IS v_total NUMBER := 0; BEGIN SELECT
NVL(SUM(custo),0) INTO v_total FROM CH_MOVIMENTACAO WHERE
id_moto = p_id_moto; RETURN v_total; EXCEPTION WHEN
NO_DATA_FOUND THEN RETURN 0; END;

PROCEDURE prc_registrar_auditoria( p_tabela IN VARCHAR2,
p_operacao IN VARCHAR2, p_id_registro IN NUMBER, p_detalhe IN
VARCHAR2 ) IS BEGIN INSERT INTO CH_AUDITORIA (tabela,
operacao, id_registro, detalhe) VALUES (UPPER(p_tabela),
UPPER(p_operacao), p_id_registro, p_detalhe); END;

PROCEDURE prc_registrar_mov( p_id_moto IN NUMBER,
p_id_setor IN NUMBER, p_tipo_mov IN VARCHAR2, p_custo IN
NUMBER DEFAULT 0, p_obs IN VARCHAR2 DEFAULT NULL ) IS v_ok
NUMBER; v_existe NUMBER; v_id_mov NUMBER; BEGIN -- FKs
SELECT COUNT(*) INTO v_existe FROM CH_MOTOCICLETA WHERE
id_moto = p_id_moto; IF v_existe = 0 THEN
RAISE_APPLICATION_ERROR(c_err_fk_inexistente, 'Moto inexistente:
'||p_id_moto); END IF;

SELECT COUNT(*) INTO v_existe FROM CH_SETOR WHERE
id_setor = p_id_setor;
IF v_existe = 0 THEN
RAISE_APPLICATION_ERROR(c_err_fk_inexistente, 'Setor inexistente:
'||p_id_setor);
END IF;

-- validações básicas
v_ok := fun_valida_tipo(p_tipo_mov);
IF v_ok = 0 THEN RAISE_APPLICATION_ERROR(c_err_tipo_invalido,

```

```

'Tipo inválido:'||p_tipo_mov); END IF;
IF p_custo < 0 THEN RAISE_APPLICATION_ERROR(-20011, 'Custo não
pode ser negativo'); END IF;

INSERT INTO CH_MOVIMENTACAO (id_moto, id_setor, tipo_mov,
dt_mov, custo, observacao)
VALUES (p_id_moto, p_id_setor, UPPER(p_tipo_mov), SYSDATE,
p_custo, p_obs)
RETURNING id_mov INTO v_id_mov;

DBMS_OUTPUT.PUT_LINE('Movimentacao criada:
ID_MOV='||v_id_mov);

prc_registrar_auditoria('CH_MOVIMENTACAO','INSERT',v_id_mov,
'Nova movimentacao - moto='||p_id_moto||
', setor='||p_id_setor||', tipo='||UPPER(p_tipo_mov)|||
', custo='||TO_CHAR(p_custo,'FM9999990D00'));

END;

PROCEDURE prc_atualiza_status_moto(p_id_moto IN
NUMBER, p_novo_status IN VARCHAR2) IS v_ok NUMBER; v_rows
NUMBER; BEGIN v_ok := fun_valida_status(p_novo_status); IF v_ok =
0 THEN RAISE_APPLICATION_ERROR(c_err_status_invalido, 'Status
inválido: '||p_novo_status); END IF;

UPDATE CH_MOTOCICLETA SET status =
UPPER(p_novo_status) WHERE id_moto = p_id_moto;
v_rows := SQL%ROWCOUNT;
IF v_rows = 0 THEN
RAISE_APPLICATION_ERROR(c_err_fk_inexistente, 'Moto inexistente:
'||p_id_moto);
END IF;

```

```

DBMS_OUTPUT.PUT_LINE('Status da moto'||p_id_moto||' atualizado
para                               '||UPPER(p_novo_status));
prc_registrar_auditoria('CH_MOTOCICLETA','UPDATE',p_id_moto,'STA
TUS='||UPPER(p_novo_status));

END;

PROCEDURE prc_rel_somas IS v_total_mov NUMBER;
v_total_custo NUMBER; BEGIN SELECT COUNT(*),
NVL(SUM(custo),0) INTO v_total_mov, v_total_custo FROM
CH_MOVIMENTACAO; DBMS_OUTPUT.PUT_LINE('Total de
movimentacoes: '||v_total_mov); DBMS_OUTPUT.PUT_LINE('Soma de
custos: '||TO_CHAR(v_total_custo,'FM9999990D00'));

DBMS_OUTPUT.PUT_LINE('Top 5 motos por custo:');
FOR r IN (
SELECT m.id_moto, fun_soma_custo_moto(m.id_moto) soma
FROM CH_MOTOCICLETA m
ORDER BY soma DESC
FETCH FIRST 5 ROWS ONLY
)
LOOP
DBMS_OUTPUT.PUT_LINE(' - Moto'||r.id_moto||':'
'||TO_CHAR(r.soma,'FM9999990D00'));
END LOOP;

END; END PKG_CH; / PROMPT ..package regras ok

/* ===== [05] WRAPPERS &
TRIGGERS ===== */ CREATE OR REPLACE
PROCEDURE PRC_CH_REGISTRAR_MOV( p_id_moto IN NUMBER,

```

```

p_id_setor IN NUMBER, p_tipo_mov IN VARCHAR2, p_custo IN
NUMBER DEFAULT 0, p_obs IN VARCHAR2 DEFAULT NULL ) AS BEGIN
PKG_CH.prc_registrar_mov(p_id_moto,    p_id_setor,    p_tipo_mov,
p_custo, p_obs); END; /


CREATE          OR          REPLACE          PROCEDURE
PRC_CH_ATUALIZA_STATUS_MOTO(   p_id_moto   IN   NUMBER,
p_novo_status   IN   VARCHAR2      )   AS   BEGIN
PKG_CH.prc_atualiza_status_moto(p_id_moto,      p_novo_status);
END; /


CREATE OR REPLACE PROCEDURE PRC_CH_REL_SOMAS AS
BEGIN PKG_CH.prc_rel_somas; END; /


-- trigger: PLACA sempre maiúscula CREATE OR REPLACE
TRIGGER TRG_CH_MOTO_PLACA_UP BEFORE INSERT OR UPDATE
OF PLACA ON CH_MOTOCICLETA FOR EACH ROW BEGIN
:NEW.PLACA := UPPER(:NEW.PLACA); END; /


-- trigger: auditoria INSERT em MOV CREATE OR REPLACE
TRIGGER     TRG_CH_MOV_AUD_INS     AFTER     INSERT     ON
CH_MOVIMENTACAO     FOR     EACH     ROW     BEGIN
PKG_CH.prc_registrar_auditoria( p_tabela => 'CH_MOVIMENTACAO',
p_operacao => 'INSERT', p_id_registro => :NEW.ID_MOV, p_detalhe =>
'TRG:  moto='||:NEW.ID_MOTO||  ',  setor='||:NEW.ID_SETOR||  ',
tipo='||:NEW.TIPO_MOV||                                ',
custo='||TO_CHAR(:NEW.CUSTO,'FM9999990D00') ); END; /


-- trigger: auditoria UPDATE/DELETE em MOV CREATE OR
REPLACE TRIGGER TRG_CH_MOV_AUD_UPDDEL AFTER UPDATE OR
DELETE ON CH_MOVIMENTACAO FOR EACH ROW DECLARE v_op
VARCHAR2(10); v_id NUMBER; v_det VARCHAR2(4000); BEGIN IF
UPDATING THEN v_op := 'UPDATE'; v_id := :NEW.ID_MOV; v_det :=
```

```

'TRG:           id='||:NEW.ID_MOV||'           custo:      '||

NVL(TO_CHAR(:OLD.CUSTO,'FM9999990D00'),'0')||'->'||

NVL(TO_CHAR(:NEW.CUSTO,'FM9999990D00'),'0'); ELSIF DELETING
THEN v_op := 'DELETE'; v_id := :OLD.ID_MOV; v_det := 'TRG:
id='||:OLD.ID_MOV||' (removido); END IF;

          PKG_CH.prc_registrar_auditoria('CH_MOVIMENTACAO', v_op,
v_id, v_det); END; /


-- trigger: auditoria ao mudar STATUS da moto CREATE OR
REPLACE TRIGGER TRG_CH_MOTO_STATUS_AUD AFTER UPDATE OF
STATUS ON CH_MOTOCICLETA FOR EACH ROW BEGIN
PKG_CH.prc_registrar_auditoria( p_tabela => 'CH_MOTOCICLETA',
p_operacao => 'UPDATE', p_id_registro => :NEW.ID_MOTO, p_detalhe
=> 'TRG: STATUS '||:OLD.STATUS||'->'||:NEW.STATUS ); END; / PROMPT
..wrappers e triggers ok

/*
=====
[06] EXPORT JSON
=====
JSON manual, gravado em
CH_JSON_OUT
=====
*/
CREATE TABLE CH_JSON_OUT ( ORD NUMBER,
LINHA VARCHAR2(4000) );

CREATE OR REPLACE PACKAGE PKG_CH_JSON AS
PROCEDURE prc_gerar_json_mov_det; END PKG_CH_JSON; /


CREATE OR REPLACE PACKAGE BODY PKG_CH_JSON AS
FUNCTION fmt_num(p_n NUMBER) RETURN VARCHAR2 IS BEGIN
RETURN
TO_CHAR(p_n,'FM9999990D00','NLS_NUMERIC_CHARACTERS=.,');
END; PROCEDURE prc_gerar_json_mov_det IS v_ord NUMBER := 0;
v_first BOOLEAN := TRUE; v_count NUMBER;

```

```

PROCEDURE          out_line(p      VARCHAR2)      IS
BEGIN
v_ord           :=           v_ord           +           1;
INSERT INTO CH_JSON_OUT(ORD, LINHA) VALUES (v_ord, p);
END;

BEGIN DELETE FROM CH_JSON_OUT;

out_line('[');

FOR r           IN (SELECT
m.id_moto,
s.nome_setor      AS setor,
m.tipo_mov        AS tipo,
TO_CHAR(m.dt_mov,'YYYY-MM-DD"T"HH24:MI:SS') AS dt_mov,
NVL(m.observacao,"") AS obs,
m.custo
FROM CH_MOVIMENTACAO m
JOIN CH_SETOR s ON s.id_setor = m.id_setor
WHERE m.id_moto IN (1,6,11)
ORDER BY m.id_moto, m.dt_mov
)
LOOP
IF v_first       THEN
v_first         := FALSE;
ELSE
out_line(',');
END IF;

out_line(
'{"'||'"id_moto":'||r.id_moto||','||'

```

```

    ""setor":'||REPLACE(r.setor,"","","")||",||

    ""tipo":'||REPLACE(r.tipo                               ,"""")||",||

    ""dt_mov":'||r.dt_mov||",||

    ""obs":'||REPLACE(r.obs                           ,"""")||",||

    ""custo":'||fmt_num(r.custo)||

}

);

END                                     LOOP;

out_line(']');

SELECT  COUNT(*)  INTO  v_count   FROM  CH_JSON_OUT;
DBMS_OUTPUT.PUT_LINE('JSON gerado em CH_JSON_OUT com
'||v_count||' linhas.');

COMMIT;

END prc_gerar_json_mov_det; END PKG_CH_JSON; / PROMPT
..export ok

BEGIN PKG_CH_JSON.prc_gerar_json_mov_det; END; /

-- Preview e contagem PROMPT === Preview CH_JSON_OUT
SELECT * FROM CH_JSON_OUT WHERE ORD <= 20 ORDER BY ORD;

PROMPT === Total de linhas SELECT COUNT(*) AS
QTD_LINHAS FROM CH_JSON_OUT;

-- ===== [07] SPOOL (OPCIONAL)
===== -- escolha UM dos blocos abaixo (remova
os "--" para habilitar):

-- (7) SPOOL simples no diretório atual SET SQLFORMAT JSON
SET LONG 1000000 LONGCHUNKSIZE 1000000 SET PAGES 0

```

```
TRIMSPPOOL ON LINESIZE 32767 -- Dica: garantir UTF-8 no cliente --
Windows: set NLS_LANG=.AL32UTF8 (na mesma janela) --
PowerShell: chcp 65001 SPOOL ch_movimentos.json SELECT LINHA
FROM CH_JSON_OUT ORDER BY ORD; SPOOL OFF

PROMPT == FIM ==
```

- Mongo DB

MONGO - POWERSHEL

0. Pré-requisitos e pasta de trabalho

0.1) VERIFIQUE SE O SERVIÇO MONGODB ESTÁ ATIVO

```
Get-Service MongoDB | Select-Object Status,Name
```

0.2) VERSÕES DO CLIENTE

```
& "$env:LOCALAPPDATA\Programs\mongosh\mongosh.exe" --version
```

0.3) LOCALIZAR MONGOIMPORT.EXE (DATABASE TOOLS)

```
$mongoimportExe = (Get-ChildItem -Path "C:\Program Files","C:\Program Files(x86)","$env:LOCALAPPDATA","$env:LOCALAPPDATA\Programs" -Recurse -Filter mongoimport.exe -ErrorAction SilentlyContinue | Select-Object -First 1).FullName $mongoimportExe
```

0.4) PASTA DE TRABALHO

```
New-Item -ItemType Directory -Force -Path "$env:USERPROFILE\Documents\Mongo" | Out-Null Set-Location "$env:USERPROFILE\Documents\Mongo"
```

1. Converter o JSON do Oracle

1.1) LER O ARQUIVO DO ORACLE E TRANSFORMAR EM OBJETO

```
$orig = Get-Content .\ch_movimentos.json -Raw |  
ConvertFrom-Json
```

1.2) EXTRAIR APENAS A COLUNA "LINHA" E FILTRAR OS OBJETOS JSON

```
$linhas = $orig.results[0].items | ForEach-Object { $.linha }  
$objetos = $linhas | Where-Object { $ -match '^\\s*{.}\\s*$' }  
  
if (-not $objetos -or $objetos.Count -eq 0) { throw "Não encontrei objetos JSON nas linhas." }
```

1.3) MONTAR O ARRAY JSON VÁLIDO E SALVAR EM UTF-8

```
$conteudoFinal = "[rn" + ($objetos -join ",rn") + "rn]" Set-Content .\ch_movimentos_corrigido.json $conteudoFinal -Encoding utf8
```

1.4) SANITY CHECK

```
$null = Get-Content .\ch_movimentos_corrigido.json -Raw |  
ConvertFrom-Json "OK: ch_movimentos_corrigido.json reconstruído e válido." | Write-Host
```

2. Importar para o Mongo

2.1) IMPORTAR COMO JSON ARRAY

```
& $mongoimportExe --db lorarch --collection staging --drop --
file ".\ch_movimentos_corrigido.json" --jsonArray
```

2.2) CONFERIR

```
& "$env:LOCALAPPDATA\Programs\mongosh\mongosh.exe"
"mongodb://localhost:27017/lorarch" --eval
"db.staging.countDocuments()"
```

3. Criar coleção final motos (validator + carga via aggregation)

```
$js = @' use("lorarch");

// Limpa a coleção de destino (idempotente) db.motos.drop();

// Cria com schema validator db.createCollection("motos", {
validator: { $jsonSchema: { bsonType: "object", required: ["id_moto_sql","movimentacoes","total_custos"], properties: { id_moto_sql: { bsonType: "int" }, placa: { bsonType: ["string","null"] }, modelo:{ bsonType: ["string","null"] }, status:{ bsonType: ["string","null"] }, movimentacoes: { bsonType: "array", items: { bsonType: "object", required: ["setor","tipo","dt_mov","obs","custo"], properties: { setor:{ bsonType:"string" }, tipo: { bsonType:"string", enum:["ENTRADA","TRANSF","SAIDA"] }, dt_mov:{ bsonType:"string" }, // mantemos string ISO vinda do Oracle obs: { bsonType:"string" }, custo:{ bsonType:["double","int","decimal"] } } } }, total_custos: { bsonType:["double","int","decimal"] } } } });

// Monta os documentos finais a partir de 'staging'
db.motos.insertMany( db.staging.aggregate([ { $group: { _id: "$id_moto", movimentacoes: { $push: { setor: "$setor", tipo: "$tipo",
```

```

dt_mov: "$dt_mov", obs: { $ifNull: ["$obs","",] }, custo: { $toDouble:
"$custo" } }}, total_custos: { $sum: { $toDouble: "$custo" } } }}, {
$project: { _id: 0, id_moto_sql: { $toInt: "$_id" }, placa: null, modelo:
null, status: null, movimentacoes: 1, total_custos: 1 }}]).toArray() );

// Índices úteis db.motos.createIndex({ id_moto_sql: 1 }, {
unique: true }); db.motos.createIndex({ "movimentacoes.setor": 1 });
db.motos.createIndex({ total_custos: -1 });

// Prints rápidos print("Docs em staging:", db.staging.countDocuments());
print("Docs em motos:", db.motos.countDocuments()); // Exemplo print("Exemplo:",
EJSON.stringify(db.motos.findOne(), null, 2));'@

Set-Content -Path ".\mongo-setup.js" -Value $js -Encoding
UTF8

```

3.2) SETUP

```

& "$env:LOCALAPPDATA\Programs\mongosh\mongosh.exe"
"mongodb://localhost:27017/lorarch" --file ".\mongo-setup.js"

```

4. Corrigir acentuação

```

Set-Content -Path ".\fix-acentos.js" -Encoding UTF8 -Value @'
use("lorarch");

// Atualiza o array movimentacoes, consertando "obs"
db.motos.updateMany( {}, [ { $set: { movimentacoes: { $map: { input:
"$movimentacoes", as: "m", in: { $mergeObjects: [ "$$m", { obs: {
$replaceAll: { input: { $replaceAll: { input: "$$m.obs", find: "padrão",

```

```
replacement: "padrão" } }, find: "Transferência", replacement:  
"Transferência" }}]} } } } } }]); '@
```

4.2) Rodar o fix

```
& "$env:LOCALAPPDATA\Programs\mongosh\mongosh.exe"  
"mongodb://localhost:27017/lorarch" --file ".\fix-acentos.js"
```

4.3) Verificações rápidas

CONTAGENS

```
& "$env:LOCALAPPDATA\Programs\mongosh\mongosh.exe"  
lorarch --eval "db.staging.countDocuments();  
db.motos.countDocuments();"
```

EXEMPLO DE OBSERVAÇÕES CORRIGIDAS, MOTO 6 E MOTO 1

```
& "$env:LOCALAPPDATA\Programs\mongosh\mongosh.exe"  
lorarch --eval  
"printjson(db.motos.findOne({id_moto_sql:6},{_id:0,'movimentacoes  
.obs':1}))" &  
"$env:LOCALAPPDATA\Programs\mongosh\mongosh.exe" lorarch --  
eval  
"printjson(db.motos.findOne({id_moto_sql:1},{_id:0,'movimentacoes  
.obs':1}))"
```

5. Consultas de validação

```
$queries = @' use("lorarch");
```

```

print("TOP 5 por total_custos"); printjson( db.motos.find({},
{id_moto_sql:1,total_custos:1,_id:0}) .sort({total_custos:-1}) .limit(5)
.toArray() );

print("\nUm exemplo com setor = 'Entrega'; printjson(
db.motos.find( { "movimentacoes.setor": "Entrega" }, { id_moto_sql:1,
"movimentacoes.$":1, _id:0 } ).limit(5).toArray() );

print("\nFiltro por período 2025-10-01 .. 2025-10-10");
printjson( db.motos.aggregate([ { $unwind: "$movimentacoes" }, {
$addFields: { movData: { $dateFromString: { dateString:
"$movimentacoes.dt_mov" } } }, { $match: { movData: { $gte:
ISODate("2025-10-01T00:00:00Z"), $lte: ISODate("2025-10-
10T23:59:59Z") } } }, { $project: { _id:0, id_moto_sql:1,
setor:"$movimentacoes.setor", tipo:"$movimentacoes.tipo",
dt:"$movimentacoes.dt_mov", custo:"$movimentacoes.custo" } } ], { $limit: 10 } ]).toArray() ); '@ Set-Content -Path ".\mongo-queries.js" -
Value $queries -Encoding UTF8

```

5.2) Executar

```

& "$env:LOCALAPPDATA\Programs\mongosh\mongosh.exe"
"mongodb://localhost:27017/lorarch" --file ".\mongo-queries.js"

```

6. (Depois) Atalho: executar tudo com um script

DENTRO DE C:\USERS\MARCOS RAMALHO\DOCUMENTS\MONGO

```
.\run-sprint4.ps1
```

24. Prints de Execução .SQL, Mongo DB, Mongo DB Compass

Planilha Query Builder

```
1 linha inserido.

Procedimento PL/SQL concluido com sucesso.

==== Conferência de carga ====
>>Query Run In:Resultado da Consulta
=====
```

Saída do Script x Resultado da Consulta x Resultado da Consulta 1 x Resultado da Consulta 2 x Resultado da Consulta 3 x

SQL | Todas as Linhas Extraídas: 3 em 0,072 segundos

T	QTD
1 CH_SETOR	15
2 CH_MOTOCICLETA	15
3 CH_MOVIMENTACAO	45



The screenshot shows the Oracle SQL Developer interface. The title bar has multiple tabs open, all labeled 'SPRINT-04~' followed by a number from 1 to 8. Below the title bar is a toolbar with various icons. The main area is divided into two panes. The left pane contains the following PL/SQL code:

```
Package PKG_CH compilado

Package Body PKG_CH compilado

..package regras ok
```

SPRINT-04.sql * SPRINT-04~1 * SPRINT-04~2 * SPRINT-04~3 * SPRINT-04~4 * SPRINT-04~5 * SPRINT-04~6 * SPRINT-04~7 * SPRINT-04~8

Planilha Query Builder

```

Procedure PRC_CH_REGISTRAR_MOV compilado
Procedure PRC_CH_ATUALIZA_STATUS_MOTO compilado
Procedure PRC_CH_REL_SOMAS compilado
Trigger TRG_CH_MOTO_PLACA_UP compilado
Trigger TRG_CH_MOV_AUD_INS compilado
Trigger TRG_CH_MOV_AUD_UPDDEL compilado
Trigger TRG_CH_MOTO_STATUS_AUD compilado
..wrappers e triggers ok

```

Saída do Script | Resultado da Consulta | Resultado da Consulta 1 | Resultado da Consulta 2 | Resultado da Consulta 3

SQL | Todas as Linhas Extraídas: 20 em 0,072 segundos

ORD	LINHA
1	1 [
2	2 {"id_moto":1,"setor":"Recepcao","tipo":"ENTRADA","dt_mov":"2025-09-29T00:00:00","obs":"Entrada padrão","custo":65.00}
3	,
4	4 {"id_moto":1,"setor":"Recepcao","tipo":"ENTRADA","dt_mov":"2025-09-30T00:00:00","obs":"Entrada padrão","custo":64.00}
5	,
6	6 {"id_moto":1,"setor":"Recepcao","tipo":"ENTRADA","dt_mov":"2025-10-01T00:00:00","obs":"Entrada padrão","custo":63.00}
7	,
8	8 {"id_moto":1,"setor":"Recepcao","tipo":"ENTRADA","dt_mov":"2025-10-02T00:00:00","obs":"Entrada padrão","custo":62.00}
9	,
10	10 {"id_moto":1,"setor":"Recepcao","tipo":"ENTRADA","dt_mov":"2025-10-03T00:00:00","obs":"Entrada padrão","custo":61.00}
11	,
12	12 {"id_moto":1,"setor":"Recepcao","tipo":"ENTRADA","dt_mov":"2025-10-04T00:00:00","obs":"Entrada padrão","custo":60.00}
13	,
14	14 {"id_moto":1,"setor":"Recepcao","tipo":"ENTRADA","dt_mov":"2025-10-05T00:00:00","obs":"Entrada padrão","custo":59.00}
15	,
16	16 {"id_moto":1,"setor":"Recepcao","tipo":"ENTRADA","dt_mov":"2025-10-06T00:00:00","obs":"Entrada padrão","custo":58.00}
17	,
18	18 {"id_moto":1,"setor":"Recepcao","tipo":"ENTRADA","dt_mov":"2025-10-07T00:00:00","obs":"Entrada padrão","custo":57.00}
19	,
20	20 {"id_moto":1,"setor":"Recepcao","tipo":"ENTRADA","dt_mov":"2025-10-08T00:00:00","obs":"Entrada padrão","custo":56.00}

Saída do Script | Resultado da Consulta | Resultado da Consulta 1 | Resultado da Consulta 2 | Resultado da Consulta 3

SQL | Todas as Linhas Extraídas: 1 em 0,011 segundos

QTD_LINHAS
1
91

Saída do Script | Resultado da Consulta | Resultado da Consulta 1 | Resultado da Consulta 2 | Resultado da Consulta 3

SQL | Todas as Linhas Extraídas: 91 em 0,046 segundos

LINHA

```
1 [
2 {"id_moto":1,"setor":"Recepcao","tipo":"ENTRADA","dt_mov":"2025-09-29T00:00:00","obs":"Entrada padrão","custo":65.00}
3 ,
4 {"id_moto":1,"setor":"Recepcao","tipo":"ENTRADA","dt_mov":"2025-09-30T00:00:00","obs":"Entrada padrão","custo":64.00}
5 ,
6 {"id_moto":1,"setor":"Recepcao","tipo":"ENTRADA","dt_mov":"2025-10-01T00:00:00","obs":"Entrada padrão","custo":63.00}
7 ,
8 {"id_moto":1,"setor":"Recepcao","tipo":"ENTRADA","dt_mov":"2025-10-02T00:00:00","obs":"Entrada padrão","custo":62.00}
9 ,
10 {"id_moto":1,"setor":"Recepcao","tipo":"ENTRADA","dt_mov":"2025-10-03T00:00:00","obs":"Entrada padrão","custo":61.00}
11 ,
12 {"id_moto":1,"setor":"Recepcao","tipo":"ENTRADA","dt_mov":"2025-10-04T00:00:00","obs":"Entrada padrão","custo":60.00}
13 ,
14 {"id_moto":1,"setor":"Recepcao","tipo":"ENTRADA","dt_mov":"2025-10-05T00:00:00","obs":"Entrada padrão","custo":59.00}
15 ,
16 {"id_moto":1,"setor":"Recepcao","tipo":"ENTRADA","dt_mov":"2025-10-06T00:00:00","obs":"Entrada padrão","custo":58.00}
17 ,
18 {"id_moto":1,"setor":"Recepcao","tipo":"ENTRADA","dt_mov":"2025-10-07T00:00:00","obs":"Entrada padrão","custo":57.00}
19 ,
20 {"id_moto":1,"setor":"Recepcao","tipo":"ENTRADA","dt_mov":"2025-10-08T00:00:00","obs":"Entrada padrão","custo":56.00}
21 ,
22 {"id_moto":1,"setor":"Recepcao","tipo":"ENTRADA","dt_mov":"2025-10-09T00:00:00","obs":"Entrada padrão","custo":55.00}
23 ,
24 {"id_moto":1,"setor":"Recepcao","tipo":"ENTRADA","dt_mov":"2025-10-10T00:00:00","obs":"Entrada padrão","custo":54.00}
25 ,
26 {"id_moto":1,"setor":"Recepcao","tipo":"ENTRADA","dt_mov":"2025-10-11T00:00:00","obs":"Entrada padrão","custo":53.00}
```

Saída do Script | Resultado da Consulta | Resultado da Consulta 1 | Resultado da Consulta 2 | Resultado da Consulta 3

SQL | Todas as Linhas Extraídas: 91 em 0,046 segundos

LINHA

```
50 {"id_moto":6,"setor":"Entrega","tipo":"TRANSF","dt_mov":"2025-10-08T00:00:00","obs":"Transferência de setor","custo":112.00}
51 ,
52 {"id_moto":6,"setor":"Entrega","tipo":"TRANSF","dt_mov":"2025-10-09T00:00:00","obs":"Transferência de setor","custo":110.00}
53 ,
54 {"id_moto":6,"setor":"Entrega","tipo":"TRANSF","dt_mov":"2025-10-10T00:00:00","obs":"Transferência de setor","custo":108.00}
55 ,
56 {"id_moto":6,"setor":"Entrega","tipo":"TRANSF","dt_mov":"2025-10-11T00:00:00","obs":"Transferência de setor","custo":106.00}
57 ,
58 {"id_moto":6,"setor":"Entrega","tipo":"TRANSF","dt_mov":"2025-10-12T00:00:00","obs":"Transferência de setor","custo":104.00}
59 ,
60 {"id_moto":6,"setor":"Entrega","tipo":"TRANSF","dt_mov":"2025-10-13T00:00:00","obs":"Transferência de setor","custo":102.00}
61 ,
62 {"id_moto":11,"setor":"Comercial","tipo":"SAIDA","dt_mov":"2025-09-29T00:00:00","obs":"Saida para cliente","custo":75.00}
63 ,
64 {"id_moto":11,"setor":"Comercial","tipo":"SAIDA","dt_mov":"2025-09-30T00:00:00","obs":"Saida para cliente","custo":70.00}
65 ,
66 {"id_moto":11,"setor":"Comercial","tipo":"SAIDA","dt_mov":"2025-10-01T00:00:00","obs":"Saida para cliente","custo":65.00}
67 ,
68 {"id_moto":11,"setor":"Comercial","tipo":"SAIDA","dt_mov":"2025-10-02T00:00:00","obs":"Saida para cliente","custo":60.00}
69 ,
70 {"id_moto":11,"setor":"Comercial","tipo":"SAIDA","dt_mov":"2025-10-03T00:00:00","obs":"Saida para cliente","custo":55.00}
71 ,
72 {"id_moto":11,"setor":"Comercial","tipo":"SAIDA","dt_mov":"2025-10-04T00:00:00","obs":"Saida para cliente","custo":50.00}
73 ,
74 {"id_moto":11,"setor":"Comercial","tipo":"SAIDA","dt_mov":"2025-10-05T00:00:00","obs":"Saida para cliente","custo":45.00}
```

Saída do Script | Resultado da Consulta | Resultado da Consulta 1 | Resultado da Consulta 2 | Resultado da Consulta 3

SQL | Todas as Linhas Extraídas: 91 em 0,046 segundos

```
-- LINHA
26 {"id_moto":1,"setor":"Recepcao","tipo":"ENTRADA","dt_mov":"2025-10-11T00:00:00","obs":"Entrada padrão","custo":53.00}
27 ,
28 {"id_moto":1,"setor":"Recepcao","tipo":"ENTRADA","dt_mov":"2025-10-12T00:00:00","obs":"Entrada padrão","custo":52.00}
29 ,
30 {"id_moto":1,"setor":"Recepcao","tipo":"ENTRADA","dt_mov":"2025-10-13T00:00:00","obs":"Entrada padrão","custo":51.00}
31 ,
32 {"id_moto":6,"setor":"Entrega","tipo":"TRANSF","dt_mov":"2025-09-29T00:00:00","obs":"Transferência de setor","custo":130.00}
33 ,
34 {"id_moto":6,"setor":"Entrega","tipo":"TRANSF","dt_mov":"2025-09-30T00:00:00","obs":"Transferência de setor","custo":128.00}
35 ,
36 {"id_moto":6,"setor":"Entrega","tipo":"TRANSF","dt_mov":"2025-10-01T00:00:00","obs":"Transferência de setor","custo":126.00}
37 ,
38 {"id_moto":6,"setor":"Entrega","tipo":"TRANSF","dt_mov":"2025-10-02T00:00:00","obs":"Transferência de setor","custo":124.00}
39 ,
40 {"id_moto":6,"setor":"Entrega","tipo":"TRANSF","dt_mov":"2025-10-03T00:00:00","obs":"Transferência de setor","custo":122.00}
41 ,
42 {"id_moto":6,"setor":"Entrega","tipo":"TRANSF","dt_mov":"2025-10-04T00:00:00","obs":"Transferência de setor","custo":120.00}
43 ,
44 {"id_moto":6,"setor":"Entrega","tipo":"TRANSF","dt_mov":"2025-10-05T00:00:00","obs":"Transferência de setor","custo":118.00}
45 ,
46 {"id_moto":6,"setor":"Entrega","tipo":"TRANSF","dt_mov":"2025-10-06T00:00:00","obs":"Transferência de setor","custo":116.00}
47 ,
48 {"id_moto":6,"setor":"Entrega","tipo":"TRANSF","dt_mov":"2025-10-07T00:00:00","obs":"Transferência de setor","custo":114.00}
49 ,
50 {"id_moto":6,"setor":"Entrega","tipo":"TRANSF","dt_mov":"2025-10-08T00:00:00","obs":"Transferência de setor","custo":112.00}
```

Saída do Script | Resultado da Consulta | Resultado da Consulta 1 | Resultado da Consulta 2 | Resultado da Consulta 3

SQL | Todas as Linhas Extraídas: 91 em 0,046 segundos

```
-- LINHA
65 ,
66 {"id_moto":11,"setor":"Comercial","tipo":"SAIDA","dt_mov":"2025-10-01T00:00:00","obs":"Saida para cliente","custo":65.00}
67 ,
68 {"id_moto":11,"setor":"Comercial","tipo":"SAIDA","dt_mov":"2025-10-02T00:00:00","obs":"Saida para cliente","custo":60.00}
69 ,
70 {"id_moto":11,"setor":"Comercial","tipo":"SAIDA","dt_mov":"2025-10-03T00:00:00","obs":"Saida para cliente","custo":55.00}
71 ,
72 {"id_moto":11,"setor":"Comercial","tipo":"SAIDA","dt_mov":"2025-10-04T00:00:00","obs":"Saida para cliente","custo":50.00}
73 ,
74 {"id_moto":11,"setor":"Comercial","tipo":"SAIDA","dt_mov":"2025-10-05T00:00:00","obs":"Saida para cliente","custo":45.00}
75 ,
76 {"id_moto":11,"setor":"Comercial","tipo":"SAIDA","dt_mov":"2025-10-06T00:00:00","obs":"Saida para cliente","custo":40.00}
77 ,
78 {"id_moto":11,"setor":"Comercial","tipo":"SAIDA","dt_mov":"2025-10-07T00:00:00","obs":"Saida para cliente","custo":35.00}
79 ,
80 {"id_moto":11,"setor":"Comercial","tipo":"SAIDA","dt_mov":"2025-10-08T00:00:00","obs":"Saida para cliente","custo":30.00}
81 ,
82 {"id_moto":11,"setor":"Comercial","tipo":"SAIDA","dt_mov":"2025-10-09T00:00:00","obs":"Saida para cliente","custo":25.00}
83 ,
84 {"id_moto":11,"setor":"Comercial","tipo":"SAIDA","dt_mov":"2025-10-10T00:00:00","obs":"Saida para cliente","custo":20.00}
85 ,
86 {"id_moto":11,"setor":"Comercial","tipo":"SAIDA","dt_mov":"2025-10-11T00:00:00","obs":"Saida para cliente","custo":15.00}
87 ,
88 {"id_moto":11,"setor":"Comercial","tipo":"SAIDA","dt_mov":"2025-10-12T00:00:00","obs":"Saida para cliente","custo":10.00}
89 ,
90 {"id_moto":11,"setor":"Comercial","tipo":"SAIDA","dt_mov":"2025-10-13T00:00:00","obs":"Saida para cliente","custo":5.00}
91 ]
```

Mongo

```
PS C:\Users\Marcos Ramalho\Documents\Mongo>
PS C:\Users\Marcos Ramalho\Documents\Mongo> Set-Content -Path ".\mongo-setup.js" -Value $js -Encoding UTF8
PS C:\Users\Marcos Ramalho\Documents\Mongo> # 3.2) setup
PS C:\Users\Marcos Ramalho\Documents\Mongo> & $env:LOCALAPPDATA\Programs\mongosh\mongosh.exe "mongodb://localhost:27017/lorarch" --file ".\mongo-setup.js"
Docs em staging: 45
Docs em motos: 3
```

```
"movimentacoes": [
  {
    "setor": "Comercial",
    "tipo": "SAIDA",
    "dt_mov": "2025-09-29T00:00:00",
    "obs": "Saída para cliente",
    "custo": 75
  },
  {
    "setor": "Comercial",
    "tipo": "SAIDA",
    "dt_mov": "2025-10-01T00:00:00",
    "obs": "Saída para cliente",
    "custo": 65
  },
  {
    "setor": "Comercial",
    "tipo": "SAIDA",
    "dt_mov": "2025-10-02T00:00:00",
    "obs": "Saída para cliente",
    "custo": 60
  },
  {
    "setor": "Comercial",
    "tipo": "SAIDA",
    "dt_mov": "2025-09-30T00:00:00",
    "obs": "Saída para cliente",
    "custo": 70
  },
  {
    "setor": "Comercial",
    "tipo": "SAIDA",
    "dt_mov": "2025-10-08T00:00:00",
    "obs": "Saída para cliente",
    "custo": 30
  },
  {
    "setor": "Comercial",
    "tipo": "SAIDA",
    "dt_mov": "2025-10-04T00:00:00",
    "obs": "Saída para cliente",
    "custo": 50
  },
  {
    "setor": "Comercial",
    "tipo": "SAIDA",
    "dt_mov": "2025-10-09T00:00:00",
    "obs": "Saída para cliente",
    "custo": 25
  },
  {
    "setor": "Comercial",
    "tipo": "SAIDA",
    "dt_mov": "2025-10-10T00:00:00",
    "obs": "Saída para cliente",
    "custo": 20
  },
  {
    "setor": "Comercial",
    "tipo": "SAIDA",
    "dt_mov": "2025-10-11T00:00:00",
    "obs": "Saída para cliente",
    "custo": 15
  },
  {
    "setor": "Comercial",
    "tipo": "SAIDA",
    "dt_mov": "2025-10-12T00:00:00",
    "obs": "Saída para cliente",
    "custo": 10
  }
]
```

```
{
  "setor": "Comercial",
  "tipo": "SAIDA",
  "dt_mov": "2025-10-05T00:00:00",
  "obs": "Saída para cliente",
  "custo": 45
},
{
  "setor": "Comercial",
  "tipo": "SAIDA",
  "dt_mov": "2025-10-07T00:00:00",
  "obs": "Saída para cliente",
  "custo": 35
},
{
  "setor": "Comercial",
  "tipo": "SAIDA",
  "dt_mov": "2025-10-06T00:00:00",
  "obs": "Saída para cliente",
  "custo": 40
},
],
"total_custos": 600,
"id_moto_sql": 11,
"placa": null,
"modelo": null,
"status": null
}
```

```
PS C:\Users\Marcos Ramalho\Documents\Mongo> Set-Content -Path ".\fix-acentos.js" -Encoding UTF8 -Value @'
>>> use("lorarch");'
```



```
mongosh mongo://localhost > Windows PowerShell > + >
PS C:\Users\Marcos Ramalho\Documents\Mongo> Set-Content -Path ".\mongo-queries.js" -Value $queries -Encoding UTF8
PS C:\Users\Marcos Ramalho\Documents\Mongo> & "$env:LOCALAPPDATA\Programs\mongosh\mongosh.exe" "mongodb://localhost:27017/lorarch" --file ".\mongo-queries.js"
TOP 5 por total_custos:
[
  {
    total_custos: 1740,
    id_moto_sql: 6
  },
  {
    total_custos: 870,
    id_moto_sql: 1
  },
  {
    total_custos: 600,
    id_moto_sql: 11
  }
]

Um exemplo com setor = 'Entrega':
[
  {
    movimentacoes: [
      {
        setor: 'Entrega',
        tipo: 'TRANSF',
        dt_mov: '2025-09-30T00:00:00',
        obs: 'Transferência de setor',
        custo: 128
      }
    ],
    id_moto_sql: 6
  }
]

Filtro por periodo 2025-10-01 ... 2025-10-10:
[
  {
    id_moto_sql: 11,
    setor: 'Comercial',
    tipo: 'SAIDA',
    dt: '2025-10-01T00:00:00',
    custo: 65
  }
]
```

```
mongosh mongodb://localho X Windows PowerShell X + ▾
{
  id_moto_sql: 11,
  setor: 'Comercial',
  tipo: 'SAIDA',
  dt: '2025-10-02T00:00:00',
  custo: 60
},
{
  id_moto_sql: 11,
  setor: 'Comercial',
  tipo: 'SAIDA',
  dt: '2025-10-08T00:00:00',
  custo: 30
},
{
  id_moto_sql: 11,
  setor: 'Comercial',
  tipo: 'SAIDA',
  dt: '2025-10-04T00:00:00',
  custo: 50
},
{
  id_moto_sql: 11,
  setor: 'Comercial',
  tipo: 'SAIDA',
  dt: '2025-10-09T00:00:00',
  custo: 25
},
{
  id_moto_sql: 11,
  setor: 'Comercial',
  tipo: 'SAIDA',
  dt: '2025-10-10T00:00:00',
  custo: 20
},
{
  id_moto_sql: 11,
  setor: 'Comercial',
  tipo: 'SAIDA',
  dt: '2025-10-03T00:00:00',
  custo: 55
},
```

```
mongosh mongodb://localho X | Windows PowerShell X + ▾
dt: '2025-10-05T00:00:00',
custo: 118
},
{
  id_moto_sql: 6,
  setor: 'Entrega',
  tipo: 'TRANSF',
  dt: '2025-10-06T00:00:00',
  custo: 116
},
{
  id_moto_sql: 6,
  setor: 'Entrega',
  tipo: 'TRANSF',
  dt: '2025-10-07T00:00:00',
  custo: 114
},
{
  id_moto_sql: 6,
  setor: 'Entrega',
  tipo: 'TRANSF',
  dt: '2025-10-08T00:00:00',
  custo: 112
},
{
  id_moto_sql: 6,
  setor: 'Entrega',
  tipo: 'TRANSF',
  dt: '2025-10-09T00:00:00',
  custo: 110
},
{
  id_moto_sql: 6,
  setor: 'Entrega',
  tipo: 'TRANSF',
  dt: '2025-10-10T00:00:00',
  custo: 108
}
]
== FIM SPRINT 4 OK ==
Log salvo em: C:\Users\Marcos Ramalho\Documents\Mongo\sprint4_log_20251016_101532.txt
PS C:\Users\Marcos Ramalho\Documents\Mongo> |
```

```
[mongosh mongodb://localhost:27017]  Windows PowerShell
]
\nFiltro por período 2025-10-01 .. 2025-10-10:
[{"id_moto_sql": 6, "setor": "Entrega", "tipo": "TRANSF", "dt": "2025-10-01T00:00:00", "custo": 126}, {"id_moto_sql": 6, "setor": "Entrega", "tipo": "TRANSF", "dt": "2025-10-02T00:00:00", "custo": 124}, {"id_moto_sql": 6, "setor": "Entrega", "tipo": "TRANSF", "dt": "2025-10-03T00:00:00", "custo": 122}, {"id_moto_sql": 6, "setor": "Entrega", "tipo": "TRANSF", "dt": "2025-10-04T00:00:00", "custo": 120}, {"id_moto_sql": 6, "setor": "Entrega", "tipo": "TRANSF", "dt": "2025-10-05T00:00:00", "custo": 118}, {"id_moto_sql": 6, "setor": "Entrega", "tipo": "TRANSF", "dt": "2025-10-06T00:00:00", "custo": 116}, {"id_moto_sql": 6, "setor": "Entrega", "tipo": "TRANSF", "dt": "2025-10-07T00:00:00", "custo": 114}, {"id_moto_sql": 6, "setor": "Entrega", "tipo": "TRANSF", "dt": "2025-10-08T00:00:00", "custo": 112}, {"id_moto_sql": 6, "setor": "Entrega", "tipo": "TRANSF", "dt": "2025-10-09T00:00:00", "custo": 110}, {"id_moto_sql": 6, "setor": "Entrega", "tipo": "TRANSF", "dt": "2025-10-10T00:00:00", "custo": 108}]]
```

```
mongosh mongodb://localhost  X  Windows PowerShell  X  +  ▾
mongoimport: C:\Program Files\MongoDB\Tools\100\bin\mongoimport.exe
Detectado formato SQL Developer. Convertendo...
OK: C:\Users\Marcos Ramalho\Documents\Mongo\ch_movimentos_corrigido.json validado.
Importando para lorarch.staging...
2025-10-16T10:15:36.035-0300      connected to: mongodb://localhost/
2025-10-16T10:15:36.051-0300      dropping: lorarch.staging
2025-10-16T10:15:36.070-0300      45 document(s) imported successfully. 0 document(s) failed to import.
staging.countDocuments() = 45
Docs em staging: 45
Docs em motos: 3
ReferenceError: toJSON is not defined
TOP 5 por total_custos:
[
  {
    total_custos: 1740,
    id_moto_sql: 6
  },
  {
    total_custos: 870,
    id_moto_sql: 1
  },
  {
    total_custos: 600,
    id_moto_sql: 11
  }
]
\nUm exemplo com setor = 'Entrega':
[
  {
    movimentacoes: [
      {
        setor: 'Entrega',
        tipo: 'TRANSF',
        dt_mov: '2025-09-29T00:00:00',
        obs: 'Transferência de setor',
        custo: 130
      }
    ],
    id_moto_sql: 6
  }
]
\nFiltro por período 2025-10-01 .. 2025-10-10:
```

```

mongosh mongodb://localhost  X  Windows PowerShell  X  +  -
id_moto_sql: 11,
setor: 'Comercial',
tipo: 'SAIDA',
dt: '2025-10-05T00:00:00',
custo: 45
},
{
  id_moto_sql: 11,
  setor: 'Comercial',
  tipo: 'SAIDA',
  dt: '2025-10-07T00:00:00',
  custo: 35
},
{
  id_moto_sql: 11,
  setor: 'Comercial',
  tipo: 'SAIDA',
  dt: '2025-10-06T00:00:00',
  custo: 40
}
]
PS C:\Users\Marcos Ramalho\Documents\Mongo> .\run-sprint4.ps1
== SPRINT 4 RUN 16/10/2025 10:15:32 ==
WorkDir: C:\Users\Marcos Ramalho\Documents\Mongo
MongoDB service: Running
2.5.8
mongoimport: C:\Program Files\MongoDB\Tools\100\bin\mongoimport.exe
Detetado formato SQL Developer. Convertendo...
OK: C:\Users\Marcos Ramalho\Documents\Mongo\ch_movimentos_corrigido.json validado.
Importando para lorarch.staging...
2025-10-16T10:15:36.035-0300      connected to: mongodb://localhost/
2025-10-16T10:15:36.051-0300      dropping: lorarch.staging
2025-10-16T10:15:36.070-0300      45 document(s) imported successfully. 0 document(s) failed to import.
staging.countDocuments() = 45
Docs em staging: 45
Docs em motos: 3
ReferenceError: toJSON is not defined
TOP 5 por total_custos:
[
  {
    total_custos: 1740,
    id_moto_sql: 6

```

MongoDB Compass

The screenshot shows the MongoDB Compass interface with the following details:

- Connections:** LorArch
- Databases:**
 - admin**: Storage size: 20.48 kB, Collections: 0, Indexes: 1
 - config**: Storage size: 36.86 kB, Collections: 0, Indexes: 2
 - local**: Storage size: 36.86 kB, Collections: 1, Indexes: 1
 - lorarch**: Storage size: 40.96 kB, Collections: 2, Indexes: 5
- View:** Database Name
- Buttons:** Open MongoDB shell, Create database, Refresh

MongoDB Compass - LorArch/lorarch.motos

Connections Edit View Collection Help

Compass

My Queries

CONNECTIONS (1)

Search connections

LorArch > lorarch > motos

Documents 3 Aggregations Schema Indexes 4 Validation

Type a query: { field: 'value' } or [Generate query](#).

[EXPLAIN](#) [Reset](#) [Find](#) [Options](#)

[ADD DATA](#) [EXPORT DATA](#) [UPDATE](#) [DELETE](#)

25 1–3 of 3

```
_id: ObjectId('68f0eff9d686bbd2b8cebea4')
  movementacoes: Array (15)
    total_custos: 1740
    id_moto_sql: 6
    placas: null
    modelo: null
    status: null

_id: ObjectId('68f0eff9d686bbd2b8cebea5')
  movementacoes: Array (15)
    total_custos: 870
    id_moto_sql: 1
    placas: null
    modelo: null
    status: null

_id: ObjectId('68f0eff9d686bbd2b8cebea6')
  movementacoes: Array (15)
    total_custos: 600
    id_moto_sql: 11
    placas: null
    modelo: null
    status: null
```

MongoDB Compass - LorArch/lorarch.motos

Connections Edit View Collection Help

Compass

My Queries

CONNECTIONS (1)

Search connections

LorArch > lorarch > motos

Documents 3 Aggregations Schema Indexes 4 Validation

Your pipeline is currently empty. Need help getting started? [Generate aggregation](#).

[EXPLAIN](#) [Export](#) [Run](#) [Options](#)

[SAVE](#) [CREATE NEW](#) [EXPORT TO LANGUAGE](#)

[PREVIEW](#) [STAGES](#) [TEXT](#) [WIZARD](#)

3 Documents in the collection

Preview of documents

```
_id: ObjectId('68f0eff9d686bbd2b8cebea4')
  movementacoes: Array (15)
    total_custos: 1740
    id_moto_sql: 6
    placas: null
    modelo: null
    status: null

_id: ObjectId('68f0eff9d686bbd2b8cebea5')
  movementacoes: Array (15)
    total_custos: 870
    id_moto_sql: 1
    placas: null
    modelo: null
    status: null

_id: ObjectId('68f0eff9d686bbd2b8cebea6')
  movementacoes: Array (15)
    total_custos: 600
    id_moto_sql: 11
    placas: null
    modelo: null
    status: null
```

MongoDB Compass - LorArch/lorarch.motos

Connections Edit View Collection Help

Compass

My Queries

CONNECTIONS (1)

Search connections

LorArch > lorarch > motos

Documents 3 Aggregations Schema Indexes 4 Validation

[Create Index](#) [Refresh](#)

[VIEWING](#) [INDEXES](#) [SEARCH INDEXES](#)

Name & Definition	Type	Size	Usage	Properties	Status
<code>_id_</code>	REGULAR	20.5 KB	2 (since Thu Oct 16 2025)	UNIQUE	READY
<code>_id ↑</code>					
<code>id_moto_sql_1</code>	REGULAR	20.5 KB	0 (since Thu Oct 16 2025)	UNIQUE	READY
<code>id_moto_sql ↑</code>					
<code>movementacoes.setor_1</code>	REGULAR	20.5 KB	0 (since Thu Oct 16 2025)		READY
<code>movementacoes.setor ↑</code>					
<code>total_custos_-1</code>	REGULAR	20.5 KB	0 (since Thu Oct 16 2025)		READY
<code>total_custos ↓</code>					

MongoDB Compass - LorArch/lorarch.staging

Connections Edit View Collection Help

Compass

My Queries

CONNECTIONS (1)

LorArch > lorarch > staging

Documents 45 Aggregations Schema Indexes 1 Validation

Type a query: { field: 'value' } or [Generate query](#)

[EXPLAIN](#) [Reset](#) [Find](#) [Options](#)

[ADD DATA](#) [EXPORT DATA](#) [UPDATE](#) [DELETE](#)

25 1 - 25 of 45

`_id: ObjectId('68f0eff854d44296b183384c')
 id_moto: 1
 setor: "Recepcao"
 tipo: "ENTRADA"
 dt_mov: "2025-10-03T00:00:00"
 obs: "Entrada padrão"
 custo: 61`

`_id: ObjectId('68f0eff854d44296b183384d')
 id_moto: 1
 setor: "Recepcao"
 tipo: "ENTRADA"
 dt_mov: "2025-10-04T00:00:00"
 obs: "Entrada padrão"
 custo: 60`

`_id: ObjectId('68f0eff854d44296b183384e')
 id_moto: 1
 setor: "Recepcao"
 tipo: "ENTRADA"
 dt_mov: "2025-10-05T00:00:00"
 obs: "Entrada padrão"
 custo: 59`

[Run aggregation](#)

MongoDB Compass - LorArch/lorarch.staging

Connections Edit View Collection Help

Compass

My Queries

CONNECTIONS (1)

LorArch > lorarch > staging

Documents 45 Aggregations Schema Indexes 1 Validation

Your pipeline is currently empty. Need help getting started? [Generate aggregation](#)

[EXPLAIN](#) [Export](#) [Run](#) [Options](#)

[SAVE](#) [CREATE NEW](#) [EXPORT TO LANGUAGE](#)

Untitled PREVIEW STAGES TEXT WIZARD

45 Documents in the collection

Preview of documents

`_id: ObjectId('68f0eff854d44296b183384c')
 id_moto: 1
 setor: "Recepcao"
 tipo: "ENTRADA"
 dt_mov: "2025-10-03T00:00:00"
 obs: "Entrada padrão"
 custo: 61`

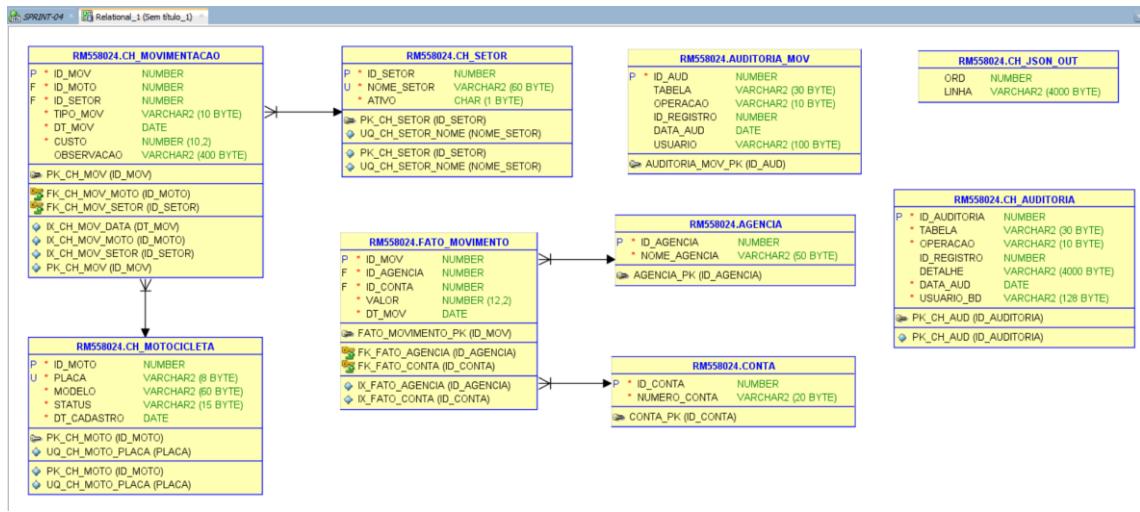
`_id: ObjectId('68f0eff854d44296b183384d')
 id_moto: 1
 setor: "Recepcao"
 tipo: "ENTRADA"
 dt_mov: "2025-10-04T00:00:00"
 obs: "Entrada padrão"
 custo: 60`

`_id: ObjectId('68f0eff854d44296b183384e')
 id_moto: 1
 setor: "Recepcao"
 tipo: "ENTRADA"
 dt_mov: "2025-10-05T00:00:00"
 obs: "Entrada padrão"
 custo: 59`

[Add stage](#)

Learn more about aggregation pipeline stages

Modelo Relacional



Link

Youtube : <https://youtu.be/n1fgHrcxrig>

Github: <https://github.com/Ramalho044/LorArch-BD.git>