

# **FACULDADE DE INFORMÁTICA E ADMINISTRAÇÃO PAULISTA**

Caike Dametto - RM: 558614

Guilherme Janunzzi - RM: 558461

**MottoMap**

São Paulo/SP

2025

## SUMÁRIO

Descritivo da Solução.....	2
Prints de Tela da Execução.....	3,4,5,6
Prints de Tela das Exceções Tratadas.....	7,8,9

## Descritivo da Solução

O MotoMap é um sistema criado para ajudar a empresa Mottu a organizar e controlar melhor os pátios onde ficam as motos. Hoje, esse controle é feito de forma manual, o que gera erros, atrasos e baixa produtividade. Com o MotoMap, isso muda.

O sistema permite que cada filial cadastre seu pátio com linhas e colunas, como uma grade. Assim, fica fácil saber onde cada moto está. As motos são separadas em áreas como: Prontas para uso, Com problema simples, Com problema grave, Irrecuperáveis e Minha Mottu.

Quando uma moto chega ao pátio, um colaborador faz a triagem, registra se há algum problema e escolhe onde a moto vai ficar. A escolha da vaga pode ser feita com ajuda de câmeras com inteligência artificial, que mostram as vagas livres.

O sistema também tem controle de usuários:

- Administradores: cadastram filiais, motos e colaboradores.
- Colaboradores de pátio: registram entrada e movimentações.
- Mecânicos: resolvem problemas e mudam o status da moto.

A aplicação funciona no celular e se conecta com duas APIs:

- A API Java (Spring Boot) cuida do cadastro de filiais, motos, posições, problemas e usuários.
- A API .NET (ASP.NET Core) cuida das movimentações internas e do histórico das motos.

O banco de dados usado é o Oracle/H2 e tudo roda em containers Docker, hospedados em uma máquina virtual da Azure.

Com o MotoMap, a Mottu ganha mais controle, agilidade e organização nos seus pátios.

## Prints de Tela da Execução

### Execução da Função F\_VALIDA\_COMPLEXIDADE\_SENHA

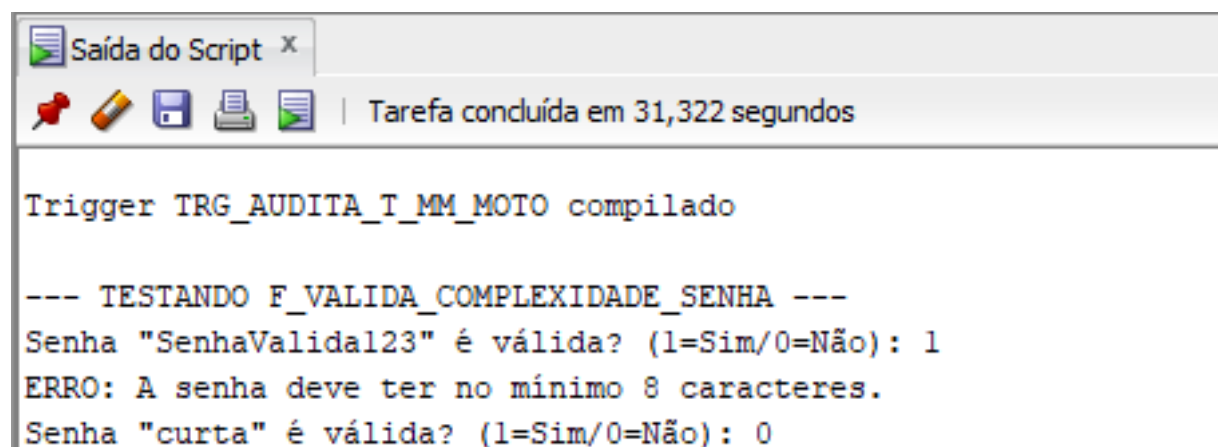
```

-----
-- Demonstração da Função 2: F_VALIDA_COMPLEXIDADE_SENHA
-----
DECLARE
    v_resultado NUMBER;
BEGIN
    DBMS_OUTPUT.PUT_LINE('--- TESTANDO F_VALIDA_COMPLEXIDADE_SENHA ---');
    v_resultado := F_VALIDA_COMPLEXIDADE_SENHA('SenhaValida123');
    DBMS_OUTPUT.PUT_LINE('Senha "SenhaValida123" é válida? (1=Sim/0=Não): ' || v_resultado);

    v_resultado := F_VALIDA_COMPLEXIDADE_SENHA('curta');
    DBMS_OUTPUT.PUT_LINE('Senha "curta" é válida? (1=Sim/0=Não): ' || v_resultado);

    -- Teste de exceção (senha curta)
    DBMS_OUTPUT.PUT_LINE('-- Testando exceção de senha curta --');
    v_resultado := F_VALIDA_COMPLEXIDADE_SENHA('abc');
END;
/

```



## Execução do Procedure P\_RELATORIO\_MOTOS\_JSON e F\_RELACIONAL\_PARA\_JSON

```
-- Demonstração do Procedure 1: P_RELATORIO_MOTOS_JSON

BEGIN
    DBMS_OUTPUT.PUT_LINE(CHR(10) || '--- TESTANDO P_RELATORIO_MOTOS_JSON ---');
    -- Execução com filial que tem dados
    P_RELATORIO_MOTOS_JSON(p_cd_filial => 1);

    -- Teste de exceção (filial inexistente)
    DBMS_OUTPUT.PUT_LINE(CHR(10) || '-- Testando exceção de filial inexistente --');
    P_RELATORIO_MOTOS_JSON(p_cd_filial => 999);
END;
```

Saída do Script x

Tarefa concluída em 31,322 segundos

```
--- TESTANDO P_RELATORIO_MOTOS_JSON ---
Relatório de Motos em JSON para a Filial 1

[
  {"cd_moto": 1, "ds_placa": "BRA2E19", "nm_modelo": "MOTTU SPORT ESD", "ds_marca": "MOTTU", "nr_ano": 2023, "ds_status": "Disponível"},
  {"cd_moto": 2, "ds_placa": "DUS3A70", "nm_modelo": "POP 110I", "ds_marca": "Honda", "nr_ano": 2024, "ds_status": "Em Manutenção"},
  {"cd_moto": 6, "ds_placa": "SAH2D23", "nm_modelo": "MOTTU SPORT", "ds_marca": "MOTTU", "nr_ano": 2024, "ds_status": "Vendido"}
]
```

## Execução do Procedure P\_RELATORIO\_AGREGADO\_MOVIMENTACOES

```
-- Demonstração do Procedure 2: P_RELATORIO_AGREGADO_MOVIMENTACOES

BEGIN
  DBMS_OUTPUT.PUT_LINE(CHR(10) || '--- TESTANDO P_RELATORIO_AGREGADO_MOVIMENTACOES ---');
  P_RELATORIO_AGREGADO_MOVIMENTACOES;
END;
/
```

Saída do Script x		
Tarefa concluída em 31,322 segundos		
--- TESTANDO P_RELATORIO_AGREGADO_MOVIMENTACOES ---		
FILIAL	TIPO MOVIMENTAÇÃO	VALOR SOMADO
Mottu MG	Entrada no Pátio	120.00
Mottu MG	Envio para Oficina	300.00
Subtotal Mottu MG		420.00
Mottu RJ	Entrada no Pátio	100.00
Mottu RJ	Transferência entre Filiais	50.00
Subtotal Mottu RJ		150.00
Mottu SP	Entrada no Pátio	200.00
Mottu SP	Envio para Oficina	700.50
Mottu SP	Retorno da Oficina	50.25
Mottu SP	Saída para Venda	15,000.00
Subtotal Mottu SP		15,950.75
TOTAL GERAL		16,520.75
Procedimento PL/SQL concluído com sucesso.		

## Execução do Trigger TRG\_AUDITA\_T\_MM\_MOTO

```

-- Demonstração do Trigger: TRG_AUDITA_T_MM_MOTO

DECLARE
  v_cd_moto T_MM_MOTO.CD_MOTO%TYPE;
BEGIN
  DBMS_OUTPUT.PUT_LINE(CHR(10) || '--- TESTANDO TRG_AUDITA_T_MM_MOTO ---');

  -- Teste INSERT
  DBMS_OUTPUT.PUT_LINE('-- Testando INSERT --');
  INSERT INTO T_MM_MOTO (CD_MOTO, CD_FILIAL, CD_MODELO, CD_STATUS_MOTO, DS_PLACA, DS_CHASSI
  VALUES (SEQ_MOTO.NEXTVAL, 1, 3, 1, 'XYZ1A23', '9C2KC1600NR999999')
  RETURNING CD_MOTO INTO v_cd_moto;
  DBMS_OUTPUT.PUT_LINE('Moto ' || v_cd_moto || ' inserida.');
```

```

  -- Teste UPDATE
  DBMS_OUTPUT.PUT_LINE('-- Testando UPDATE --');
  UPDATE T_MM_MOTO SET CD_STATUS_MOTO = 2, DS_PLACA = 'XYZ5B67' WHERE CD_MOTO = v_cd_moto;
  DBMS_OUTPUT.PUT_LINE('Moto ' || v_cd_moto || ' atualizada.');
```

```

  -- Teste DELETE
  DBMS_OUTPUT.PUT_LINE('-- Testando DELETE --');
  DELETE FROM T_MM_MOTO WHERE CD_MOTO = v_cd_moto;
  DBMS_OUTPUT.PUT_LINE('Moto ' || v_cd_moto || ' deletada.');
```

```

  COMMIT;
END;
/
```

Saída do Script x

Tarefa concluída em 31,322 segundos

```

--- TESTANDO TRG_AUDITA_T_MM_MOTO ---
-- Testando INSERT --
Moto 7 inserida.
-- Testando UPDATE --
Moto 7 atualizada.
-- Testando DELETE --
Moto 7 deletada.

Procedimento PL/SQL concluído com sucesso.
```

Saída do Script x

Tarefa concluída em 31,322 segundos

Procedimento PL/SQL concluído com sucesso.

CD_AUDITORIA	NM_USUARIO	DT_OPERACAO	NM_TABELA
3	RM558461	21/09/25 00:27:34,198475000	T_MM_MOTO
2	RM558461	21/09/25 00:27:34,191605000	T_MM_MOTO
1	RM558461	21/09/25 00:27:34,187512000	T_MM_MOTO

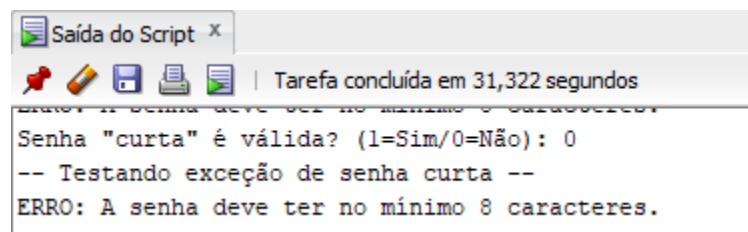
  

0	NM_TABELA	DS_OPERACAO	DS_VALORES_ANTIGOS	DS_VALORES_NOVOS
0:27:34,198475000	T_MM_MOTO	DELETE	PLACA: XYZ5B67; CHASSI: 9C2KC1600NR999999; STATUS: 2	PLACA: XYZ5B67; CHASSI: 9C2KC1600NR999999; STATUS: 2
0:27:34,191605000	T_MM_MOTO	UPDATE	PLACA: XYZ1A23; CHASSI: 9C2KC1600NR999999; STATUS: 1	PLACA: XYZ1A23; CHASSI: 9C2KC1600NR999999; STATUS: 1
0:27:34,187512000	T_MM_MOTO	INSERT		

## Prints de Tela das Exceções Tratadas

### Exceção da Função F\_VALIDA\_COMPLEXIDADE\_SENHA

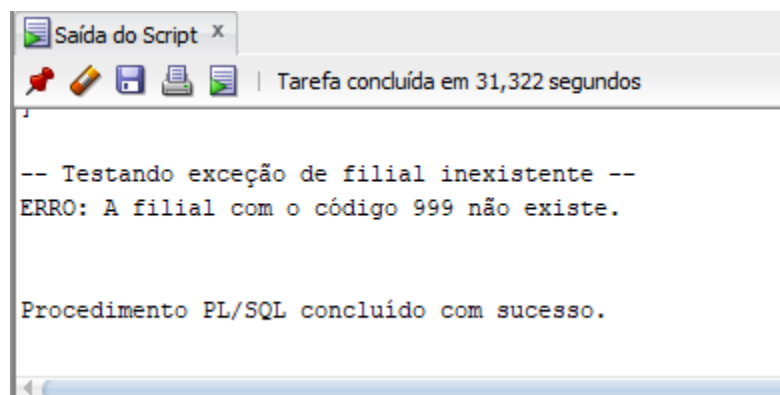
```
-- Teste de exceção (senha curta)
DBMS_OUTPUT.PUT_LINE('-- Testando exceção de senha curta --');
v_resultado := F_VALIDA_COMPLEXIDADE_SENHA('abc');
```





## Exceção do Procedure P\_RELATORIO\_MOTOS\_JSON

```
-- Teste de exceção (filial inexistente)
DBMS_OUTPUT.PUT_LINE(CHR(10) || '-- Testando exceção de filial inexistente --');
P_RELATORIO_MOTOS_JSON(p_cd_filial => 999);
```



Saída do Script x

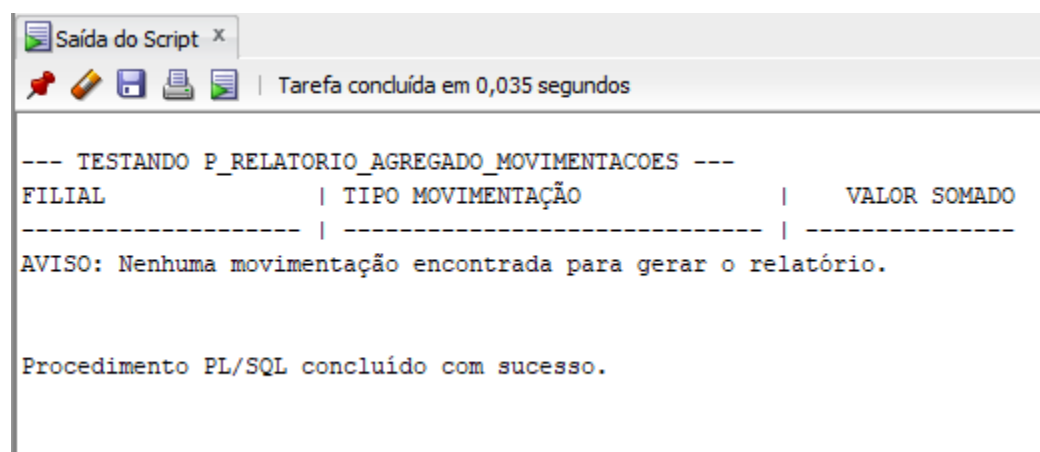
Tarefa concluída em 31,322 segundos

```
-- Testando exceção de filial inexistente --
ERRO: A filial com o código 999 não existe.

Procedimento PL/SQL concluído com sucesso.
```

## Exceção do Procedure P\_RELATORIO\_AGREGADO\_MOVIMENTACOES

```
-- -----  
-- Demonstração do Procedure 2: P_RELATORIO_AGREGADO_MOVIMENTACOES  
-- -----  
  
BEGIN  
    DBMS_OUTPUT.PUT_LINE(CHR(10) || '--- TESTANDO P_RELATORIO_AGREGADO_MOVIMENTACOES ---');  
    P_RELATORIO_AGREGADO_MOVIMENTACOES;  
END;  
/
```



Saída do Script x

Tarefa concluída em 0,035 segundos

```
--- TESTANDO P_RELATORIO_AGREGADO_MOVIMENTACOES ---  
FILIAL          | TIPO MOVIMENTAÇÃO          | VALOR SOMADO  
----- | ----- | -----  
AVISO: Nenhuma movimentação encontrada para gerar o relatório.  
  
Procedimento PL/SQL concluído com sucesso.
```