FIAP – Faculdade de Informática e Administração Paulista

Análise e Desenvolvimento de Sistemas

MASTERING RELATIONAL AND NON-RELATIONAL DATABASE

SISTEMA DE GESTÃO DE ABRIGOS EM SITUAÇÕES DE DESASTRES NATURAIS

São Paulo

2025

Glenda Delfy Vela Mamani – RM 552667

Lucas Alcântara Carvalho – RM 95111

Renan Bezerra dos Santos – RM 553228

MASTERING RELATIONAL AND NON-RELATIONAL DATABASE

SISTEMA DE GESTÃO DE ABRIGOS EM SITUAÇÕES DE DESASTRES NATURAIS

São Paulo

2025

# SUMÁRIO

1 INTRODUÇÃO .............................................................................................................. 4

2 MODELAGEM RELACIONAL .................................................................................... 5

3 FUNÇÕES, TRIGGERS, CURSOR, BLOCO ANONIMO E RELATÓRIO................. 6

4 CONSULTAS SQL ........................................................................................................ 8

5 USO DO MONGODB .................................................................................................. 11

6 CONCLUSÃO .............................................................................................................. 12

# 1 INTRODUÇÃO

Os desastres naturais são eventos imprevisíveis que afetam drasticamente a segurança, a saúde e o bem-estar da população. Em situações de emergência, como enchentes, deslizamentos ou ondas de calor extremo, a existência de abrigos bem estruturados e monitorados é essencial para garantir uma resposta rápida e eficaz. No entanto, a gestão eficiente desses abrigos exige controle em tempo real de dados como ocupação, recursos disponíveis e condições ambientais.

Este projeto propõe o desenvolvimento de uma solução tecnológica voltada à gestão de abrigos temporários em cenários de desastres naturais, utilizando banco de dados relacional (Oracle) para armazenar informações estruturadas e banco de dados não relacional (MongoDB) para registros históricos de ocupação. A proposta visa unir consultas avançadas em SQL, modelagem normalizada e técnicas modernas de armazenamento para oferecer uma base sólida e escalável para aplicações de apoio humanitário.

A seguir, são apresentadas a modelagem relacional adotada, consultas elaboradas para suporte à tomada de decisão, e o uso do MongoDB como complemento no registro de dados não estruturados e históricos.

# 2 MODELAGEM RELACIONAL

Diagrama

O conteúdo gerado por IA pode estar incorreto.

# 3 FUNÇÕES, TRIGGERS, CURSOR, BLOCO ANONIMO E RELATÓRIO

* Função – Avaliação de Risco por Ocupação

CREATE OR REPLACE FUNCTION fn\_risco\_abrigo(p\_id IN NUMBER)  
RETURN VARCHAR2 IS  
 v\_ocupacao NUMBER;  
 v\_capacidade NUMBER;  
 v\_percentual NUMBER;  
 v\_risco VARCHAR2(10);  
BEGIN  
 SELECT ocupacao, capacidade  
 INTO v\_ocupacao, v\_capacidade  
 FROM GS\_ABRIGO  
 WHERE id = p\_id;  
  
 v\_percentual := (v\_ocupacao / v\_capacidade) \* 100;  
  
 IF v\_percentual < 50 THEN  
 v\_risco := 'BAIXO';  
 ELSIF v\_percentual <= 80 THEN  
 v\_risco := 'MÉDIO';  
 ELSE  
 v\_risco := 'ALTO';  
 END IF;  
  
 RETURN v\_risco;  
END;



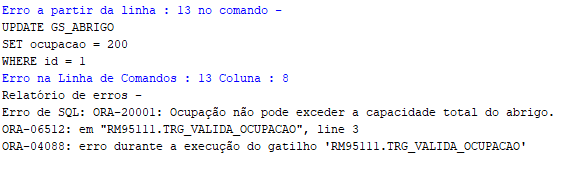
* Função - Ranking do Abrigo por Ocupação

CREATE OR REPLACE FUNCTION fn\_posicao\_abrigo(p\_id IN NUMBER)  
RETURN NUMBER IS  
 v\_posicao NUMBER := 0;  
BEGIN  
 FOR r IN (  
 SELECT id  
 FROM GS\_ABRIGO  
 ORDER BY ocupacao DESC  
 ) LOOP  
 v\_posicao := v\_posicao + 1;  
 IF r.id = p\_id THEN  
 RETURN v\_posicao;  
 END IF;  
 END LOOP;  
 RETURN NULL; -- Caso não encontre o abrigo  
END;



* Trigger - Validação de Ocupação Máxima

CREATE OR REPLACE TRIGGER trg\_valida\_ocupacao  
BEFORE INSERT OR UPDATE ON GS\_ABRIGO  
FOR EACH ROW  
BEGIN  
 IF :NEW.ocupacao > :NEW.capacidade THEN  
 RAISE\_APPLICATION\_ERROR(-20001, 'Ocupação não pode exceder a capacidade total do abrigo.');  
 END IF;  
END;



* Cursor com LOOP - Exibir nome e risco de cada abrigo ativo

DECLARE  
 CURSOR c\_abrigos IS  
 SELECT id, nome FROM GS\_ABRIGO WHERE ativo = 'S';  
  
 v\_id GS\_ABRIGO.id%TYPE;  
 v\_nome GS\_ABRIGO.nome%TYPE;  
 v\_risco VARCHAR2(10);  
BEGIN  
 OPEN c\_abrigos;  
 LOOP  
 FETCH c\_abrigos INTO v\_id, v\_nome;  
 EXIT WHEN c\_abrigos%NOTFOUND;  
  
 v\_risco := fn\_risco\_abrigo(v\_id);  
  
 DBMS\_OUTPUT.PUT\_LINE('Abrigo: ' || v\_nome || ' | Risco: ' || v\_risco);  
 END LOOP;  
 CLOSE c\_abrigos;  
END;

Texto

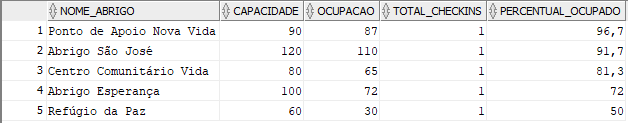
O conteúdo gerado por IA pode estar incorreto.

* Bloco Anônimo – Verificar se Abrigo está Ativo

DECLARE  
 v\_id NUMBER := 1; -- ID do abrigo a ser testado  
 v\_status CHAR(1);  
BEGIN  
 SELECT ativo INTO v\_status  
 FROM GS\_ABRIGO  
 WHERE id = v\_id;  
  
 IF v\_status = 'S' THEN  
 DBMS\_OUTPUT.PUT\_LINE('Abrigo ID ' || v\_id || ' está ATIVO e pode receber pessoas.');  
 ELSE  
 DBMS\_OUTPUT.PUT\_LINE('Abrigo ID ' || v\_id || ' está INATIVO no sistema.');  
 END IF;  
EXCEPTION  
 WHEN NO\_DATA\_FOUND THEN  
 DBMS\_OUTPUT.PUT\_LINE('Abrigo ID ' || v\_id || ' não encontrado.');  
END;



* Relatório – Quantidade de Check-ins por Abrigo com Percentual de Ocupação

SELECT  
 a.nome AS nome\_abrigo,  
 a.capacidade,  
 a.ocupacao,  
 COUNT(c.id) AS total\_checkins,  
 ROUND((a.ocupacao / a.capacidade) \* 100, 1) AS percentual\_ocupado  
FROM  
 GS\_ABRIGO a  
LEFT JOIN  
 GS\_CHECKIN c ON c.abrigo\_id = a.id  
GROUP BY  
 a.nome, a.capacidade, a.ocupacao  
ORDER BY  
 percentual\_ocupado DESC;

# 4 CONSULTAS SQL

* Abrigos com Total de Check-ins, Sensores e Último Alerta

SELECT

a.nome AS nome\_abrigo,

COUNT(DISTINCT c.id) AS total\_checkins,

COUNT(DISTINCT s.id\_sensor) AS total\_sensores,

MAX(al.data\_alerta) AS ultimo\_alerta

FROM

GS\_ABRIGO a

LEFT JOIN

GS\_CHECKIN c ON c.abrigo\_id = a.id

LEFT JOIN

GS\_SENSOR s ON s.id\_abrigo = a.id

LEFT JOIN

GS\_ALERTA al ON al.id\_abrigo = a.id

GROUP BY

a.nome

ORDER BY

total\_checkins DESC;

Interface gráfica do usuário, Aplicativo, Tabela

O conteúdo gerado por IA pode estar incorreto.

* Últimas Leituras por Sensor com Nome e Região do Abrigo

SELECT

ab.nome AS nome\_abrigo,

ab.regiao,

se.tipo\_sensor,

ls.valor,

ls.data\_leitura

FROM

GS\_LEITURA\_SENSOR ls

JOIN

GS\_SENSOR se ON ls.id\_sensor = se.id\_sensor

JOIN

GS\_ABRIGO ab ON se.id\_abrigo = ab.id

WHERE

ls.data\_leitura >= SYSDATE - 7 -- últimos 7 dias

ORDER BY

ls.data\_leitura DESC;

Interface gráfica do usuário

O conteúdo gerado por IA pode estar incorreto.

* Agrupamento de Usuários por Tipo (Gestor x Cidadão)

SELECT

tipo\_usuario,

COUNT(\*) AS total\_usuarios

FROM

GS\_USUARIO

GROUP BY

tipo\_usuario

ORDER BY

total\_usuarios DESC;

Tabela

O conteúdo gerado por IA pode estar incorreto.

* Comparação: Quantidade de Check-ins vs. Ocupação Registrada

SELECT

a.nome AS nome\_abrigo,

COUNT(c.id) AS checkins\_realizados,

a.ocupacao AS ocupacao\_registrada,

(a.ocupacao - COUNT(c.id)) AS diferenca

FROM

GS\_ABRIGO a

LEFT JOIN

GS\_CHECKIN c ON c.abrigo\_id = a.id

GROUP BY

a.nome, a.ocupacao

ORDER BY

diferenca DESC;

Uma imagem contendo Interface gráfica do usuário

O conteúdo gerado por IA pode estar incorreto.

# 5 USO DO MONGODB

Para complementar o armazenamento relacional feito em Oracle, foi utilizada uma base de dados não relacional MongoDB com o objetivo de registrar informações históricas de ocupação dos abrigos ao longo do tempo. Este tipo de dado, por ser semiestruturado e crescente, não se encaixa bem no modelo relacional tradicional. Por isso, a escolha pelo MongoDB foi estratégica e adequada.

A collection criada no MongoDB chama-se historico\_ocupacao, dentro da base abrigo. Nela, são armazenados documentos contendo os seguintes campos:

* id\_abrigo: identificador do abrigo relacionado
* nome: nome do abrigo
* ocupacao\_atual: número de pessoas presentes no momento
* capacidade\_total: limite máximo do abrigo
* percentual\_ocupado: cálculo da ocupação em percentual
* registrado\_em: data e hora do registro

Esses documentos são inseridos periodicamente, simulando snapshots horários da situação dos abrigos. Com isso, é possível realizar análises futuras sobre evolução de ocupação, prever superlotação e até alimentar painéis gráficos com séries temporais.

# 6 CONCLUSÃO

O desenvolvimento do projeto de banco de dados para o sistema de gestão de abrigos em desastres naturais permitiu aplicar de forma prática os principais conceitos estudados ao longo da disciplina. A modelagem relacional normalizada, a criação das tabelas e a implementação da package PL/SQL com procedures, funções, cursores, triggers e relatórios demonstram domínio das técnicas de banco de dados estruturado em Oracle.

Além disso, a integração com MongoDB trouxe um diferencial importante para o sistema, ao permitir o armazenamento de dados históricos de ocupação dos abrigos — algo que seria menos eficiente em um banco relacional. Essa separação entre dados estruturados e não estruturados mostra uma arquitetura flexível e moderna, adequada a sistemas reais.

Com as consultas SQL avançadas e a organização dos dados, é possível gerar relatórios que ajudam na tomada de decisão e no acompanhamento da operação dos abrigos. Dessa forma, o projeto se mostra viável, escalável e alinhado com os objetivos de oferecer suporte tecnológico em contextos críticos como desastres naturais.