

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

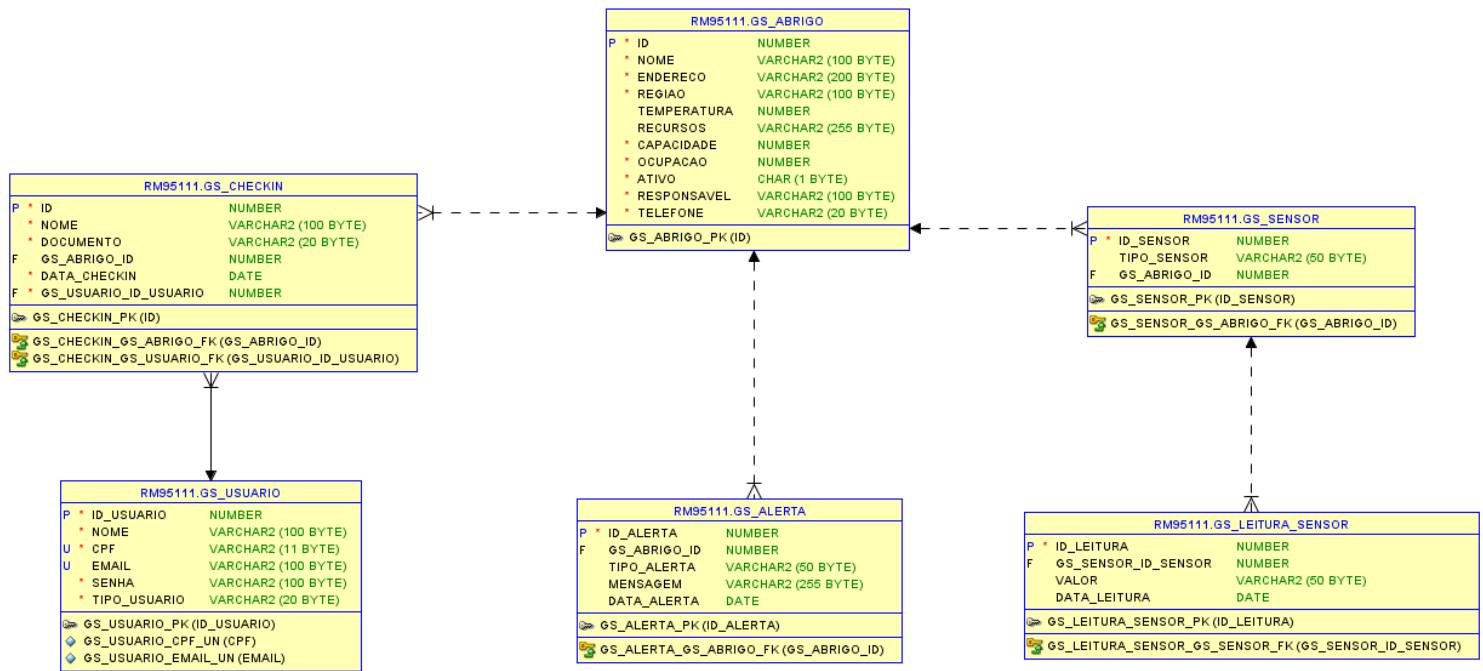
1 INTRODUÇÃO

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



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;
```

	RISCO
1	ALTO

- 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;
```

	RANKING
1	2

- 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;
```

Erro a partir da linha : 13 no comando -

```
UPDATE GS_ABRIGO
SET ocupacao = 200
WHERE id = 1
Erro na Linha de Comandos : 13 Coluna : 8
Relatório de erros -
Erro de SQL: ORA-20001: Ocupação não pode exceder a capacidade total do abrigo.
ORA-06512: em "RM95111.TRG_VALIDA_OCUPACAO", line 3
ORA-04088: erro durante a execução do gatilho 'RM95111.TRG_VALIDA_OCUPACAO'
```

- 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;
```

```
Abrigo: Ponto de Apoio Nova Vida | Risco: ALTO
Abrigo: Abrigo Esperança | Risco: MÉDIO
Abrigo: Centro Comunitário Vida | Risco: ALTO
Abrigo: Refúgio da Paz | Risco: MÉDIO
Abrigo: Abrigo São José | Risco: ALTO
```

- 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;
```

```
Abrigo ID 1 está ATIVO e pode receber pessoas.
```

- 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;
```

⚡ NOME_ABRIGO	⚡ CAPACIDADE	⚡ OCUPACAO	⚡ TOTAL_CHECKINS	⚡ PERCENTUAL_OCUPADO
1 Ponto de Apoio Nova Vida	90	87	1	96,7
2 Abrigo São José	120	110	1	91,7
3 Centro Comunitário Vida	80	65	1	81,3
4 Abrigo Esperança	100	72	1	72
5 Refúgio da Paz	60	30	1	50

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;
```

	Nome_Abrigo	Total_Checkins	Total_Sensores	Ultimo_Alerta
1	Abrigo Esperança	1	1	01/06/25
2	Abrigo São José	1	1	01/06/25
3	Refúgio da Paz	1	1	01/06/25
4	Ponto de Apoio Nova Vida	1	1	01/06/25
5	Centro Comunitário Vida	1	1	01/06/25

- 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;
```

	NOME_ABRIGO	REGIAO	TIPO_SENSOR	VALOR	DATA_LEITURA
1	Abrigo São José	Zona Oeste	Sensor de Movimento	Detectado	01/06/25
2	Refúgio da Paz	Zona Leste	Sensor de Qualidade do Ar	Boa	01/06/25
3	Centro Comunitário Vida	Zona Norte	Sensor de Fumaça	0.02 ppm	01/06/25
4	Abrigo Esperança	Zona Sul	Sensor de Umidade	58%	01/06/25
5	Ponto de Apoio Nova Vida	Centro	Sensor de Temperatura	27.4	01/06/25

- 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;
```

	TIPO_USUARIO	TOTAL_USUARIOS
1	cidadão	3
2	gestor	2

- 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;
```

	NOME_ABRIGO	CHECKINS_REALIZADOS	OCUPACAO_REGISTRADA	DIFERENCA
1	Abrigo São José	1	110	109
2	Ponto de Apoio Nova Vida	1	87	86
3	Abrigo Esperança	1	72	71
4	Centro Comunitário Vida	1	65	64
5	Refúgio da Paz	1	30	29

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.