CHALLENGE - ODONTOPREV

Turmas - TDSPB e TDSPS

análise e desenvolvimento de sistemas

MASTERING RELATIONAL AND NON RELATIONAL DATABASE

Larissa Araújo

RM 96496

Larissa Lopes

RM 552628

Luna Faustino

RM 552473

Sumário

[1. Objetivo do Projeto 3](#_Toc200317332)

[2. Escopo 3](#_Toc200317333)

[3. Requisitos Funcionais 4](#_Toc200317334)

[I. **Cadastro de Pessoas em Abrigos** 4](#_Toc200317335)

[II. **Visualização de Abrigos Próximos** 4](#_Toc200317336)

[III. **Consulta de Situação dos Abrigos** 4](#_Toc200317337)

[IV. **Busca por Desaparecidos** 5](#_Toc200317338)

[V. **Envio de Relatos Livres** 5](#_Toc200317339)

[VI. **Indicação de Rota Segura (Simulada)** 5](#_Toc200317340)

[VII. **Controle de Recursos por Abrigo (via API)** 5](#_Toc200317341)

[VIII. **Geração de Relatórios e Indicadores (via API)** 5](#_Toc200317342)

[4. Requisitos Não Funcionais 5](#_Toc200317343)

[I. Desempenho 5](#_Toc200317344)

[II. Disponibilidade 5](#_Toc200317345)

[III. Usabilidade 6](#_Toc200317346)

[IV. Segurança 6](#_Toc200317347)

[V. Escalabilidade 6](#_Toc200317348)

[VI. Portabilidade 6](#_Toc200317349)

[VII. Integração com Banco de Dados 6](#_Toc200317350)

[VIII. Confiabilidade 6](#_Toc200317351)

[5. Modelo CONCEITUAL 7](#_Toc200317352)

[Entidades: 7](#_Toc200317353)

[I. T\_GSAB\_PESSOA 7](#_Toc200317354)

[II. T\_GSAB\_USUARIO 7](#_Toc200317355)

[III. T\_GSAB\_TIPO\_USUARIO 7](#_Toc200317356)

[IV. T\_GSAB\_ABRIGO 7](#_Toc200317357)

[V. T\_GSAB\_CHECK\_IN 7](#_Toc200317358)

[VI. T\_GSAB\_RECURSO 8](#_Toc200317359)

[VII. T\_GSAB\_ESTOQUE\_RECURSO 8](#_Toc200317360)

[VIII. T\_GSAB\_ENDERECO 8](#_Toc200317361)

[IX. T\_GSAB\_CIDADE 8](#_Toc200317362)

[X. T\_GSAB\_ESTADO 8](#_Toc200317363)

[XI. T\_GSAB\_PAIS 9](#_Toc200317364)

[RELACIONAMENTOS: 9](#_Toc200317365)

[6. DIAGRAMAS 0](#_Toc200317366)

[7. Execução dos scripts 0](#_Toc200317367)

[Scripts 0](#_Toc200317368)

[8. INFORMAÇÕES 1](#_Toc200317369)

[9. LINKS 1](#_Toc200317370)

# Objetivo do Projeto

O objetivo deste projeto é desenvolver um sistema integrado para gerenciamento de abrigos emergenciais e apoio à evacuação de populações em áreas de risco durante eventos extremos, como enchentes, deslizamentos e desastres naturais. A solução visa fornecer uma plataforma eficiente e acessível tanto para autoridades quanto para a população, permitindo:

* O mapeamento e visualização de abrigos disponíveis com rotas seguras.
* O cadastro e controle de pessoas acolhidas nos abrigos.
* A gestão de recursos e estoques essenciais (como água, alimentos, remédios).
* A geração de relatórios e indicadores para tomada de decisão em tempo real.
* A busca por desaparecidos e acompanhamento da situação dos abrigos.

A proposta inclui uma aplicação web para uso administrativo e uma aplicação mobile voltada ao público geral, promovendo agilidade, transparência e organização no atendimento emergencial.

# Escopo

O projeto tem como escopo o desenvolvimento de uma aplicação mobile voltada para auxiliar a população durante eventos extremos, como enchentes, deslizamentos e outros desastres naturais. A solução visa fornecer informações em tempo real, promover a evacuação segura e facilitar o acesso a abrigos emergenciais. Entre as principais funcionalidades da aplicação, destacam-se:

* Visualização de abrigos próximos com base na localização do usuário.
* Indicação de rotas seguras para chegada aos abrigos (simulada).
* Consulta da situação dos abrigos, incluindo capacidade atual, responsáveis e recursos disponíveis.
* Cadastro de pessoas nos abrigos e controle de ocupação.
* Busca por pessoas desaparecidas por nome, CPF ou cidade.
* Canal para envio de relatos livres e informações relevantes por parte dos usuários.
* Geração de indicadores e relatórios sobre a ocupação dos abrigos e consumo de recursos (via interface administrativa).

O projeto também inclui:

* Modelagem e implementação do banco de dados relacional (Oracle) para informações estruturadas.
* Uso de banco de dados NoSQL (MongoDB) para armazenar dados não estruturados, como relatos livres, logs e metadados de imagens.
* Criação de procedures, funções e triggers para validação e consistência dos dados.
* Integração dos dados com a aplicação mobile via API.

A aplicação será projetada com foco em acessibilidade, usabilidade e agilidade no atendimento às necessidades emergenciais.

# Requisitos Funcionais

A seguir, estão listados os principais requisitos funcionais da aplicação mobile desenvolvida para gerenciamento de abrigos e evacuação durante eventos extremos:

### **Cadastro de Pessoas em Abrigos**

* Permitir o registro de pessoas acolhidas, incluindo nome, CPF, data de nascimento, condição médica e status de desaparecimento.

### **Visualização de Abrigos Próximos**

* Exibir em mapa ou lista os abrigos mais próximos ao usuário com base na localização atual.

### **Consulta de Situação dos Abrigos**

* Permitir que o usuário visualize informações detalhadas de um abrigo, como:
  + Capacidade total e atual.
  + Nome do responsável.
  + Recursos disponíveis.

### **Busca por Desaparecidos**

* Permitir a busca de pessoas desaparecidas por nome, CPF, cidade ou abrigo.

### **Envio de Relatos Livres**

* Permitir que usuários enviem mensagens, relatos e observações de forma livre, que serão armazenadas em banco NoSQL.

### **Indicação de Rota Segura (Simulada)**

* Exibir uma rota sugerida do usuário até o abrigo mais próximo (simulada, sem cálculo dinâmico).

### **Controle de Recursos por Abrigo (via API)**

* Visualização dos recursos cadastrados, quantidade disponível e estimativa de consumo.

### **Geração de Relatórios e Indicadores (via API)**

* Visualização de indicadores como:
  + Total de pessoas desaparecidas.
  + Ocupação média dos abrigos.
  + Recursos mais consumidos.

# Requisitos Não Funcionais

Os requisitos não funcionais a seguir garantem a qualidade, desempenho e segurança da aplicação mobile desenvolvida:

### Desempenho

* A aplicação deve responder às requisições do usuário em no máximo 2 segundos para ações comuns (consultas, cadastros, visualizações).

### Disponibilidade

* A aplicação deve estar disponível 24 horas por dia, principalmente durante situações de emergência.

### Usabilidade

* A interface deve ser intuitiva, com navegação simples, acessível para pessoas com diferentes níveis de familiaridade com tecnologia.

### Segurança

* Os dados pessoais devem ser protegidos com autenticação e uso de criptografia no tráfego (HTTPS).
* Restrições de acesso devem ser aplicadas de acordo com o tipo de usuário (ex: administrador vs. cidadão).

### Escalabilidade

* O sistema deve ser capaz de lidar com um aumento súbito de usuários e dados durante eventos críticos, utilizando infraestrutura escalável.

### Portabilidade

A aplicação deve ser compatível com dispositivos Android e iOS, utilizando frameworks multiplataforma (ex: React Native com Expo).

### Integração com Banco de Dados

* O sistema deve se comunicar com bancos de dados relacionais (Oracle) e não relacionais (MongoDB) via APIs, garantindo sincronização e consistência dos dados.

### Confiabilidade

* A aplicação deve ter baixo índice de falhas, com tratamento de erros adequado para garantir a continuidade do uso mesmo em situações imprevistas.

# Modelo CONCEITUAL

## Entidades:

### T\_GSAB\_PESSOA

* Atributos: id\_pessoa, nm\_pessoa, nr\_cpf, dt\_nascimento, ds\_condicao\_medica, st\_desaparecido
* Chave Primária: id\_pessoa

### T\_GSAB\_USUARIO

* Atributos: id\_usuario, nm\_usuario, ds\_email, ds\_senha, ds\_codigo\_google, id\_tipo\_usuario
* Chave Primária: id\_usuario
* Chave Estrangeira: id\_tipo\_usuario (relaciona com a entidade Tipo de Usuário)

### T\_GSAB\_TIPO\_USUARIO

* Atributos: id\_tipo\_usuario, ds\_tipo
* Chave Primária: id\_tipo\_usuario

### T\_GSAB\_ABRIGO

* Atributos: id\_abrigo, nm\_abrigo, nr\_capacidade, nr\_ocupacao\_atual, id\_endereco, id\_usuario\_responsavel
* Chave Primária: id\_abrigo
* Chaves Estrangeiras: id\_endereco (relaciona com Endereço), id\_usuario\_responsavel (relaciona com Usuário)

### T\_GSAB\_CHECK\_IN

* Atributos: id\_checkin, dt\_entrada, dt\_saida, id\_abrigo, id\_pessoa
* Chave Primária: id\_checkin
* Chaves Estrangeiras: id\_abrigo (relaciona com Abrigo), id\_pessoa (relaciona com Pessoa)

### T\_GSAB\_RECURSO

* Atributos: id\_recurso, ds\_recurso, qt\_pessoa\_dia, st\_consumivel
* Chave Primária: id\_recurso

### T\_GSAB\_ESTOQUE\_RECURSO

* Atributos: id\_estoque\_recurso, id\_recurso, id\_abrigo, qt\_disponivel
* Chave Primária: id\_estoque\_recurso
* Chaves Estrangeiras: id\_recurso (relaciona com Recurso), id\_abrigo (relaciona com Abrigo)

### T\_GSAB\_ENDERECO

* Atributos: id\_endereco, ds\_cep, ds\_logradouro, nr\_numero, ds\_complemento, id\_cidade
* Chave Primária: id\_endereco
* Chave Estrangeira: id\_cidade (relaciona com Cidade)

### T\_GSAB\_CIDADE

* Atributos: id\_cidade, nm\_cidade, id\_estado
* Chave Primária: id\_cidade
* Chave Estrangeira: id\_estado (relaciona com Estado)

### T\_GSAB\_ESTADO

* Atributos: id\_estado, nm\_estado, id\_pais
* Chave Primária: id\_estado
* Chave Estrangeira: id\_pais (relaciona com País)

### T\_GSAB\_PAIS

* Atributos: id\_pais, nm\_pais
* Chave Primária: id\_pais

## RELACIONAMENTOS:

* Pessoa - Check-in:

Uma pessoa pode ter vários check-ins, cada check-in pertence a uma única pessoa.

* Abrigo - Check-in:

Um abrigo pode ter vários check-ins, cada check-in ocorre em apenas um abrigo.

* Abrigo - Estoque de Recursos:

Um abrigo pode ter vários recursos no estoque, cada item do estoque está vinculado a um recurso e a um abrigo.

* Usuário - Abrigo:

Cada abrigo possui um usuário responsável, mas um mesmo usuário pode ser responsável por vários abrigos.

* Endereço - Abrigo:

Cada abrigo está vinculado a um endereço único.

* Cidade - Estado - País (via Endereço):

Um endereço pertence a uma cidade, que pertence a um estado, que pertence a um país.

* Usuário - Tipo de Usuário:

Um tipo de usuário pode ser associado a vários usuários, mas cada usuário pertence a apenas um tipo.

# DIAGRAMAS

Diagrama Lógico – Abrigue-se

Diagrama

O conteúdo gerado por IA pode estar incorreto.

Figure 1- Diagrama Lógico

Diagrama Relacional – Abrigue-se

Diagrama

O conteúdo gerado por IA pode estar incorreto.

Figure 2 - Diagrama Relacional

# Execução dos scripts

## Scripts

Os scripts estão nessa pagina do git hub:  
<https://github.com/larissalaga/GS-2025-DATABASE/blob/main/SQL's/Packages/PackageUnica.sql>

# INFORMAÇÕES

Foi criado um script em Python para geração de dados para popular as tabelas criadas, com instruções no “Read-me” nesse link :

<https://github.com/larissalaga/GS-2025-DATABASE/tree/main/InsercaoDadosUsandoPython>

# LINKS

<https://github.com/larissalaga/GS-2025-DATABASE>

https://youtu.be/06MwPdLChfU