README - GeoRelato

# 🌍 GeoRelato - Sistema de Registro de Catástrofes Naturais

> Um sistema inteligente de monitoramento e registro de eventos naturais dentro de um raio geográfico definido.

![GeoRelato Banner](https://img.freepik.com/free-vector/global-warming-banner-template\_23-2148620392.jpg)

## 💡 Visão Geral

\*\*GeoRelato\*\* é uma aplicação de linha de comando (CLI) desenvolvida em Python que permite o \*\*registro, consulta e organização de relatos de catástrofes naturais\*\* em um raio de até 10 km a partir de um ponto central. É ideal para auxiliar equipes de monitoramento em situações emergenciais.

## 🚀 Funcionalidades

* - 📍 \*\*Cadastro de relatores\*\* (nome, documento, email, telefone, localização).
* - 🌪️ \*\*Registro de relatos\*\*: tipo, descrição, data, hora e localização.
* - 🎯 \*\*Validação geográfica\*\*: apenas relatos dentro de um raio permitido são aceitos.
* - 🔍 \*\*Buscas inteligentes\*\*:
* - Por tipo de catástrofe
* - Por intervalo de datas
* - Por proximidade geográfica

## 🧠 Justificativas Técnicas

### ✔️ Organização por Classes

* - \*\*Relator\*\*: abstração de quem reporta o evento, separando dados pessoais e localização.
* - \*\*Relato\*\*: encapsula os dados do evento, com vínculo opcional ao relator.
* - \*\*SistemaCadastro\*\*: núcleo do sistema que gerencia armazenamento, buscas e validações.

### 🧭 Validação Geográfica com `geopy`

from geopy.distance import geodesic

* - Garante que os relatos respeitem um \*\*raio geográfico real\*\*, aumentando a precisão da triagem de eventos.

### 🧾 Armazenamento em JSON

* - Simples e legível.
* - Facilita o transporte e persistência dos dados.
* - Usa `json.dump` e `json.load` com estruturas personalizadas para serializar objetos.

### ⚡ CLI com `Typer`

import typer

* - Permite criar comandos acessíveis direto do terminal com ajuda automática (`--help`).
* - Substitui o uso de menus interativos por uma interface mais \*\*profissional, limpa e modular\*\*.

### 🧪 Estrutura de Busca Eficiente

* - Usa listas e dicionários para buscas por:
* - Tipo de evento (`List[Relato]`)
* - Datas com `datetime.strptime`
* - Localizações por distância geodésica

## 🛠️ Como Usar

### 1. Instale as dependências:

pip install typer geopy

### 2. Execute os comandos:

➕ Cadastrar relator:

python main.py adicionar-relator "Maria Silva" "123456789" "maria@email.com" "11999999999" -23.5 -46.6

🌀 Cadastrar relato:

python main.py adicionar-relato "enchente" "Rua alagada após chuva" "2025-06-01" "14:00" -23.51 -46.62 "123456789"

🔍 Buscar por tipo:

python main.py buscar-por-tipo "enchente"

🗓️ Buscar por período:

python main.py buscar-por-periodo "2025-05-01" "2025-06-01"

📍 Buscar por localização:

python main.py buscar-por-localizacao -23.5 -46.6 5

## 📂 Estrutura do Projeto

GeoRelato/

│

├── main.py # Script principal

├── dados.json # Armazena relatos

├── relatores.json # Armazena relatores

└── README.md # Você está aqui!

## 🎓 Aprendizados e Destaques

✅ Uso de orientação a objetos para manter organização

✅ Serialização/deserialização completa entre objetos Python e JSON

✅ Integração com bibliotecas reais de geolocalização

✅ Uso de CLI moderna e amigável com `Typer`

## ✨ Contribuições Futuras

* - Adicionar suporte a mapas visuais com `folium` ou `plotly`.
* - Criar uma API REST para acessar os relatos via HTTP.
* - Interface gráfica leve com `Tkinter` ou `PyWebIO`.

## 🧑‍💻 Autor

Desenvolvido por \*\*Lucas Andrade Souza\*\*

🔗 [Repositório no GitHub](https://github.com/lucasouza06/Desafio-GeoRelato)

> 📬 Para dúvidas, contribuições ou sugestões, entre em contato por email ou GitHub!