Persistência Poliglota com MongoDB e SQLite

Alunos: Lucas Bezerra, Gabriel de Almeida e Victor de Almeida Professor: Rircardo Roberto

Explicação da Arquitetura Adotada

O projeto utiliza uma arquitetura de persistência poliglota, combinando dois bancos de dados distintos:

1. SQLite:

- Utilizado para armazenar informações relacionadas às cidades.
- É um banco de dados relacional leve, ideal para armazenar dados estruturados.

2. MongoDB:

- Utilizado para armazenar informações sobre locais de interesse.
- É um banco de dados NoSQL, adequado para dados semiestruturados, como coordenadas geográficas e descrições.

Fluxo Geral da Aplicação

- A aplicação foi desenvolvida utilizando Streamlit, uma biblioteca Python para criação de interfaces web interativas.
- O sistema é dividido em quatro funcionalidades principais:
 - 1. Cadastro de Cidades: Permite inserir cidades no banco SQLite.
 - Cadastro de Locais: Permite inserir locais de interesse no MongoDB.
 - 3. **Consulta de Locais**: Exibe os locais cadastrados em uma cidade, utilizando mapas interativos com a biblioteca **Folium**.
 - 4. **Geoprocessamento**: Realiza cálculos de distância e busca de locais próximos com base em coordenadas fornecidas.

Integração de Tecnologias

- Streamlit: Interface gráfica para interação com o usuário.
- SQLite: Persistência de dados relacionais.
- MongoDB: Persistência de dados semi-estruturados.
- Folium: Geração de mapas interativos.
- Streamlit-Folium: Integração de mapas Folium na interface Streamlit.

Exemplos de Consultas Realizadas

1. Cadastro de Cidades

- Entrada: Nome da cidade e estado.
- Processo: Os dados s\u00e3o enviados para o banco SQLite.
- Exemplo:
 - o Entrada: Cidade: João Pessoa, Estado: Paraíba.
 - Resultado: A cidade é cadastrada no banco de dados.

2. Cadastro de Locais

- Entrada: Nome do local, cidade, latitude, longitude e descrição.
- Processo: Os dados são enviados para o MongoDB.
- Exemplo:
 - Entrada: Local: Praia de Tambaú, Cidade: João Pessoa, Latitude:
 -7.11532, Longitude: -34.861, Descrição: Praia urbana famosa na cidade.
 - Resultado: O local é cadastrado no banco de dados.

3. Consulta de Locais

- Entrada: Seleção de uma cidade cadastrada.
- Processo: Os locais associados à cidade são exibidos em um mapa interativo.
- Exemplo:
 - Entrada: Cidade: João Pessoa.
 - Resultado: Exibição de locais como marcadores no mapa.

4. Geoprocessamento

Busca de Locais Próximos:

- o Entrada: Coordenadas de referência e raio de busca.
- Resultado: Locais dentro do raio especificado são exibidos no mapa.

Cálculo de Distância:

- Entrada: Coordenadas de dois pontos.
- Resultado: Distância entre os pontos em quilômetros.

Configuração do MongoDB:

Para que o sistema funcione corretamente, é necessário configurar o MongoDB e importar os dados iniciais. Siga os passos abaixo:

1. Instalar o MongoDB

Baixe e instale o MongoDB Community Server a partir do site oficial: https://www.mongodb.com/try/download/community. Após a instalação, inicie o serviço do MongoDB.

2. Criar a Collection

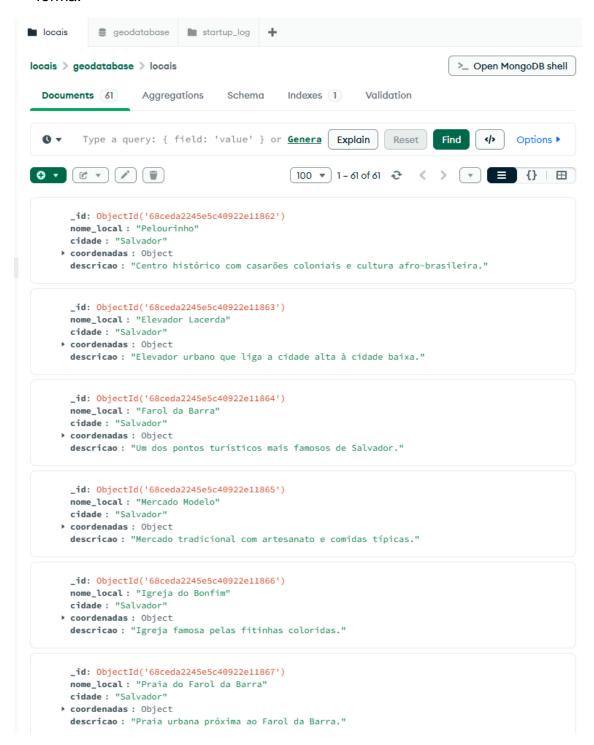
Abra o MongoDB Compass e Crie uma nova Connection com o Nome de Locais.

3. Importar o Arquivo locais_bahia.json

Abra o MongoDB Compass e Clique no simbolo de + para importar o .Json presente no diretorio atual.

4. Verificar os Dados Importados

Após importado, você vai verificar que seu Compass ficará da seguinte forma:



Prints do sistema em funcionamento.





