

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 semi-estruturados, 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.
 2. **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 são enviados para o banco SQLite.
- **Exemplo:**
 - 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:**
 - 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:

The screenshot shows the MongoDB Compass interface. The sidebar on the left has three items: 'locais', 'geodatabase', and 'startup_log'. The main area is titled 'locais > geodatabase > locais' and has a button 'Open MongoDB shell'. Below this is a tab bar with 'Documents' (61), 'Aggregations', 'Schema', 'Indexes' (1), and 'Validation'. The 'Documents' tab is active, showing a query bar with the text 'Type a query: { field: 'value' } or Genera' and buttons 'Explain', 'Reset', 'Find', and 'Options'. Below the query bar is a toolbar with icons for adding, editing, deleting, and refreshing documents, along with a dropdown for '100' and a pagination indicator '1 - 61 of 61'. The main area displays a list of 61 documents, each with the following fields: `_id`, `nome_local`, `cidade`, `coordenadas`, and `descricao`. The documents are as follows:

- `_id`: ObjectId('68ceda2245e5c40922e11862')
`nome_local`: "Pelourinho"
`cidade`: "Salvador"
`coordenadas`: Object
`descricao`: "Centro histórico com casarões coloniais e cultura afro-brasileira."
- `_id`: ObjectId('68ceda2245e5c40922e11863')
`nome_local`: "Elevador Lacerda"
`cidade`: "Salvador"
`coordenadas`: Object
`descricao`: "Elevador urbano que liga a cidade alta à cidade baixa."
- `_id`: ObjectId('68ceda2245e5c40922e11864')
`nome_local`: "Farol da Barra"
`cidade`: "Salvador"
`coordenadas`: Object
`descricao`: "Um dos pontos turísticos mais famosos de Salvador."
- `_id`: ObjectId('68ceda2245e5c40922e11865')
`nome_local`: "Mercado Modelo"
`cidade`: "Salvador"
`coordenadas`: Object
`descricao`: "Mercado tradicional com artesanato e comidas típicas."
- `_id`: ObjectId('68ceda2245e5c40922e11866')
`nome_local`: "Igreja do Bonfim"
`cidade`: "Salvador"
`coordenadas`: Object
`descricao`: "Igreja famosa pelas fitinhas coloridas."
- `_id`: ObjectId('68ceda2245e5c40922e11867')
`nome_local`: "Praia do Farol da Barra"
`cidade`: "Salvador"
`coordenadas`: Object
`descricao`: "Praia urbana próxima ao Farol da Barra."

Prints do sistema em funcionamento.

