

Relatório

Trabalho Prático 1

Bernardo Alves, Bernardo Venancio, Lucas Albuquerque

DCC-UFMG
Prof. Renato Vimieiro

Introdução

Este relatório apresenta o desenvolvimento de um sistema interativo para consulta ortogonal de bares e restaurantes em Belo Horizonte, utilizando árvores k-dimensionais (KD-Trees) para busca eficiente em dados espaciais. A ferramenta permite ao usuário selecionar uma área retangular no mapa da cidade e visualizar os estabelecimentos contidos nessa região, com informações complementares exibidas em uma janela.

O projeto foi implementado em Python e integra bibliotecas de visualização geográfica com técnicas de Geometria Computacional, visando a criação de uma aplicação funcional, responsiva e alinhada aos conceitos estudados na disciplina.

Descrição do Problema

A Prefeitura de Belo Horizonte disponibiliza uma base de dados com os registros de atividades econômicas no município. O problema consiste em permitir que o usuário visualize, filtre e consulte estabelecimentos comerciais (bares e restaurantes) a partir de uma área retangular desenhada no mapa da cidade.

Para isso, é necessário:

- Obter os dados dos estabelecimentos e suas coordenadas geográficas.
- Construir um mapa interativo com marcadores (pins) dos estabelecimentos.
- Implementar uma KD-Tree com busca ortogonal (range search).
- Permitir ao usuário desenhar retângulos no mapa e consultar apenas os estabelecimentos dentro da área selecionada.
- Exibir os resultados em tempo real de forma visual e interativa.

Etapas da Implementação

1. Geração do Mapa de Belo Horizonte

A primeira etapa do projeto foi a geração de um mapa da cidade de Belo Horizonte a partir da API do OpenStreetMap. Utilizamos um código em Python para gerar um

arquivo HTML contendo o mapa, centrado nas coordenadas da cidade.

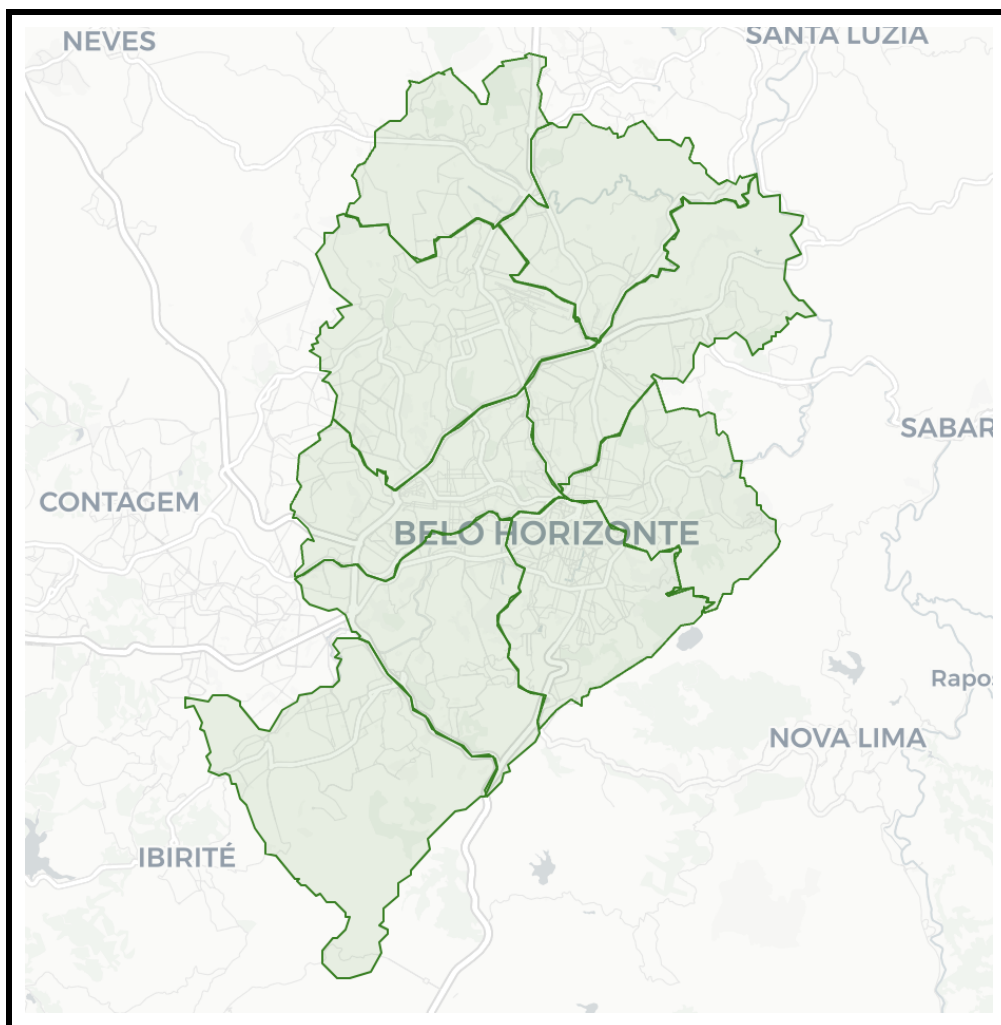


Figura 1: Mapa de BH delimitado pelas Regiões.

A biblioteca utilizada foi `folium`, que fornece integração direta com o `Leaflet.js`. Esse mapa inicial foi hospedado na plataforma `Render`, permitindo acesso via navegador.

2. Coleta e Tratamento dos Dados

Obtivemos os dados públicos da Prefeitura de Belo Horizonte no formato `CSV`, referentes aos estabelecimentos ativos em abril de 2025.

Aplicamos um filtro na coluna de CNAE principal, mantendo apenas aqueles com descrições que indicam bares e restaurantes. Em seguida, realizamos a geocodificação dos endereços utilizando a API do `OpenStreetMap` para converter os dados de endereço em coordenadas (latitude e longitude).

Foram mantidos os seguintes atributos por estabelecimento:

- Nome ou nome fantasia
- Endereço formatado
- Data de início das atividades

- Situação do alvará
- Coordenadas geográficas

3. Desenho de Retângulo no Mapa

Para possibilitar a seleção de regiões no mapa, implementamos uma funcionalidade em **JavaScript** que permite ao usuário desenhar retângulos interativamente.

A biblioteca **Leaflet.draw** foi utilizada para essa funcionalidade. Ao término do desenho, as coordenadas (latitude mínima, máxima; longitude mínima, máxima) do retângulo são capturadas e enviadas a um endpoint **Python** para processamento.

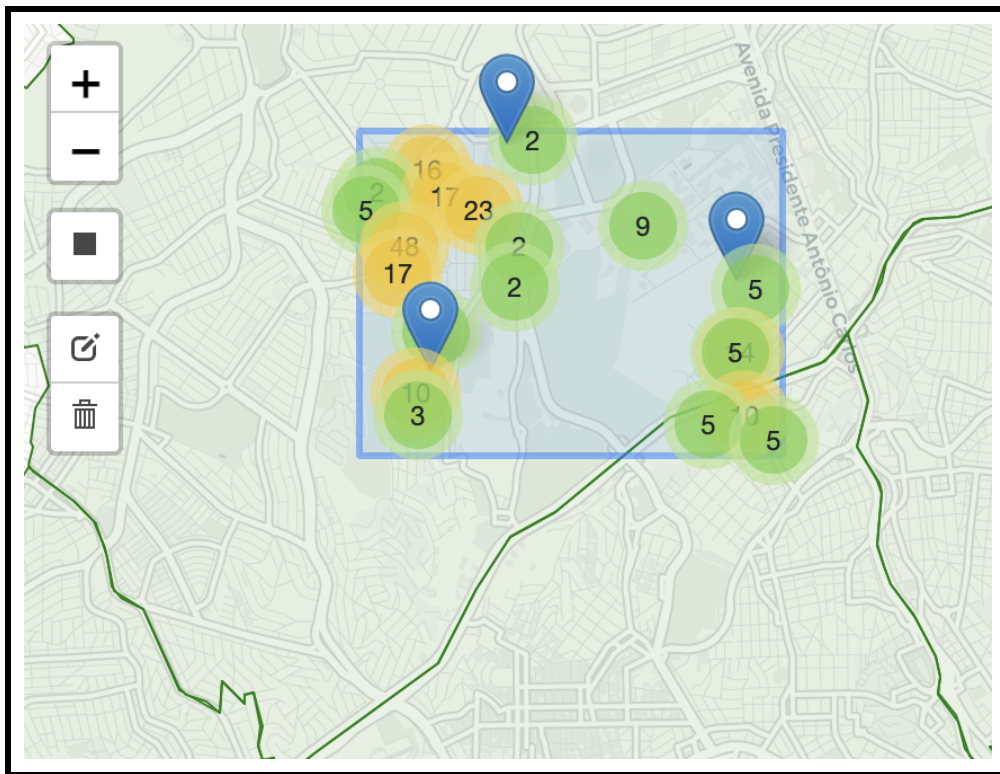


Figura 2: Desenhando o retângulo.

No canto esquerdo da Figura 2, é mostrado as funcionalidades interativas do mapa, como desenhar o retângulo de busca, dar Zoom, ou então salvar ou apagar um retângulo já criado.

4. KD-Tree e Range Search

Em **Python**, desenvolvemos uma estrutura de dados **KD-Tree** com suporte a dois eixos: latitude e longitude.

- No pré-processamento, os pontos são ordenados alternadamente por cada eixo, e a árvore é construída recursivamente.
- A operação de **range search** é usada para buscar todos os pontos que estão dentro das coordenadas do retângulo desenhado.

O código é eficiente e permite resposta quase instantânea ao filtro espacial do usuário.

Após a execução da busca, os estabelecimentos localizados dentro do retângulo selecionado são destacados no mapa com **Pins interativos**. Esses pins são **clusterizados automaticamente**, facilitando a visualização em regiões densas.

Ao clicar em um pino individual, um **popup informativo** é exibido com os seguintes dados do estabelecimento:

- Nome do bar ou restaurante;
- Endereço formatado;
- Data de início das atividades;
- Indicação se possui alvará de funcionamento.



Figura 3: Pin dos estabelecimentos, e os **Clusters** com os números de bares/restaurantes juntos .

Pontos Extras: Integração com o *Comida di Buteco*

Como funcionalidade adicional, realizamos a integração dos dados do festival gastronômico *Comida di Buteco* com os dados de bares e restaurantes da PBH. Essa funcionalidade incluiu desde a obtenção das informações até sua exibição enriquecida no mapa interativo.

1. Coleta e Processamento dos Dados

- Utilizamos **web scraping em Python** para extrair os dados diretamente do HTML do site oficial do festival.

- Foram coletadas informações como:
 - Nome dos estabelecimentos;
 - Nome do prato participante;
 - Descrição do prato;
 - Imagens dos pratos.
- As correspondências com os dados da PBH foram feitas com diferentes abordagens:
 - Comparação por **nome fantasia**;
 - Comparação por **endereço**;
 - Quando possível, comparação por **CNPJ**.
- Os casos que não puderam ser associados automaticamente foram tratados manualmente com inspeção visual.

2. Visualização Diferenciada no Mapa

- Os estabelecimentos identificados como participantes do festival foram destacados com **Pins dourados**, diferenciando-os dos demais.



Figura 4: Exemplo de prato participante destacado no mapa.

- Cada um desses pins contém um **popup personalizado**, com:
 - Nome do prato;
 - Descrição do prato;
 - Imagem do prato (coletada via scraping).

- Essa visualização diferenciada melhora a experiência do usuário e enriquece a interface com informações gastronômicas relevantes.

3. Benefícios e Considerações

- A funcionalidade extra proporciona um exemplo prático de integração de dados abertos com dados coletados automaticamente da web.
- Exige tratamento cuidadoso de inconsistências e ambiguidade nos dados.
- O resultado é uma aplicação mais informativa, atrativa e útil para o usuário final.

Link do Repositório

O código-fonte completo, bem como os dados utilizados e exemplos de uso, estão disponíveis em:

<https://github.com/bernardoam04/TP-ALG2>