

Plataforma Didática para ensino de Estruturas de Dados Multidimensionais

Bruno Porto Masquio
Universidade do Estado do Rio de Janeiro – UERJ
brunomasquio@gmail.com

Paulo Eustáquio Duarte Pinto
Universidade do Estado do Rio de Janeiro – UERJ
pauloedp@ime.uerj.br

Resumo – O objetivo deste projeto é desenvolver, em uma única plataforma livre, um ambiente didático para estudo das estruturas de dados multidimensionais, utilizadas em Sistemas de Informações Geográficas. A sua infraestrutura se baseia na linguagem de programação C++, softwares livres para modelagem geométrica e visual, constantes das ferramentas Qt creator, Graphviz e DOT. A interface do programa permite a visualização gráfica dos dados, os passos para a execução dos algoritmos relacionados às estruturas de maneira clara e didática. Além disso, serão feitos testes nas diferentes estruturas a fim de compará-las em termos de desempenho. Acredita-se que, com esse projeto, será mais fácil e mais rápido o estudo dessas estruturas de dados, pela visualização possibilitada pelo ambiente.

Palavras chave – Estruturas de dados multidimensionais, árvores

I. INTRODUÇÃO

O tratamento de Sistemas de Informações Geográficas (SIG)[1] só recentemente vem recebendo maior atenção, pois necessita de estruturas de dados de alta complexidade e relativamente pouco estudadas e conhecidas. Dentre elas, existem as estruturas multidimensionais, que têm como objetivo principal a interação com dados espaciais como, por exemplo, mapas com coordenadas.

Na plataforma desenvolvida, são usadas as estruturas de dados multidimensionais propostas por Hanan Samet[2]. Pode-se citar, como exemplos, as estruturas KD-Tree, Point Quadtree, PR Quadtree, Range Tree, MX Quadtree, descritas adiante.

A ideia que norteou a criação da ferramenta deste trabalho foi a de apresentar, de forma didática, essas estruturas. Isso permite, em especial, que alunos pudessem visualizá-las de duas formas básicas: através de sua representação espacial e de sua representação interna. Essa última característica constitui um ponto forte deste trabalho, pois não foi encontrada em nenhuma das ferramentas pesquisadas.

II. DESCRIÇÃO DAS ESTRUTURAS DE DADOS MULTIDIMENSIONAIS JÁ IMPLEMENTADAS

A. KD-Tree

A KD-Tree é uma árvore binária com k dimensões. Cada nó não folha pode ser pensado como um hiperplano que divide implicitamente o espaço em duas partes, conhecidas como half-spaces. A direção do hiperplano é escolhida da seguinte maneira: cada nó da árvore está associado com uma das k dimensões, com o hiperplano perpendicular ao eixo dessa dimensão. Assim, por exemplo, se para um determinado

grupo o eixo "x" é escolhido, todos os pontos da subárvore com um valor menor de "x" do que o nó aparecem na subárvore esquerda e todos os pontos com maior "x" valor será na subárvore direita. Em tal caso, o hiperplano seria definido pelo valor x do ponto, e a sua normal seria o eixo- x .

As Figuras 1 e 2 mostram uma árvore criada, bem como a divisão espacial, para o seguinte conjunto de pontos: A(40,45), B(15,70), C(70,10), D(69,50), E(66,85) e F(85,90).

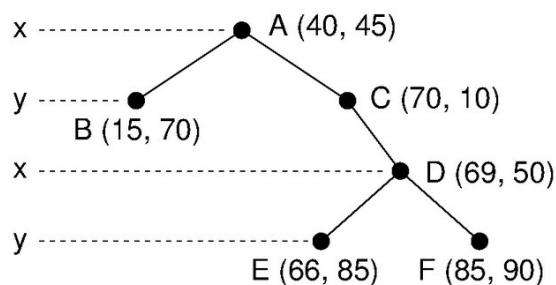


Fig 1. Hierarquia de nós da KD-Tree especificada

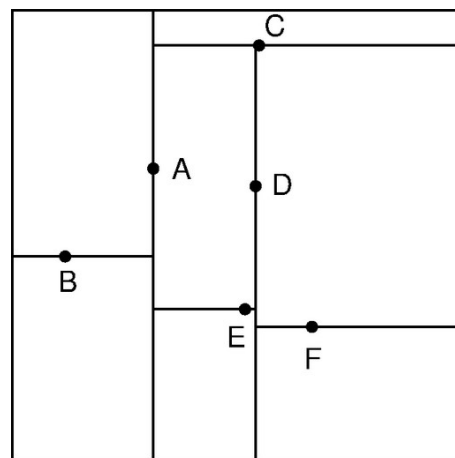


Fig 2. Divisão espacial da KD-Tree especificada

B. Range Tree

A Range Tree é uma extensão de uma árvore binária de busca, considerando cada uma das dimensões desejadas. Os nós folha representam os elementos e são duplamente encadeados enquanto os nós intermediários contêm o valor médio entre suas subárvores na dimensão usada. Cada nó intermediário possui uma subárvore associada para cada outra dimensão. Ela possui os mesmos nós folhas(elementos), porém seus nós intermediários guardam valores médios referentes às outras dimensões.

Essa estrutura de dados é muito eficiente para buscas, porém necessita de muita memória.

A Figura 3 ilustra uma Range Tree unidimensional para os pontos 6, 15, 17, 21, 24, 33 e 42. A Figura 4 ilustra, simplificada, uma Range Tree bidimensional, onde os nós maiores representam a árvore da primeira dimensão e os nós menores, os da segunda.

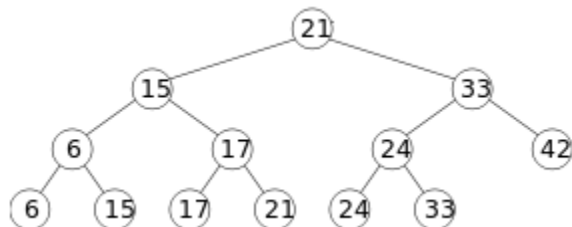


Fig 3. Exemplo de Range Tree de uma dimensão

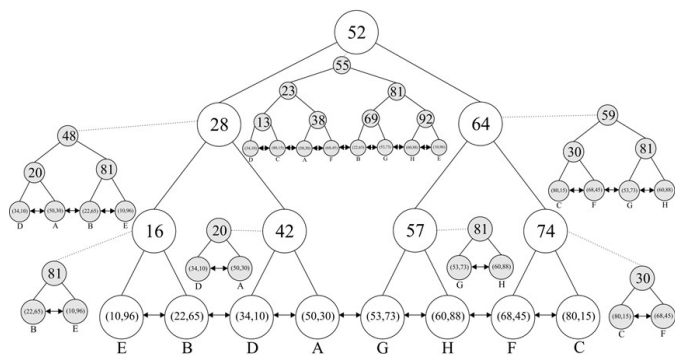


Fig 4. Exemplo de Range Tree de duas dimensões

C. Quadrees

Quadrees são árvores onde cada nó pai divide a área (ou subárea) definida por seus ascendentes em quatro quadrantes – situados respectivamente a nordeste, sudeste, noroeste e sudoeste das coordenadas 2D deste nó. Por outro lado, nós-folha não indicam subdivisão do espaço, havendo a garantia de que se um nó não teve seus quadrantes subdivididos seus filhos são nós-folha ou nulos. Há várias maneiras de implementar Quadrees. Mostraremos duas delas.

C.1. Point Quadrees

Nesta estrutura há, no máximo, um ponto por quadrante. A cada inserção de um nó, se o quadrante não dividido em que o nó inserido estiver localizado já contiver algum ponto o mesmo é desmembrado. Os quatro novos quadrantes são definidos a partir das coordenadas do ponto já anteriormente contido, como pode ser visto nas Figuras 5 e 6.

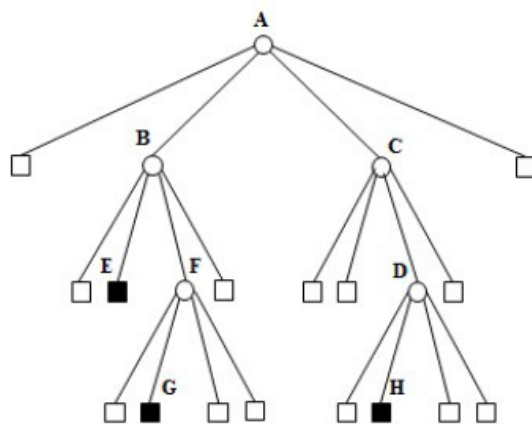


Fig 5. Exemplo de Point Quadtree (hierarquia de nós)

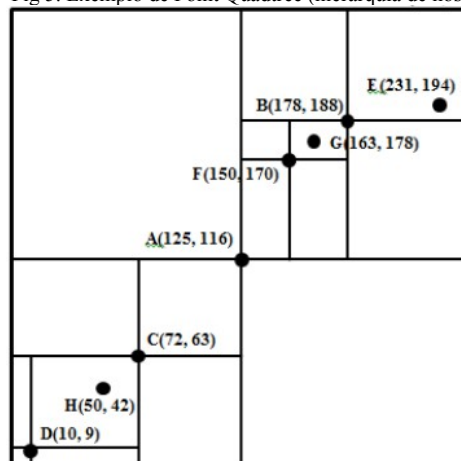


Fig 6. Exemplo de Point Quadtree (divisão espacial)

C.2. PR Quadtree

A PR Quadtree é estruturada analoga à Point Quadtree, com a ressalva de que o particionamento de quadrantes é feito de forma homogênea. Nessa estrutura, os elementos são guardados a princípio nos nós folha e os nós intermediários podem ser pensados como uma divisão simétrica do espaço em quatro quadrantes.

Apesar de ser mais estável que a Point Quadtree, pois sua construção não depende da ordem de inserção, a PR Quadtree pode não ser ideal para dados cujos nós são muito próximos, pois isso gera uma quantidade grande de nós intermediários, prejudicando tanto o desempenho em buscas quanto a quantidade de memória que é necessária.

As Figuras 7 e 8 ilustram, respectivamente, a estrutura de dados e a divisão do espaço para 5 pontos.

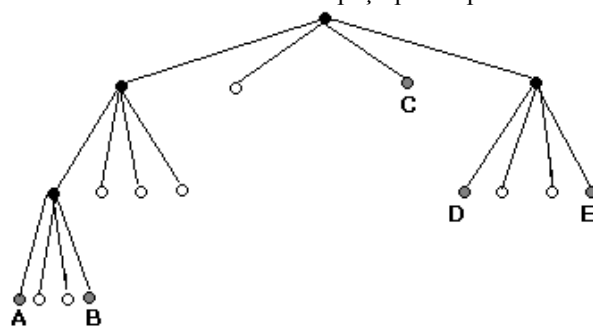


Fig 7. Exemplo de PR Quadtree (hierarquia de nós)

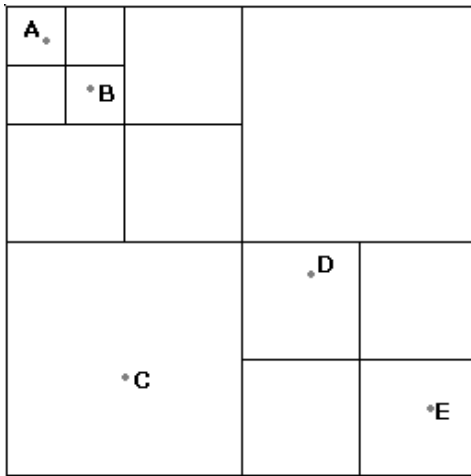


Fig 8. Exemplo de PR Quadtree (divisão espacial)

III. INFRAESTRUTURA DO SOFTWARE

Para o desenvolvimento do software de interface utilizado para representar graficamente a estrutura de dados multidimensionais, foi escolhida a linguagem de programação C++ devido ao seu potencial em manipular ponteiros, gerenciar memória e suportar programação orientada a objetos. Foi utilizado o framework multiplataforma Qt que suporta código em C++ e fornece uma grande quantidade de bibliotecas para implementar softwares com interface gráfica. Para representar os nós de forma hierárquica, foi utilizada a biblioteca para gerar grafos Graphviz, cujos gráficos a serem gerados foram codificados utilizando a linguagem de descrição de grafos DOT.

A. Linguagem DOT

DOT é uma linguagem de criação de grafos. Ela é codificada por um texto simples de uma forma que tanto nós quanto as máquinas são interpretados facilmente.

A sua sintaxe permite que se edite diversas características do grafo, como direcionamento de arestas, hierarquia dos vértices para visualização, uso de cores, legendas tanto para arestas quanto para vértices.

Essa linguagem é muito versátil e pode ser usada em qualquer aplicação que deseje se trabalhar com visualização de grafos. Em nossa plataforma, utilizamos o Graphviz para interpretação e geração de imagens.

As Figuras 9 e 10 ilustram grafos com características que podem ser implementadas através do DOT.

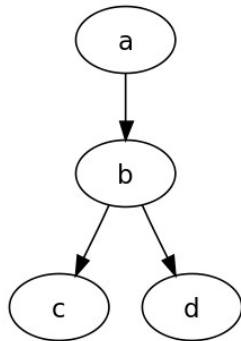


Fig 9. Exemplo 1 de grafo com o DOT

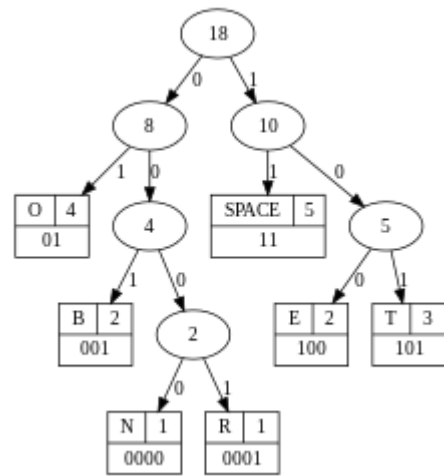


Fig 10. Exemplo 2 de grafo com o DOT

B. Graphviz

O nome Graphviz vem da abreviação de Graph Visualization Software (Software de Visualização de Grafos). É um pacote de ferramentas, de código aberto, desenvolvido nos laboratórios de pesquisa da AT&T para desenhar grafos especificados na linguagem DOT. O Graphviz possui algoritmos capazes de calcular automaticamente o melhor posicionamento dos nós e arestas de um grafo em uma imagem.

O formato da imagem é especificado pelo programador dentre uma variedade suportada bem grande, como jpg, png, jpeg, bpm, psd, pdf, gif dentre outros.

C. QT

Qt é um framework multiplataforma que permite compilar aplicações para Windows, Mac, Linux ou outros sistemas Unix. Suporta código em C++, possui código aberto, desenvolvido pela empresa norueguesa Trolltech (posteriormente adquirida pela Nokia). O Qt fornece uma grande quantidade de bibliotecas para implementar softwares com interface gráfica. Um exemplo de grande sucesso da utilização do Qt é o projeto KDE.

Para exibir os gráficos do resultado do particionamento do espaço e da representação hierárquica dos nós no software de interface, foram utilizadas algumas classes do Qt. As mais relevantes são: QGraphicsView, QGraphicsScene, QGraphicsItem, QByteArray e QPixmap.

Para gerar o gráfico do resultado do particionamento do espaço, foram inseridos objetos da classe QGraphicsItem (QGraphicsRectItem para representar pontos e QGraphicsLineItem para representar linhas de particionamento) em um QGraphicsScene, e posteriormente este QGraphicsScene é exibido em um QGraphicsView.

Para gerar o gráfico da representação hierárquica dos nós, é obtido um array de char contendo a imagem renderizada pelo Graphviz, convertido em um QByteArray e depois gerado um pixmap em um QPixmap. O QPixmap gerado é inserido em um QGraphicsScene, e posteriormente este QGraphicsScene é exibido em um QGraphicsView.

IV. FERRAMENTA DESENVOLVIDA

A. Interface

A.1 Tela inicial

A tela inicial do software consiste em uma janela que, além caracterizar o projeto, possibilita selecionar, de uma lista de estruturas disponíveis, aquela que será visualizada. A Figura 11 ilustra a atual tela inicial do programa.



Fig 11. Tela inicial do software

A.2. Tela inicial para cada estrutura de dados

Após escolhida a estrutura, estarão disponíveis um menu e três janelas (funções, particionamento do espaço e hierarquia dos nós) usadas para as operações e opções de visualização das estruturas de dados, conforme a Figura 12.

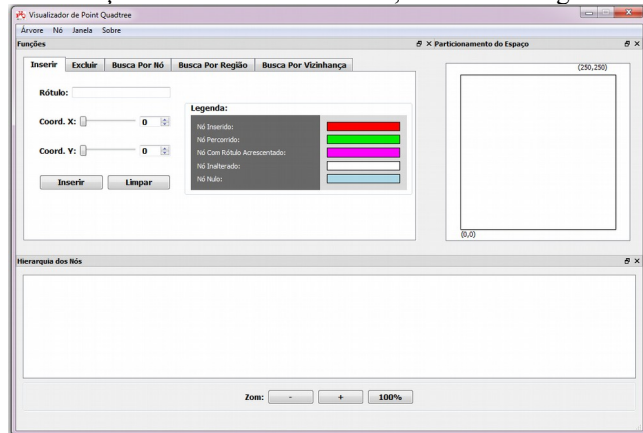


Fig 12. Tela inicial para cada estrutura de dados

A.3. Janelas desencaixáveis

Cada uma das três janelas (funções, particionamento do espaço e hierarquia dos nós) foi desenvolvida usando a classe QDockWidget do Qt, que permite o encaixe dentro da janela principal ou a flutuação de cada uma das janelas em um nível superior na área de trabalho. Cada QDockWidget consiste de uma barra de título e a área de conteúdo. Na barra de título cada janela foi rotulada e foram habilitados botões

para desencaixe e fechamento das janelas. Cada área de conteúdo foi utilizada para a criação das janelas de Funções, Particionamento do Espaço e Hierarquia dos Nós. É importante ressaltar que as janelas que podem ser desencaixadas podem ser ampliadas ou reduzidas, conforme for a melhor visualização para o usuário.

A Figura 13 ilustra as janelas que podem ser desencaixadas e expandidas.

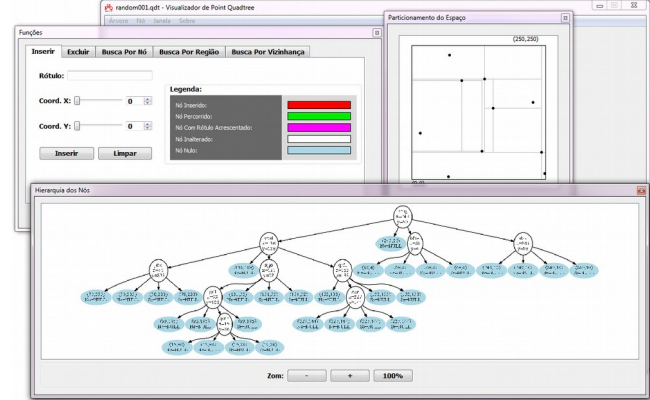


Fig 13. Janelas de funções, particionamento e hierarquia dos nós desencaixáveis.

B. Principais funcionalidades

O software possui diversas funcionalidades para manipulação dos dados e para buscas nas estruturas de dados. Durante esses processos, é possível visualizar graficamente através de cores a sequência de passos do algoritmo utilizado.

Nas buscas, os nós encontrados são exibidos na caixa de texto “Resultado” encontrada na aba “Funções” conforme a Figura 14, ou é exibida uma mensagem informativa se não houver um nó nas coordenadas correspondentes.

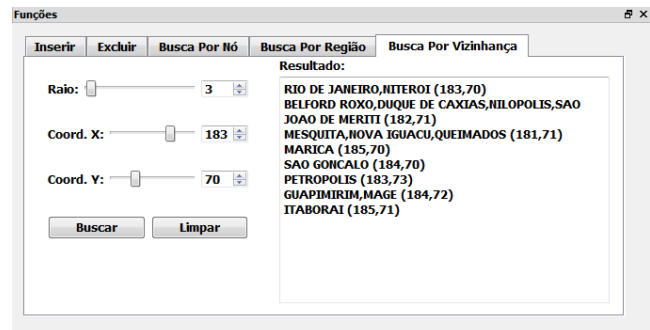


Fig 14. Aba funções exibindo os resultados de uma busca por vizinhança

B.1. Busca por nó

A busca por nó verifica a existência de um nó em coordenadas que foram informadas pelo usuário.

A Figura 15 ilustra uma busca por nó em uma PR Quadtree.

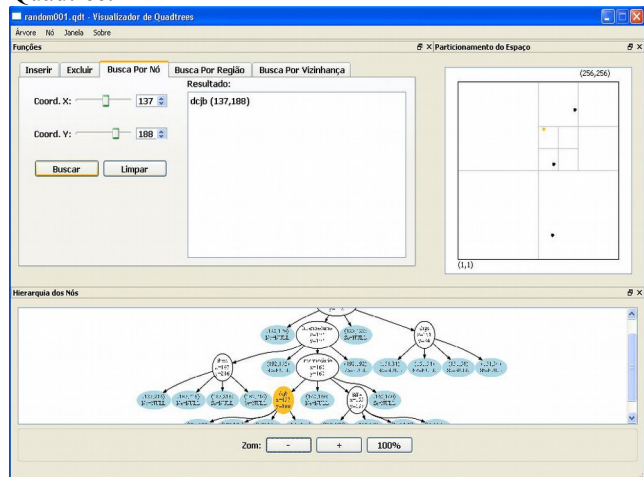


Fig 15. Busca por nó no software

B.2. Busca por região

A busca por região define um retângulo no espaço de particionamento e busca por pontos nessa região.

A Figura 16 ilustra uma busca por região em uma PR Quadtree.

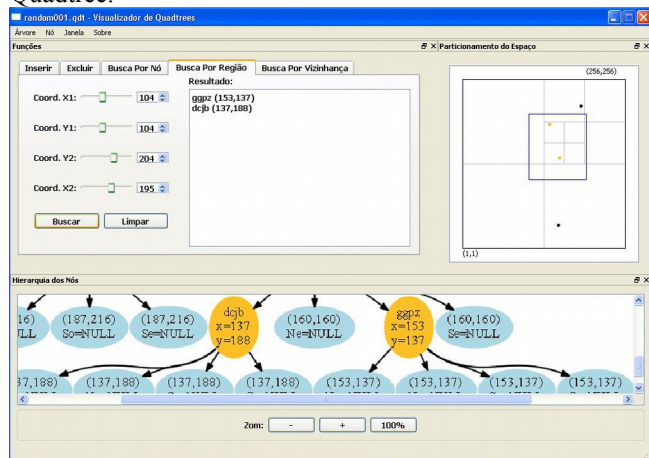


Fig 16. Busca por região no software

B.3. Busca por vizinhança

A busca por vizinhança realiza a pesquisa em um círculo definido por um ponto (centro da busca) e um raio que define a vizinhança a ser verificada.

A Figura 17 ilustra uma busca por vizinhança em uma Point Quadtree.

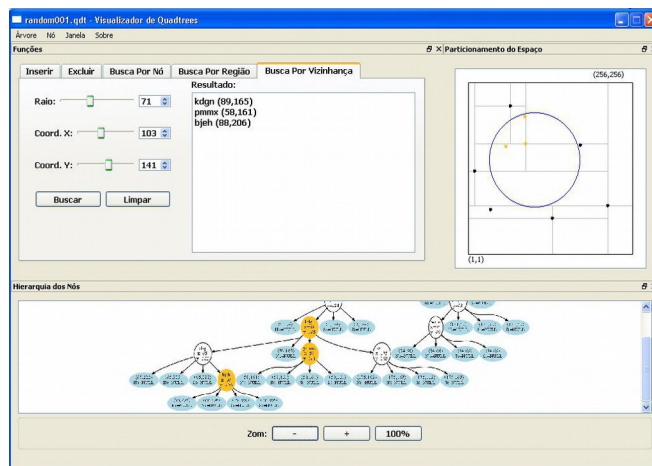


Fig 17. Busca por vizinhança no software

B.4. Geração randômica de dados

É possível criar uma árvore a partir de coordenadas geradas randômicamente. Para isso, é solicitado o número de nós ao usuário.

Essa função se encontra no menu “Árvore” e gera um arquivo com a árvore criada.

A Figura 18 ilustra a criação de uma Point Quadtree através de coordenadas randômicamente geradas e a janela para entrada da quantidade de nós.

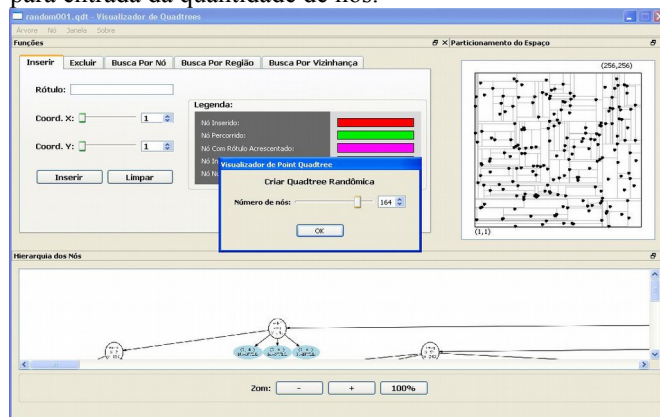


Fig 18. Geração de árvore randômica no software

B.5. Salvar e carregar arquivos

A manipulação dos dados por arquivos permite salvar uma árvore ou carregá-la a partir de um arquivo em memória secundária, podendo ele estar em coordenadas cartesianas conforme limites definidos pelo software ou em coordenadas UTM[3] conforme limites definidos pelos fusos e zonas do território brasileiro e verificados no momento em que um arquivo é carregado.

A Figura 19 ilustra uma busca feita em uma Point Quadtree que foi construída a partir de um arquivo em coordenadas UTM de todos os municípios do Brasil..

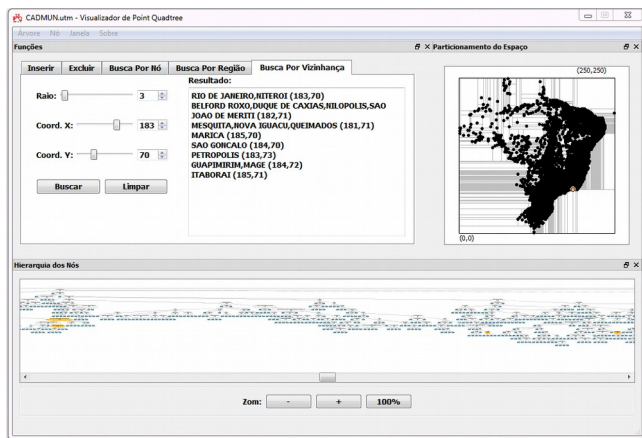


Fig 19. Busca em árvore carregada a partir de arquivo com coordenadas UTM dos municípios do Brasil.

V. CONCLUSÕES

Até o momento, o projeto em desenvolvimento já permite o estudo de quatro estruturas de dados para SIG, estando disponível para uso em disciplinas da graduação e mestrado no IME/UERJ. Embora só tenhamos avaliações qualitativas desse ambiente, ele parece ser bastante adequado à proposta inicial. Ressaltamos novamente, a característica única que possui de permitir a visualização interna das estruturas de dados, bem como ilustrar o efeito de muitas operações nessas estruturas.

No presente ano deveremos ampliar o leque de estruturas contempladas, bem como iniciar uma fase de comparação de desempenho dessas estruturas.

Após essa fase, pretendemos disponibilizar o ambiente como software livre.

REFERÊNCIAS

- [1] Sistemas de Informações Geográficas. Disponível em http://pt.wikipedia.org/wiki/Sistema_de_informação_geográfica - Visitado em 02/02/2015.
- [2] Samet, Hanan - Foundations of Multidimensional and Metric Data Structures. 1st edition. Local de publicação: San Francisco, CA. Morgan Kaufmann, 2006
- [3] Universal Transversa de Mercator. Disponível em http://pt.wikipedia.org/wiki/Universal_Transversa_de_Mercator - Visitado em 02/02/2015.
- [4] Disponível em <http://pt.wikipedia.org/wiki/Hiperplano> - Visitado em 02/02/2015.