

UNIVERSIDADE REGIONAL DE BLUMENAU
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
CURSO DE CIÊNCIAS DA COMPUTAÇÃO – BACHARELADO

DESENVOLVIMENTO DE UMA INFRA-ESTRUTURA
BASEADA EM FRACTAIS PARA VIABILIZAR A
CONSTRUÇÃO DE UM SISTEMA DE VISUALIZAÇÃO DE
SISTEMAS DE ARQUIVOS SEMÂNTICOS

RAFAEL EMMENDOERFER FALCONE

BLUMENAU
2009

2008/2-19

RAFAEL EMMEDOERFER FALCONE

**DESENVOLVIMENTO DE UMA INFRA-ESTRUTURA
BASEADA EM FRACTAIS PARA VIABILIZAR A
CONSTRUÇÃO DE UM SISTEMA DE VISUALIZAÇÃO DE
SISTEMAS DE ARQUIVOS SEMÂNTICOS**

Trabalho de Conclusão de Curso submetido à
Universidade Regional de Blumenau para a
obtenção dos créditos na disciplina Trabalho
de Conclusão de Curso II do curso de Ciências
da Computação — Bacharelado.

Prof. Mauro Marcelo Mattos, Doutor - Orientador

**BLUMENAU
2009**

2008/2-19

**DESENVOLVIMENTO DE UMA INFRA-ESTRUTURA
BASEADA EM FRACTAIS PARA VIABILIZAR A
CONSTRUÇÃO DE UM SISTEMA DE VISUALIZAÇÃO DE
SISTEMAS DE ARQUIVOS SEMÂNTICOS**

Por

RAFAEL EMMENDOERFER FALCONE

Trabalho aprovado para obtenção dos créditos
na disciplina de Trabalho de Conclusão de
Curso II, pela banca examinadora formada
por:

Presidente: _____
Prof. Mauro Marcelo Mattos, Doutor – Orientador, FURB

Membro: _____
Prof. Antônio Carlos Tavares, Mestre – FURB

Membro: _____
Prof. Dalton Solano dos Reis, Mestre – FURB

Blumenau, 11 de fevereiro de 2009

Dedico este trabalho à minha família que sempre acreditou em minha capacidade.

AGRADECIMENTOS

A Deus, por ser o provedor da nossa existência.

À minha família, por seu amor incondicional.

Aos meus amigos, por fazerem parte da minha vida.

A todos que de alguma forma contribuíram para a realização deste trabalho.

A Consistem Sistemas por sua flexibilidade e auxílio nos meus estudos.

Ao meu orientador, Mauro Marcelo Mattos, por ter sempre acreditado e me dado “empurrões” para a realização deste trabalho.

Duvidar de tudo ou crer em tudo. São duas soluções igualmente cômodas, que nos dispensam ambas de refletir.

Jules Henri Poincaré

RESUMO

Este trabalho tem como objetivo apresentar o desenvolvimento de uma ferramenta geradora de uma infra-estrutura baseada em fractais, para viabilizar a construção de um sistema de visualização de sistemas de arquivos semânticos. Uma ferramenta para a visualização tridimensional de um sistema de arquivos semânticos foi desenvolvida por Krauspenhar (2007). Esta apresenta uma interface onde é possível criar diversos objetos e associá-los a arquivos e/ou diretórios, onde uma alteração nestes componentes do sistema de arquivos influi em uma representação nos objetos associados. Tendo em vista o ambiente desenvolvido por Krauspenhar (2007) ser estático e refletir um universo restrito, pretende-se criar uma infra-estrutura que torne possível a representação de um volume de dados compatível com aquele utilizado nos sistemas de arquivos de sistemas operacionais para equipamentos *desktop* atuais.

Palavras-chave: Fractais. Sistemas de arquivos semânticos. Sistemas de arquivos.

ABSTRACT

This work has as objective to present the development of a generating tool of an infrastructure based on fractals, to make possible the construction of a system of visualization of systems of semantic files. A tool for three-dimensional visualization of a semantic file system was developed by Krauspenhar (2007). This presents an interface where you can create various objects and assign them to files and/or where a change in these components of the file system influences on a representation on associated objects. In view of the environment developed by Krauspenhar (2007) be static and reflect a limited universe, it is intended to create an infrastructure that makes possible the representation of a data volume compatible with that used in the file systems of operating systems for equipment current desktop.

Key-words: Fractals. Semantic file systems. File systems.

LISTA DE ILUSTRAÇÕES

Figura 1 – Esboço da representação do sistema de arquivos.....	12
Figura 2 – Conjunto de Mandelbrot	17
Figura 3 – Comportamento caótico	17
Figura 4 – Fractal “Origem do Universo”	20
Figura 5 – Curva de Von Koch.....	22
Figura 6 – Processo de construção da curva de Koch	23
Figura 7 – 3 iterações da construção da curva de Hilbert.....	25
Quadro 1 – Pacotes do JCKKit.....	26
Figura 8 – Diagrama dos pacotes JCKKit	26
Quadro 2 – Sistemas de coordenadas do JCKKit	27
Figura 9 – Representação do mapeamento DICS para DDCS	27
Figura 10 – Tela de execução do sistema de Krauspenhar.....	29
Figura 11 – Exemplo de uma árvore de diretórios	30
Figura 12 – Ambientes virtuais utilizados para o estudo de memória espacial.....	31
Figura 13 – Exemplo de tradução semântica.....	33
Figura 14 – Tela de execução do Task Gallery	34
Figura 15 – Tela de execução do BumpTop.....	35
Figura 16 – Visualização de diretórios no ambiente Tactile 3D	35
Figura 17 – Visualização de uma partição Unix no ambiente XCruiser	36
Figura 18 – Diagrama de classes de uso do ator usuário.....	38
Figura 19 – Diagrama de classes de uso do ator ambiente.....	39
Figura 20 – Diagrama das classes responsável pela estrutura de dados.....	39
Figura 21 – Diagrama das classes responsável pela geração do fractal	40
Figura 22 – Fractal representativo do diretório C:\Users\Falco\Documents\Visual Studio 2005	43
Figura 23 – Representação da fragmentação ocorrida de um nível para outro posterior	43
Figura 24 – Pontos do fractal usados para cálculo do tamanho das retas do próximo nível	44
Figura 25 – Código fonte do método gerarEstruturaDados	45
Figura 26- Tela de execução deste trabalho	47
Figura 27 – Fragmentos do fractal gerado neste trabalho	49

LISTA DE SIGLAS

DCS - *Data Coordinate System*

DDCS – *Device-Dependent Coordinate System*

DICS – *Device-Independent Coordinate System*

IBM - *International Business Machine*

JCCKit - *Java Chart Construction Kit*

PDA - *Personal Digital Assistant*

LISTA DE SÍMBOLOS

% - por cento

SUMÁRIO

1 INTRODUÇÃO.....	12
1.1 OBJETIVOS DO TRABALHO	14
1.2 ESTRUTURA DO TRABALHO	14
2 FUNDAMENTAÇÃO TEÓRICA	15
2.1 SISTEMAS DINÂMICOS	15
2.2 CAOS	15
2.3 FRACTAIS.....	18
2.3.1 Auto-similaridade.....	21
2.3.2 Dimensão fractal	22
2.3.3 Estimação da dimensão fractal.....	24
2.3.4 Iteração.....	24
2.4 JCCKIT.....	25
2.4.1 Visão geral	26
2.4.2 Sistema de coordenadas	27
3 FERRAMENTA DE VISUALIZAÇÃO 3D DE UM SISTEMA DE ARQUIVOS.....	28
3.1 INTRODUÇÃO.....	28
3.2 SISTEMA DE ARQUIVOS	29
3.2.1 Estrutura hierárquica	30
3.3 MEMÓRIA ESPACIAL EM AMBIENTES VIRTUAIS	31
3.4 ARQUIVOS SEMÂNTICOS.....	32
3.5 TRABALHOS CORRELATOS	33
4 DESENVOLVIMENTO.....	37
4.1 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO.....	37
4.2 ESPECIFICAÇÃO	37
4.2.1 Diagrama de casos de uso	38
4.2.2 Diagrama de classes	39
4.2.2.1 Classe Ponto.....	40
4.2.2.2 Classe PontoDiretorio	40
4.2.2.3 Classe FractalArvoreDir.....	41
4.3 IMPLEMENTAÇÃO	42
4.3.1 Estratégia para geração do fractal	42

4.3.2 Recursão.....	44
4.3.3 Operacionalidade da implementação	46
4.4 RESULTADOS E DISCUSSÃO	47
5 CONCLUSÕES.....	50
5.1 EXTENSÕES	50
REFERÊNCIAS BIBLIOGRÁFICAS	52

1 INTRODUÇÃO

Tendo em vista o grande número de arquivos disponibilizados e compartilhados na Internet e a capacidade crescente de armazená-los nas máquinas dos usuários, verificou-se a necessidade de desenvolverem-se formas alternativas de gerenciamento dos mesmos. Uma pesquisa realizada por Farhoomand e Drury (2002) para avaliar as habilidades de usuários gerenciarem dados, concluiu que 62% dos usuários tiveram dificuldade ou impossibilidade de gerenciar seus documentos.

Neste contexto Robertson et al. (1998, p. 153) relatou que a Microsoft desenvolveu um projeto denominado Data Mountain, o qual consiste na utilização de uma interface com o usuário criada especialmente para aproveitar a habilidade humana em guardar informações sobre um determinado ambiente e sua orientação espacial. Tendo como objetivo facilitar o gerenciamento de arquivos. Testes realizados pela própria Microsoft comprovaram que este modelo de gerência tornou-se uma alternativa viável (ROBERTSON et al., 1998, p. 160).

Inspirado pelos resultados do projeto Data Mountain surgiu a idéia de criar um ambiente de visualização 3D de um sistema de arquivos baseado nos princípios de memória espacial (UNIVERSITY OF BRISTOL, 2001). Um protótipo deste ambiente foi desenvolvido por Krauspenhar (2007).

O projeto de Krauspenhar (2007) utilizou a metáfora de uma cidade para representar num ambiente 3D objetos de sistema de arquivos. Neste mundo os arquivos são representados na forma de humanóides que se movimentam através de ruas para representar operações de movimentação de arquivos. A pasta onde os arquivos estão localizados é simbolizada por uma casa e o caminho que sai desde a raiz até a casa é representado por ruas.

A Figura 1 apresenta um esboço da representação do sistema de arquivos na metáfora.

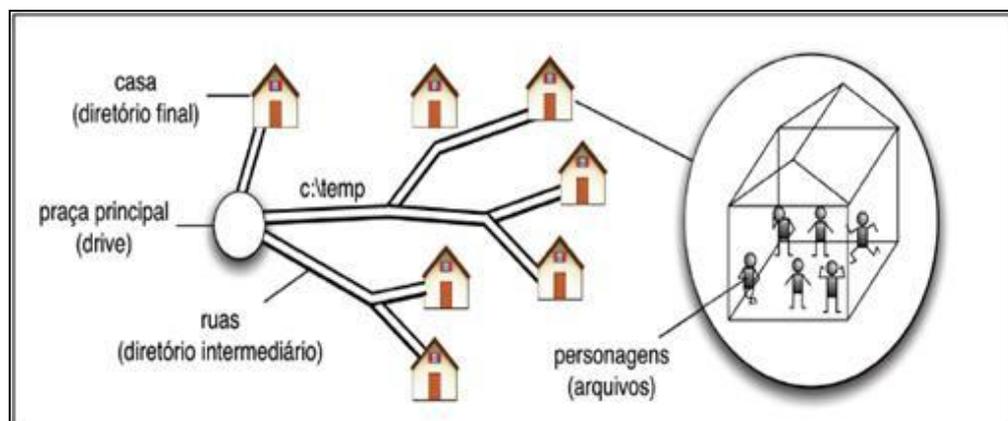


Figura 1 – Esboço da representação do sistema de arquivos

Uma das características do protótipo de Krauspenhar (2007) é que foram utilizados poucos dados para teste. De fato, em um sistema de arquivos real de um computador *desktop*, geralmente encontram-se milhares de arquivos e com frequência arquivos são criados e apagados. Esta dinâmica de funcionamento demandou a necessidade de identificação e desenvolvimento de uma infra-estrutura que permitisse não só representar a dinâmica de funcionamento do sistema de arquivos, mas também a dinâmica de funcionamento da metáfora selecionada.

Verificando-se este problema, decidiu-se investigar a possibilidade de utilização de estruturas fractais como substrato para viabilizar a construção de um modelo de testes com volume de dados mais próximo daquele encontrado na realidade.

Desta forma, pretende-se utilizar o modelo de fractais como base para o desenvolvimento desta infra-estrutura assumindo que, o modelo de árvore de diretórios segue um padrão que se assemelha a vários modelos fractais conhecido. Além disso, as operações de poda (exclusão de diretórios), ou enxerto (movimentação de diretórios de um ramo para outro) podem ser facilmente manipuladas nesta estrutura mantendo sempre uma ligação unária com a representação gráfica do ambiente descrito em Krauspenhar (2007).

O termo fractal surgiu no trabalho de Mandelbrot (1983), que considerava que a natureza não poderia ser descrita por meio da geometria convencional, dita Euclidiana. Para tanto, ele propôs um novo conceito de geometria chamada de geometria fractal. A principal característica do fractal está relacionada à sua dimensão, com a qual é possível determinar o grau de complexidade de uma linha ou a rugosidade de uma superfície.

A princípio, a escolha de modelos fractais ajusta-se aos objetivos do projeto iniciado por Mattos, Tavares e Krauspenhar (2007) (que é a construção de um sistema de visualização 3D para um sistema de arquivos), tendo em vista as características inerentes dos mesmos, como a auto-similaridade e a capacidade de representar estruturas complexas que a geometria Euclidiana não é capaz (LECHY-THOMPSON, 2001, p. 190). Sendo assim, analogamente, é possível a associação entre os objetos do sistema de arquivos, a estrutura fractal e o modelo de representação gráfica para exibição das informações.

A propriedade de auto-similaridade permite a criação de micro-universos de resolução infinita (LECHY-THOMPSON, 2001, p. 190). Esta característica associada ao modelo de metáfora descrito em Mattos, Tavares e Krauspenhar (2007) viabiliza a manutenção do *layout* da cidade mesmo com a inserção/remoção de novas pastas e/ou arquivos. Este aspecto é importante tendo em vista que a metáfora utilizada está baseada no conceito de memória espacial. Portanto, o modelo da cidade (figura 1) necessita ser mantido estável para que

possam ser obtidas as vantagens que o uso da técnica apresenta (HENDERSON, 2004; MCKAY; MARSHALL, 2001).

1.1 OBJETIVOS DO TRABALHO

O objetivo deste trabalho é desenvolver uma infra-estrutura que viabilize a construção de um ambiente de visualização de um sistema de arquivos usando algoritmos fractais.

Os objetivos específicos do trabalho são:

- a) identificar um modelo de estrutura fractal que viabilize a construção de um universo onde cada ponto folha represente uma referência a um arquivo físico;
- b) disponibilizar uma biblioteca que encapsule esta estrutura e permita a manipulação da mesma por aplicações externas;
- c) validar o modelo através da construção de um protótipo.

1.2 ESTRUTURA DO TRABALHO

O primeiro capítulo apresenta uma introdução do trabalho, os objetivos a serem apresentados e a estrutura do trabalho.

O segundo capítulo contempla a fundamentação teórica do trabalho e descreve o conceito de sistemas dinâmicos, caos e fractais. Além disto, descreve a biblioteca Java JCCKit utilizada para desenvolvimento da ferramenta apresentada e também será mostrado os trabalhos correlatos.

O terceiro capítulo descreve a ferramenta desenvolvida por Krauspenhar (2007) e seus principais conceitos.

A descrição da estratégia e principal técnica utilizada para gerar o fractal são descritos no quarto capítulo.

O quinto capítulo descreve as considerações finais sobre o trabalho, incluindo sugestões para extensões em trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo introduz de forma geral os conceitos de sistemas dinâmicos e o caos, assuntos ligados a fractais. Explica também o que são fractais e sua geometria e as principais características.

2.1 SISTEMAS DINÂMICOS

Segundo Peitgen et al. (1988 apud SANTOS, 1999, p. 5) um sistema dinâmico, basicamente, é qualquer processo que evolui no tempo. Os sistemas dinâmicos possuem um caráter multidisciplinar podendo ser encontrado em diversas áreas da ciência, como por exemplo: astronomia, biologia, ecologia, economia, engenharia química, física, meteorologia, entre outras disciplinas (DEVANEY, 1990 apud SANTOS, 1999, p. 5). Um sistema dinâmico pode ser encontrado em processos naturais ou gerado por um processo artificial.

As várias disciplinas, em que os sistemas dinâmicos podem ser aplicados, utilizam-se deste meio com o objetivo de prever possíveis resultados através dos seus processos de evolução. A matemática entra como ferramenta para exibir as características comportamentais desses sistemas, permitindo representar sua evolução.

Apesar dos sistemas dinâmicos se proporem prever estados futuros de um processo, isto não é o que ocorre com a maioria dos sistemas naturais. Estes sistemas não deterministas, que possuem comportamento aleatório e não previsível, são denominados de sistemas dinâmicos não-lineares. A idéia de caos surgiu com o estudo desses sistemas “aleatórios” e “imprevisíveis”.

2.2 CAOS

A teoria do caos entrou em voga somente nos anos 80 como fonte de explicação para uma série de fenômenos complexos da física. No entanto, como insistem os teóricos que trabalham nesse campo, o caos não é um novo paradigma ou um novo campo. Trata-se antes

de uma constatação do que uma descoberta, de uma compreensão mais ampla do que de uma teoria nova (CHERQUES, 1999).

O caos teve suas sementes lançadas no início da década de 1960, quando um meteorologista chamado Edward Lorenz desenvolveu modelos computacionais dos padrões do tempo. Como é de conhecimento geral, é muito difícil fazer uma previsão de tempo em longo prazo, ainda que se possa isolar muitos dos fatores que causam as mudanças. Lorenz, como outros, pensava que tudo o que era preciso para uma melhor previsão era um modelo mais abrangente. Então, escreveu um programa baseado em doze equações simples que em linhas gerais modelava os principais fatores que influenciam o tempo (BUFFONI, 2005; GLEICK, 1991).

Lorenz descobriu algo surpreendente: pequenas mudanças ou pequenos erros em um par de variáveis produziam efeitos tremendamente desproporcionais. Para um período de uns dois dias, elas mal faziam diferença; mas extrapolando-se para um mês ou mais, as mudanças produziam padrões completamente diferentes. Lorenz chamou sua descoberta de "efeito borboleta" (BUFFONI, 2005; GLEICK, 1991).

Para Rezende (1992, p. 29), o caos é um estado complexo caracterizado pela (aparente) imprevisibilidade de comportamento e por grande sensibilidade a pequenas mudanças nas variáveis do sistema ou nas condições iniciais. É observado tanto em sistemas muito simples quanto em sistemas complexos. A condição essencial para um sistema apresentar estado caótico é ser não-linear, isto é, apresentar uma resposta não proporcional ao estímulo.

O avanço dos estudos relacionado ao caos é consequência do avanço da capacidade de processamento dos computadores. Inovações nas técnicas de computação gráfica fizeram com que cientistas e matemáticos progredissem nos estudos dos sistemas dinâmicos que é de onde o caos deriva (CRILLY, et al. 1991 apud SANTOS, 1999, p. 6).

A incapacidade de se prever o resultado de um sistema dinâmico não está na quantidade de variáveis envolvidas no processo, e sim, na grande sensibilidade das condições iniciais destas variáveis, onde pequenas mudanças podem acarretar resultados totalmente inesperados (DEVANEY, 1990 apud SANTOS, 1999, p. 6).

Um exemplo de um sistema dinâmico que possui comportamento caótico é a função $f(z) = z^2 + c$, gerando um dos objetos mais fascinante e difícil da matemática que é o conjunto de Mandelbrot representado na Figura 2.

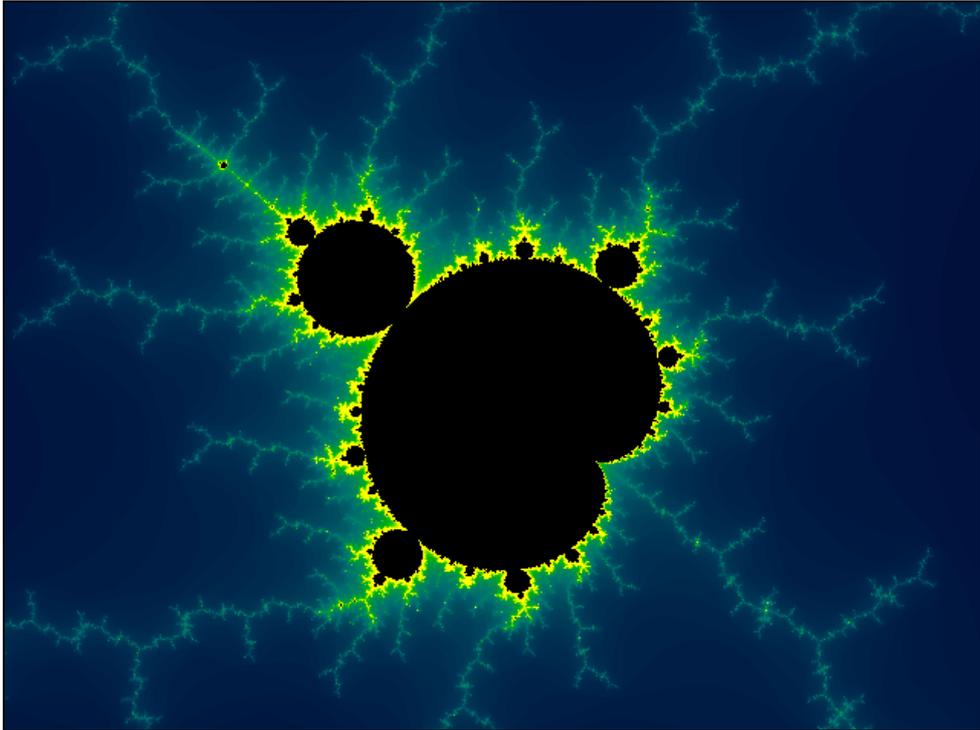
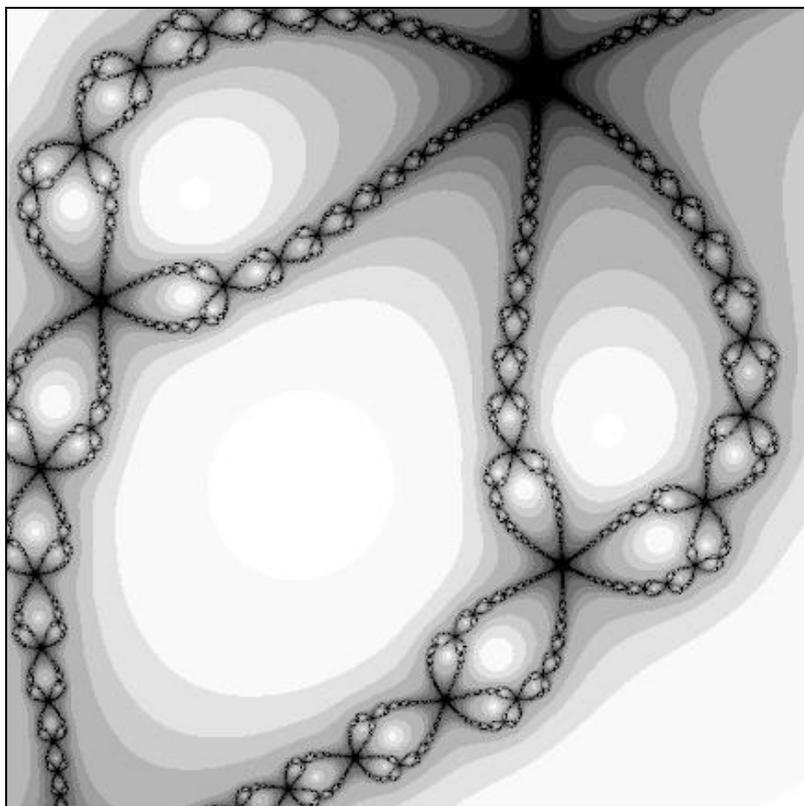


Figura 2 – Conjunto de Mandelbrot

Outro exemplo de sistema caótico pode ser gerado, observado na Figura 3, através do método de Newton para se achar raízes de equações com grau superior a dois, ou seja, valores para x que anulam equações sob a forma: $f(z) = \alpha_0 + \alpha_1 z + \alpha_2 z^2 + \alpha_2 z^3 + \dots + \alpha_m z^m = 0$.



Fonte: Santos (1999, p. 7).

Figura 3 – Comportamento caótico

Este método consiste em se estipular um valor inicial para a resposta e, a partir deste valor, iterar a função até que se aproxime o máximo possível da mesma, obtendo-se uma resposta numérica e um fator de erro (aproximação). O conjunto de Mandelbrot e o método de Newton são exemplos de funções que exibem comportamento caótico, exibindo características fractais (auto-similaridade) através de iterações em um plano complexo.

Segundo Buffoni (2005) um importante passo em direção à teoria do caos foi dado por volta de 1970, quando James A. Yorke e seu amigo, o biólogo Robert May, começaram a examinar as propriedades da assim chamada "equação logística" que, entre outras coisas, fornece um modelo simples para o crescimento da população. A maneira como essa equação funciona é que os resultados vão sempre alimentando a equação de modo a se obterem novos resultados. O interessante é que, dependendo de como é utilizado certo fator, os resultados podem se tornar altamente previsíveis ou altamente caóticos.

Mas até mesmo o caos da equação logística tem seu próprio tipo de padrão. Embora você não possa sempre prever qual será o resultado particular da equação, você sabe que ele vai cair em uma determinada faixa. Muitas outras equações se comportam de forma semelhante, produzindo o caos com uma tendência ou um modelo de organização - entre estas, estão as equações que predizem a turbulência em líquidos ou a subida e a queda dos preços do algodão (BUFFONI, 2005). Percebe-se que dentro de um processo aleatório existe um padrão, e mesmo dentro de um padrão pode existir uma aleatoriedade.

2.3 FRACTAIS

Num universo despovoado de formas geométricas perfeitas, onde proliferam superfícies irregulares, difíceis de representar e medir, a geometria fractal apresenta-se como um meio de tratar aqueles fenômenos até então considerados erráticos, imprevisíveis e aleatórios, numa palavra, caóticos (MESQUITA; MOTA, 1991).

Representar geometricamente as formas da natureza é um dos principais desafios enfrentados pelos matemáticos desde o início do século 19. Os conjuntos de formas definidos pela geometria convencional são incapazes de resolver tal problema. Por exemplo, como é possível representar formas irregulares, tais como árvores, nuvens, paisagens? Uma nuvem pode ser caracterizada como a forma geométrica de uma esfera? Ou uma montanha pode ser representada através de um cone? Uma das respostas para essas perguntas transcende a

geometria euclidiana e ficou conhecida como geometria fractal (PLOTZE; BRUNO, 2007).

Essa geometria, nada convencional, tem raízes remontando ao século 19 e algumas indicações neste sentido vêm de muito antes, na Grécia Homérica, Índia, China, entre outros. Porém, somente há poucos anos vem se consolidando com o desenvolvimento dos computadores e o auxílio de novas teorias nas áreas da física, biologia, astronomia, matemática e outras. Os fractais foram nomeados - ao invés de descobertos ou inventados - no início dos anos 80 por Benoit Mandelbrot, o "pai dos fractais" (SIQUEIRA, 2006).

Durante a década de 1970, Benoit Mandelbrot, pesquisador da International Business Machine (IBM), um matemático não ortodoxo, preferia aplicar geometria na solução dos problemas, ao invés de álgebra. Segundo suas teorias geométricas, nem um pouco clássicas, a resposta gráfica de um sistema consistia em uma noção mais completa do problema. E foi pelo fato de amar a geometria, que percebeu que sistemas aparentemente caóticos possuíam padrões simples de resposta que eram repetitivos e continham, intrinsecamente, um algoritmo de auto-repetição, capaz de gerar o todo. A essa forma geométrica chamou de fractal, que significa menor fração de um todo auto-similar capaz de gerar o conjunto. Seu nome veio do adjetivo latim *fractus*, que significa irregular ou fragmentado. A partir desse momento os fractais e caos são utilizados como ferramentas matemáticas para a modelagem visual do comportamento caótico de processos naturais e artificiais (BIRCHAL, 1994).

Esta recente área das ciências matemáticas vem tendo uma enorme aplicação. Para os biólogos, ajuda a compreender o crescimento das plantas. Para os físicos, possibilita o estudo de superfícies intrincadas. Para os médicos, dá uma nova visão da anatomia interna do corpo. Enfim, não faltam exemplos. Um dos mais belos - e, sem dúvida, o mais colorido - é o uso dos fractais na arte. Quando os computadores são alimentados com equações, eles criam magníficos desenhos abstratos (OLIVEIRA, 1994) exemplificado na Figura 4.



Fonte: Dupont; Rossetti e Siqueira (2008).

Figura 4 – Fractal “Origem do Universo”

A ciência dos fractais apresenta estruturas geométricas de grande complexidade e beleza infinita, ligadas às formas da natureza, ao desenvolvimento da vida e à própria compreensão do universo. São imagens de objetos abstratos que possuem o caráter de onipresença por terem as características do todo infinitamente multiplicadas dentro de cada parte (auto-similaridade), escapando assim, da compreensão em sua totalidade pela mente humana (DUPONT; ROSSETTI; SIQUEIRA, 2008).

Mandelbrot (1983, p. 15) definiu fractais da seguinte maneira: uma forma (curva ou superfície) para a qual a dimensão Hausdorff-Besicovitch excede estritamente a dimensão topológica, onde as formas como curvas, superfícies e sólidos apresentam dimensão topológica D_T 1, 2 e 3, respectivamente.

A dimensão Hausdorff-Besicovitch D , formulada por Hausdorff em 1919 e apresentada de forma final por Besicovich é definida a seguir.

No espaço Euclidiano R^E , ambas D_T e D são no mínimo 0 e no máximo E . Mas a semelhança termina aí. A dimensão D_T é sempre um valor inteiro, e D não necessariamente precisa ser um valor inteiro. E as duas dimensões não precisam coincidir, elas só devem satisfazer á desigualdade de Szpilrajn (MANDELBROT, 1983, p. 15), ou seja, $D \geq D_T$. No

caso das formas Euclidianas, $D = D_T$.

Considerando um conjunto de pontos no espaço Euclidiano de dimensão $E=3$. Por exemplo, o conjunto de pontos que compõem uma reta tem dimensão topológica $D_T = 1$ e dimensão Hausdorff-Besicovitch $D = 1$, como $D_T = D$ para a reta, então ela não é um fractal de acordo com a definição. Similarmente o conjunto de pontos que formam uma superfície tem $D_T = 2$ e $D = 2$ ou um conjunto de pontos que formam uma esfera $D_T = 3$ e $D = 3$ não são fractais (FEDER, 1989 apud PINTO, 2001, p. 24).

O fato de que D não precisar ser um inteiro merece uma terminologia à parte. Se for utilizada fração de um modo geral, como sinônimo de um número real não inteiro, a dimensão Hausdorff-Besicovitch pode ser chamada de dimensão fracionária. Mas D pode ser um inteiro (não maior que E mas estritamente maior que D_T , e então pode-se chamá-la de dimensão fractal.

2.3.1 Auto-similaridade

Uma importante característica das formas fractais é a sua auto-similaridade. Isso significa que uma determinada forma, tanto em escalas maiores (macroscópicas) quanto em escalas menores (microscópicas), é constituída por um mesmo conjunto de primitivas. Um exemplo clássico é a curva de Von Koch. Nele percebe-se que o conjunto total é constituído por pequenas réplicas desse mesmo conjunto e é basicamente neste princípio que se assenta o conceito de auto-similaridade, ou seja, qualquer que seja a ampliação considerada se obterá sucessivas cópias do objeto inicial. Convém distinguir dois tipos de auto-similaridade: a exata e a estatística (MESQUITA; MOTA, 1991). No caso da curva de Von Koch, a auto-similaridade é exata uma vez que parte da réplica obtida através das sucessivas iterações é idêntica a réplica em níveis de iterações anteriores.

Na Figura 5 é apresentado o conceito de auto-similaridade com a curva de Von Koch.

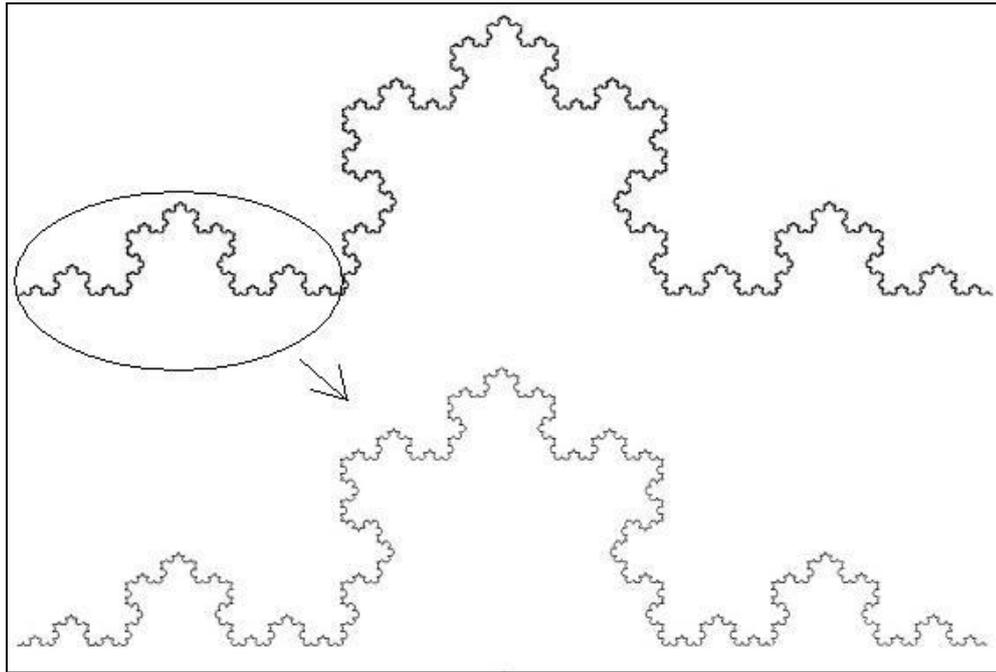


Figura 5 – Curva de Von Koch

Pode-se, no entanto, imaginar determinados tipos de figuras que ao serem ampliadas apresentem uma semelhança estatística, ou seja, uma semelhança que não sendo exata, é do mesmo tipo, apresentando os mesmos padrões e as mesmas características em termos gerais. O conjunto de Mandelbrot, mostrado na Figura 2, é um exemplo de auto-similaridade estatística, onde parte da figura ampliada apresenta formas parecidas à figura toda.

É importante salientar que as formas fractais são idealizações que podem não existir na natureza, e nem mesmo serem corretamente representadas no computador. Estes fatos se devem a dois motivos: a auto-similaridade infinita não pode ser representada na natureza e as limitações do processo de digitalização das imagens tendem a extinguir pequenos detalhes da imagem. Dessa maneira, é necessário assumir (para o caso de imagens digitais principalmente) que formas fractais possuem “fractalidade” limitada (PLOTZE; BRUNO, 2007).

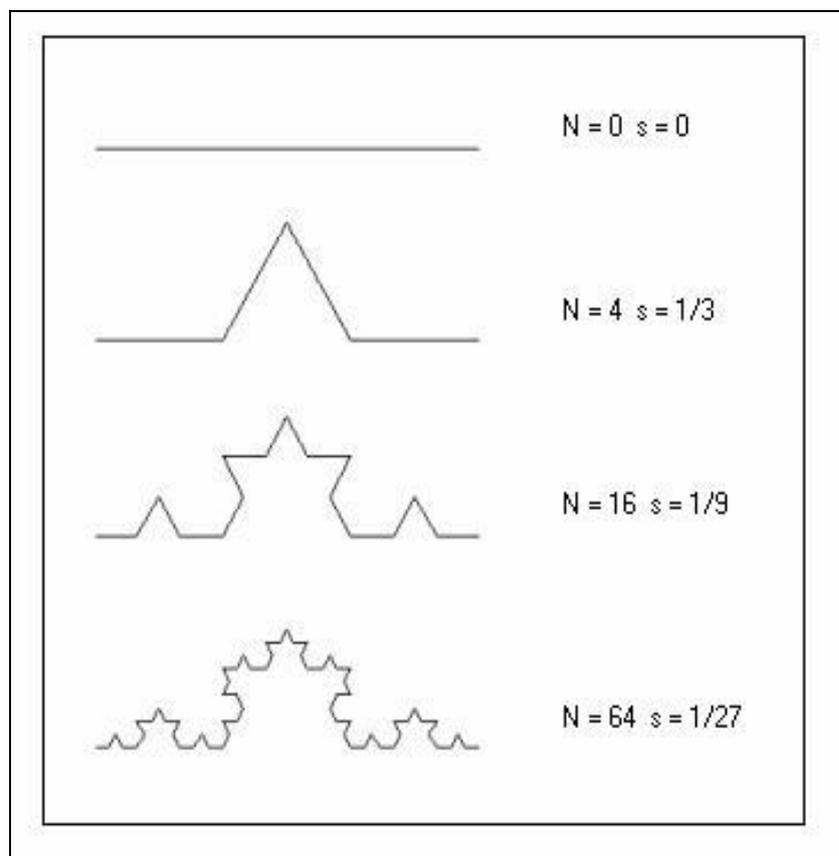
2.3.2 Dimensão fractal

A dimensão fractal pode ser determinada teoricamente se houver conhecimento do processo de formação do objeto. Um objeto geralmente considerado como 1D, um segmento de reta por exemplo, que possui a propriedade de auto-similaridade (Figura 6), pode ser dividido em N partes idênticas tais que sejam escalonadas pelo fator $s = 1/N$ de um todo.

Similarmente, para um objeto 2D, tal como a área de um quadrado, pode ser dividida em N áreas auto-similares tais que sejam escalonadas pelo fator $s = 1/N^{1/2}$. E um objeto 3D, como um cubo, pode ser dividido em N cubos menores tais que sejam escalonados pelo fator $s = 1/N^{1/3}$. Portanto um objeto auto-similar pode ser dividido em N cópias menores dele mesmo tal que seja escalonada pelo fator s (SCHROEDER, 1996 apud PINTO, 2001, p. 25), onde $s = \frac{1}{\sqrt[N]{N}} \therefore N(s) = \frac{1}{s^D} \therefore N(s) = s^{-D}$.

Assim, sua dimensão fractal é dada por $D = \frac{\log(N)}{\log(1/s)}$.

Por exemplo, qualquer segmento da curva de Koch é composta de 4 sub-segmentos tais que cada um é dividido pelo fator de 1/3 da sua parte original, portanto sua dimensão fractal é: $D = \frac{\log(4)}{\log(3)} \cong 1,26$. Pode-se ter uma idéia do processo de construção da curva de Koch observando a Figura 6.



Fonte: Pinto (2001, p. 27).

Figura 6 – Processo de construção da curva de Koch

Conforme Pinto (2001, p. 28) as imagens e formas encontradas na natureza não são

auto-similares, no entanto, existe outra classe de objetos fractais chamado auto-afins, que são formas fractais que apresentam diferentes valores de dimensão fractal para escalas de observação diferentes. Para esses tipos de fractais, não se conhecem leis que produzam tais modelos. Então, para obter a medida D é necessário construir métodos para a estimação da dimensão fractal. Um exemplo será apresentado no capítulo 4.4.

2.3.3 Estimação da dimensão fractal

O uso da estimação da dimensão fractal é de grande valia para várias áreas da ciência como já foi mencionado. Segundo Plotze e Bruno (2007) diversos pesquisadores demonstraram como os conceitos da geometria fractal, mais especificamente dimensão fractal, podem ser aplicados para extração de características e reconhecimento de padrões em imagens digitais. Aplicações podem ser encontradas em diversos campos, como: neurologia, classificação de íris, taxonomia vegetal, diagnóstico de diabetes, análise da diversidade de vegetações, dentre outros.

Existem vários métodos para a estimação da dimensão fractal, no entanto, devem-se saber as características do fractal a se analisar para o uso do método adequado. Por exemplo, segundo Russ (1994 apud PINTO, 2001, p. 31), o método de estimação *Box-Counting* é impróprio para os casos de formas auto-afins.

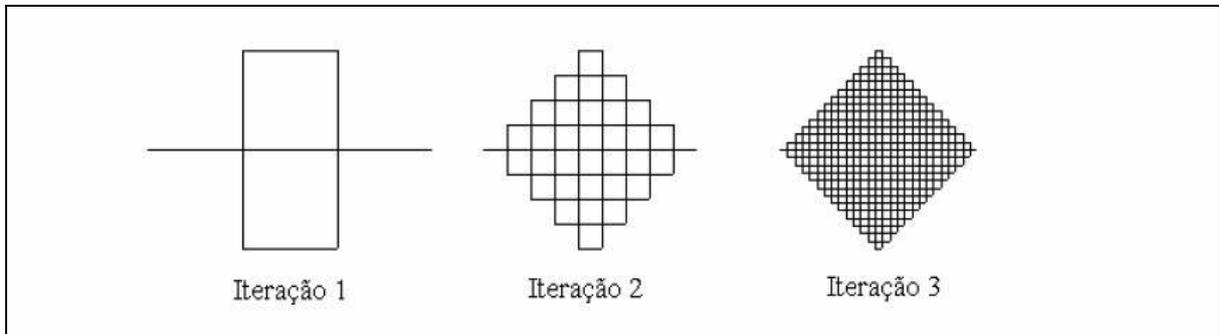
Podem-se relacionar alguns métodos de estimativa: *Box-Counting*; Rugosidade Média Quadrática; Espectro de Potência; Salsicha de Minkowski.

2.3.4 Iteração

O principal ingrediente na formulação matemática de exemplos vindos da ecologia foi a iteração (PEITGEN et al., 1988 apud SANTOS, 1999, p. 25). No jargão dos matemáticos, iteração significa repetição de um processo inúmeras vezes. É um processo muito simples, que pode ser explicado utilizando o exemplo de uma calculadora científica contendo as funções x^2 , \sqrt{x} , $\sin x$, $\exp x$ e assim por diante. Cada uma destas teclas representa uma função matemática. Ao entrar com um valor particular na calculadora e pressionar a tecla x várias vezes, está ocorrendo uma iteração da função raiz quadrada. Em cada iteração da raiz

quadrada utiliza-se o valor da iteração imediatamente anterior.

Pode-se ter uma idéia melhor de iteração observando a Figura 7, a curva de Hilbert uma variação da curva de Koch. Nessa construção cada segmento de reta é substituído por outro segmento de reta que representa $1/3$ do segmento anterior.



Fonte: Santos (1999, p. 26).

Figura 7 – 3 iterações da construção da curva de Hilbert

Na geração de fractais utilizando o processo iterativo observa-se que a partir de regras simples geram-se formas complexas. Ao contrário das formas Euclidianas, essas formas fractais possuem detalhes em todas as escalas, independentemente do fator de ampliação (PEITGEN et al., 1988 apud SANTOS, 1999, p. 27).

A dificuldade na geração dos fractais está no estabelecimento de regras para produzir um objeto com determinadas características, conforme veremos a diante através da estratégia para geração do fractal deste trabalho.

Convém distinguir iteração de interação. Iteração como foi mencionado significa repetição, já interação significa ação recíproca de dois ou mais corpos ou ação mútua entre duas partículas ou dois corpos.

2.4 JCCKIT

Java Chart Construction Kit (JCCKit) é uma biblioteca Java para construção de gráficos e diagramas científicos. Escrita para a plataforma JDK 1.1.8 com exceção para a classe Graphics2D do Java. É recomendada para uso em aplicações *Applets* e em aplicativos para *Personal Digital Assistants* (PDAs) (ELMER, 2005), podendo também ser usada em qualquer aplicativo Java.

Totalmente orientado a objetos e altamente extensível. Foi totalmente escrita em Java 1.1. Sua arquitetura é baseada em interfaces e classes abstratas o que facilita sua extensão.

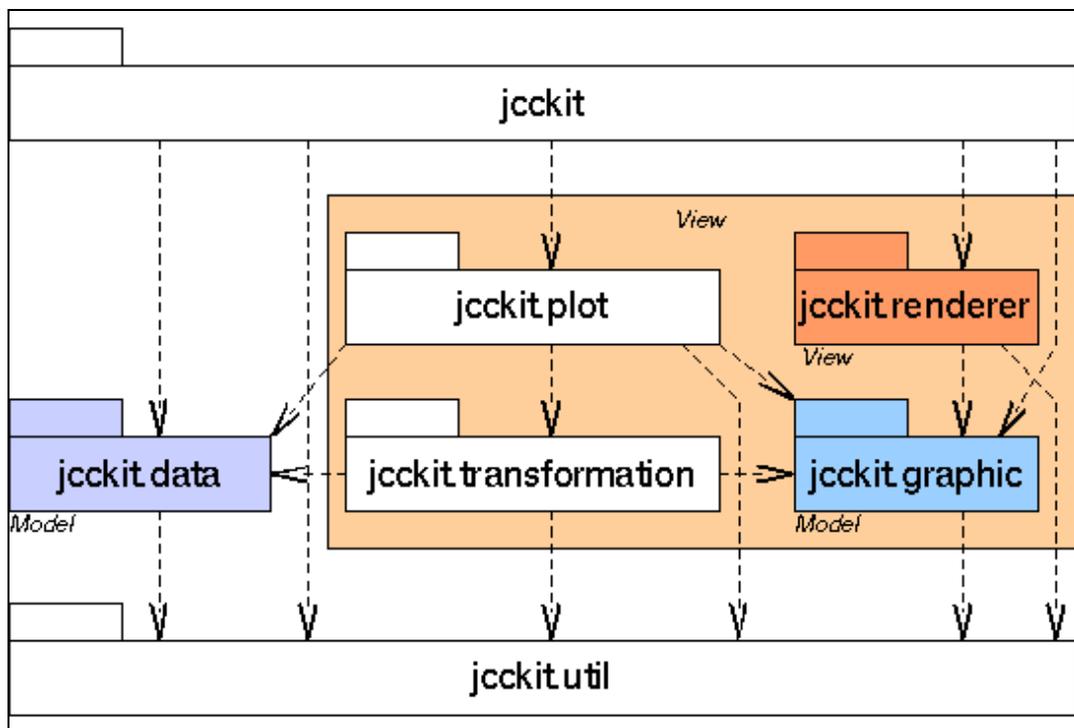
2.4.1 Visão geral

As classes do JCKKit são agrupadas em sete pacotes constituindo quatro camadas. O Quadro 1 mostra os sete pacotes do JCKKit com uma breve descrição de suas classes.

PACOTE	DESCRIÇÃO
Jcckit	Classes para uso imediato.
jcckit.plot	Classes para a geração de gráficos e diagramas baseado em dados modelados por classes do pacote jcckit.data.
jcckit.renderer	Classes que renderizam um gráfico (composto por objetos gráficos de jcckit.graphic) para um determinado dispositivo de saída.
jcckit.data	Classes para os dados de coordenadas.
jcckit.transformation	Classes que transformam dados de coordenadas em dispositivos de coordenadas.
jcckit.graphic	Classes que representam um gráfico ou diagrama no dispositivo de objetos gráficos.
jcckit.util	Utilitário de classes. Em sua maioria para configurações de renderização.

Quadro 1 – Pacotes do JCKKit

Na Figura 8 é apresentado um diagrama dos pacotes JCKKit mostrando suas dependências e suas divisões em camadas.



Fonte: Elmer (2005).

Figura 8 – Diagrama dos pacotes JCKKit

Os cinco pacotes centrais podem ser agrupados dentro de um modelo que represente o pacote de gráficos e diagramas, sendo que quatro destes pacotes são para visualização. Dentro

deste modelo de visualização existem dois pacotes abstratos, sendo um para modelo (`jcckit.graphic`) e outro para exibição (`jcckit.renderer`).

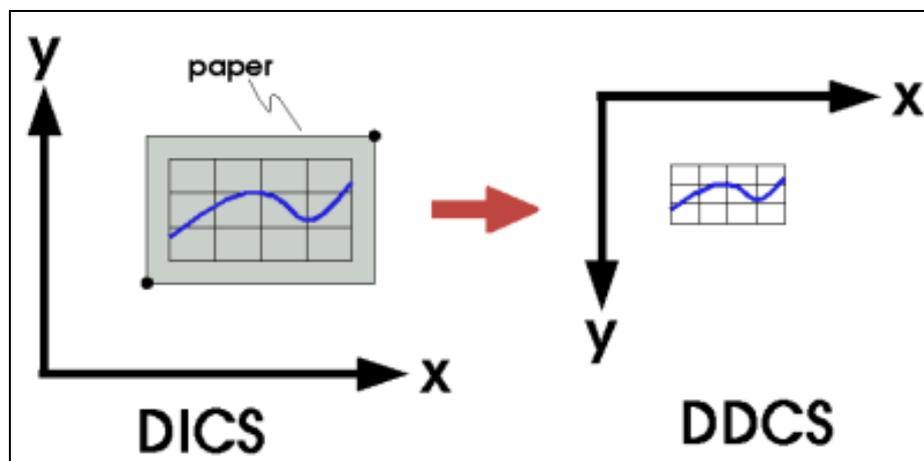
2.4.2 Sistema de coordenadas

Existem três sistemas de coordenadas no JCCKit, descritos na Quadro 2.

SISTEMA DE COORDENADA	DESCRIÇÃO
<i>Data Coordinate System (DCS)</i>	Este é o sistema de coordenadas dos dados a serem traçados.
<i>Device-Independent Coordinate System (DICS)</i>	Um sistema de coordenadas cartesianas independentes do dispositivo.
<i>Device-Dependent Coordinate System (DDCS)</i>	Sistema de coordenadas do dispositivo que apresenta a renderização de diagramas e gráficos.

Quadro 2 – Sistemas de coordenadas do JCCKit

Pontos mapeados no sistema de coordenadas DCS (representado por instâncias da classe `DataPoint`) podem ser mapeados das seguintes formas: DCS para DICS e DICS para DDCS, conforme necessidade da implementação. A Figura 9 mostra uma comparação entre os sistemas de coordenadas DICS e DDCS.



Fonte: Elmer (2005).

Figura 9 – Representação do mapeamento DICS para DDCS

3 FERRAMENTA DE VISUALIZAÇÃO 3D DE UM SISTEMA DE ARQUIVOS

Este capítulo apresenta os principais conceitos envolvidos no trabalho desenvolvido por Krauspenhar (2007) sob o título de “Ferramenta de Visualização 3D de um Sistema de Arquivos”, o qual é a base e inspiração para o desenvolvimento do presente trabalho.

3.1 INTRODUÇÃO

Esta ferramenta consiste num ambiente de visualização tridimensional de um sistema de arquivos no Windows, utilizando conceitos da metáfora de memória espacial. A ferramenta apresenta uma interface onde é possível criar diversos objetos e associá-los a arquivos e/ou diretórios, onde uma alteração nestes componentes do sistema de arquivos influi em uma representação nos objetos associados.

Através do conceito de memória espacial para a gerência de objetos do sistema de arquivos é possível uma interação com os objetos do sistema de arquivos, com o intuito de tornar mais fácil esta gerência.

Esta ferramenta está limitada a um universo estático (Figura 10). Onde o sistema de arquivos referente ao ambiente representativo não aceita o total de diretórios reais do sistema operacional.



Fonte: Krauspenhar (2007, p. 45).

Figura 10 – Tela de execução do sistema de Krauspenhar

3.2 SISTEMA DE ARQUIVOS

De acordo com Tanenbaum e Woodhull (2000, p. 27), o sistema de arquivos é a forma que o sistema operacional usa para representar a informação em um espaço de armazenamento, apresentando um modelo abstrato e independente de dispositivos. Neste modelo são implementadas chamadas de sistema para as operações de criar, remover, ler e escrever arquivos.

O sistema de arquivos é uma parte fundamental do sistema operacional, pois fornece uma visão abstrata dos dados persistidos, além de ser responsável pelo serviço de nomes, acesso a arquivos e de sua organização geral.

A maioria dos sistemas de arquivos possui o conceito de diretório, que consiste em uma forma de organizar e agrupar arquivos. As mesmas chamadas de sistema utilizadas para operações com arquivos também são aplicadas aos diretórios. Neste modelo, a estrutura hierárquica é formada e pode ser constituída por inúmeros níveis, a partir de um diretório raiz.

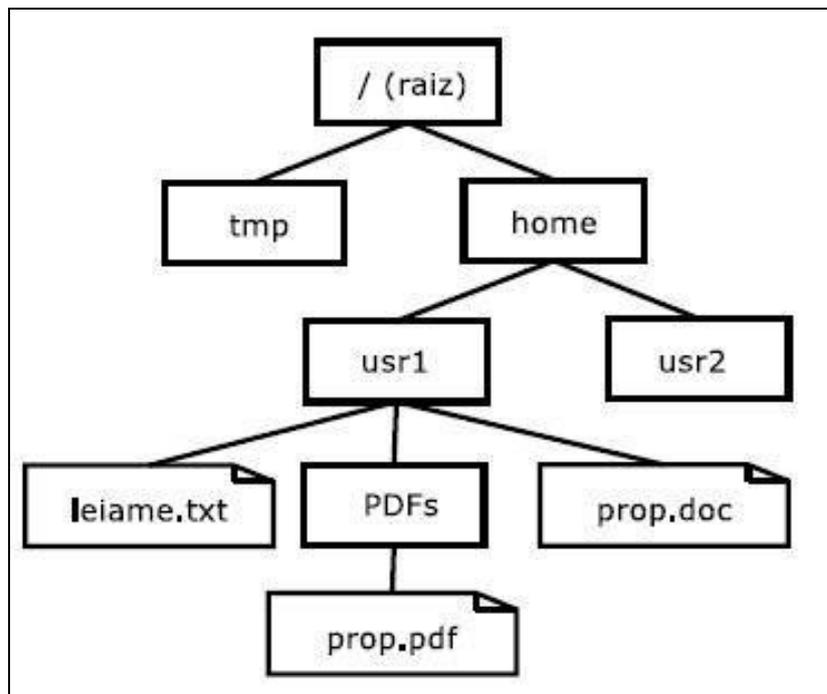
Carvalho (2005, apud KRAUSPENHAR 2007, p. 17) explica que a maioria dos arquivos armazenados em um sistema de arquivos possui um nome e um caminho, utilizados para a identificação única em tal sistema. Um caminho representa um nó de uma estrutura de diretórios. A localização de um arquivo nesta estrutura parte da raiz, percorrendo os nós

correspondentes aos diretórios até encontrar a informação desejada.

3.2.1 Estrutura hierárquica

Uma estrutura hierárquica tem como princípio organizar informações em ordem de importância ou disparidade. Normalmente consiste em uma estrutura de árvore onde cada nó da estrutura pode ter apenas um correspondente superior. Na gerência de arquivos, a estrutura hierárquica é comumente utilizada, apresentando diretórios e arquivos como elementos da estrutura (TANENBAUM; WOODHULL, 2000, p. 27).

A Figura 11 mostra uma representação de uma estrutura hierárquica de um sistema de arquivos, apresentando a árvore de diretórios, subdiretórios e arquivos.



Fonte: Carvalho (2005, apud KRAUSPENHAR, 2007, p. 18).

Figura 11 – Exemplo de uma árvore de diretórios

Para geração de uma estrutura organizacional de diretórios no sistema operacional Windows da Microsoft utiliza-se a ferramenta de gerência Windows Explorer, sendo que esta não é a única forma. Nele é possível criar diretórios, subdiretórios e incluir arquivos dentro desses diretórios, dentre outras funcionalidades que escapam do escopo deste trabalho.

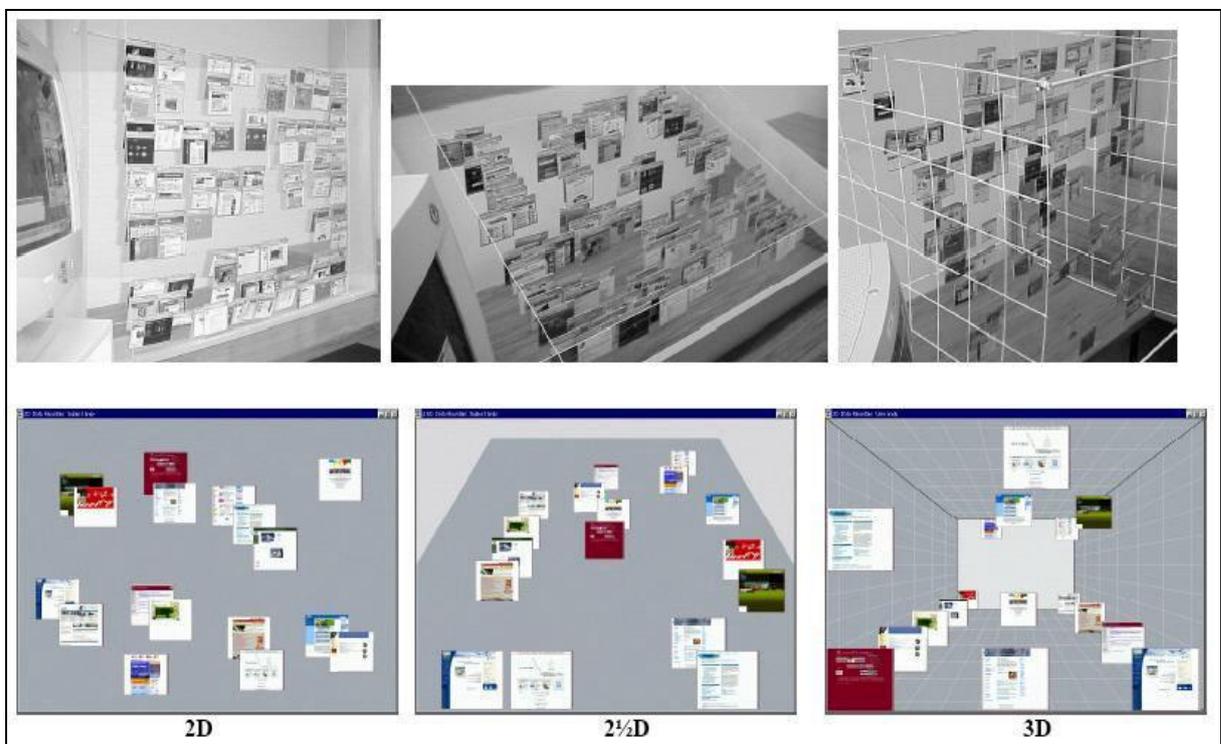
Apesar da organização de arquivos ser uma tarefa bastante pessoal, Henderson (2004 apud KRAUSPENHAR, 2007, p. 18) identificou que é comum usuários utilizarem um padrão de organização, normalmente composto pela seguinte hierarquia de conceitos: gênero, tarefa,

assunto e tempo.

3.3 MEMÓRIA ESPACIAL EM AMBIENTES VIRTUAIS

Cockburn e McKenzie (2001 apud KRAUSPENHAR, 2007, p. 19) procuraram identificar a efetividade da memória espacial em ambientes de modelos físicos reais e em modelos virtuais correspondentes, partindo dos resultados apresentados pelo projeto Data Mountain.

Utilizando seis ambientes para o estudo, três ambientes reais físicos provendo uma experiência para os usuários sem as limitações de visualização virtual e três ambientes virtuais equivalentes nas perspectivas 2D, $2\frac{1}{2}$ D e 3D, vários usuários foram submetidos às tarefas de organizar páginas e posteriormente acessar as mesmas na organização criada. O ambiente $2\frac{1}{2}$ D corresponde à visualização apenas do plano inferior em perspectiva. A Figura 12 apresenta os ambientes utilizados.



Fonte: adaptado de Cockburn e McKenzie (2001 apud KRAUSPENHAR, 2007, p. 19).

Figura 12 – Ambientes virtuais utilizados para o estudo de memória espacial

A maioria dos usuários apresentou dificuldade na tarefa de organizar os objetos no

ambiente $2\frac{1}{2}$ D e 3D, porém, ficou comprovada a capacidade espacial do cérebro humano de acessar rapidamente os documentos nestes ambientes. Nos testes realizados, o acesso a um item organizado anteriormente no ambiente 2D físico foi de 4,2 segundos, enquanto no ambiente 3D foi de 3,7 segundos (COCKBURN; MCKENZIE, 2001 apud KRAUSPENHAR, 2007, p. 19).

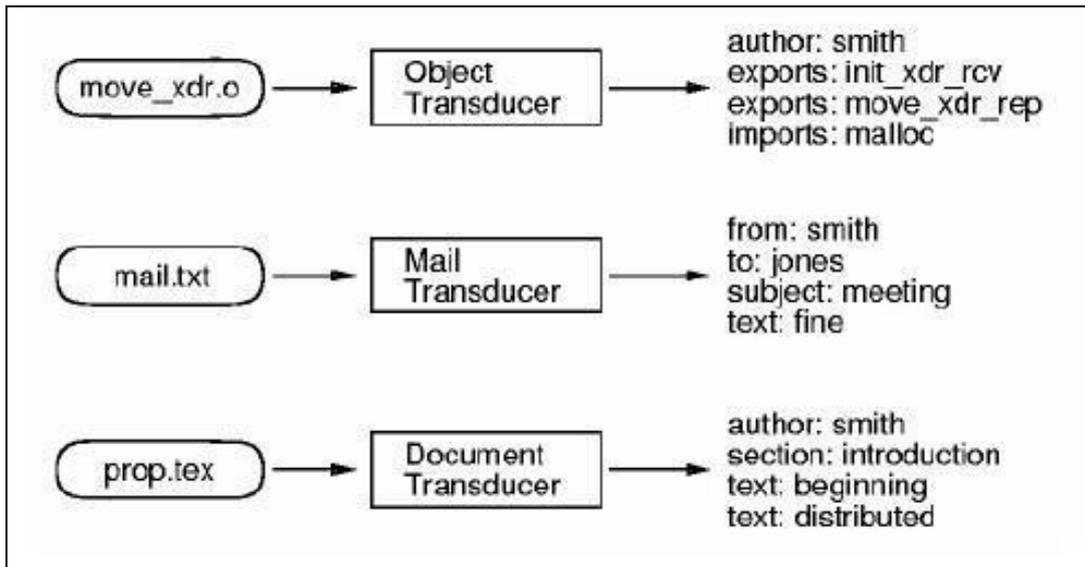
A utilização de um ambiente tridimensional também apresenta algumas facilidades, como a possibilidade de movimento e rotação, permitindo alterar o ponto de vista das informações (COCKBURN; MCKENZIE, 2001 apud KRAUSPENHAR, 2007, p. 20).

3.4 ARQUIVOS SEMÂNTICOS

De acordo com Gifford et al. (1988 apud KRAUSPENHAR, 2007, p. 21), um arquivo semântico é uma informação armazenada que provê acesso flexível e associativo aos dados do sistema pela extração automática de atributos dos arquivos através de tradutores específicos por formato. Pesquisas são efetuadas através destes atributos, definindo grupos de dados semelhantes.

A compatibilidade com os atuais sistemas de arquivos é possível através do conceito de diretório virtual, onde o resultado de cada pesquisa será armazenado. Através dos atributos dos arquivos, é possível determinar agregações de informações, associações entre arquivos e associações entre grupos de informações.

A utilização de arquivos semânticos possibilita a definição de comportamento dos usuários, definindo grupos de dados mais acessados de acordo com o tempo (GIFFORD et al., 1988 apud KRAUSPENHAR, 2007, p. 21). A Figura 13 apresenta um exemplo de tradução.



Fonte: Gifford et al. (1988 apud KRAUSPENHAR, 2007, p. 22).

Figura 13 – Exemplo de tradução semântica

3.5 TRABALHOS CORRELATOS

Ferramentas comerciais e acadêmicas foram desenvolvidas tendo em vista resolver o problema de gerenciamento de arquivos. Dentre elas, foram escolhidas o Task Gallery, BumpTop, Tactile 3D e XCruiser que apresentam uma proposta semelhante.

Em 1999, a Microsoft desenvolveu uma interface para expandir a janela do desktop em um ambiente 3D com ilimitadas janelas de desktop, chamada Task Gallery (ROBERTSON et al., 1999, p. 1) para o sistema operacional Windows. O conceito é tentar criar uma ilusão baseada na habilidade humana de visão espacial, aonde as pessoas vão intuitivamente navegar no ambiente. Os estudos utilizando o Task Gallery provaram que os usuários utilizaram o sistema de forma bastante natural, tendo comparado o ambiente ao mundo real (ROBERTSON et al., 1999, p. 5).

A interface apresentada pelo Task Gallery utiliza a técnica Perspective Wall apresentada por Mackinlay et al. (1991 apud KRAUSPENHAR, 2007, p. 26), porém não existe uma definição de que a informação principal está associada à região central da imagem 3D e as áreas laterais contém informações de contexto geral. A Figura 14 mostra a tela de execução do Task Gallery.



Fonte: Elmer (2005).

Figura 14 – Tela de execução do Task Gallery

BumpTop (AGARAWALA; BALAKRISHNAN, 2006 apud KRAUSPENHAR, 2007, p. 26) é um protótipo que está sendo desenvolvido pela Universidade de Toronto. Consiste em um ambiente de gerência de arquivos em 3D para o sistema operacional Windows utilizando conceitos de física na manipulação de documentos, como atrito e massa. Utiliza também um conceito de pilhas para organizar documentos. A visualização corresponde a uma mesa tridimensional onde é possível criar pilhas e distribuir documentos aleatoriamente. A proximidade dos itens com a tela corresponde à utilização freqüente dos mesmos. Pilhas de documentos normalmente estão ordenadas cronologicamente, pois os itens mais atuais serão inseridos no topo da pilha.

Este projeto apresenta uma alternativa para a gerência de documentos, ignorando completamente a atual estrutura hierárquica para apresentar um modelo mais próximo ao mundo real. A Figura 15 representa a execução do aplicativo BumpTop.



Fonte: Agarawala e Balakrishnan (2006 apud KRAUSPENHAR, 2007, p. 27).

Figura 15 – Tela de execução do BumpTop

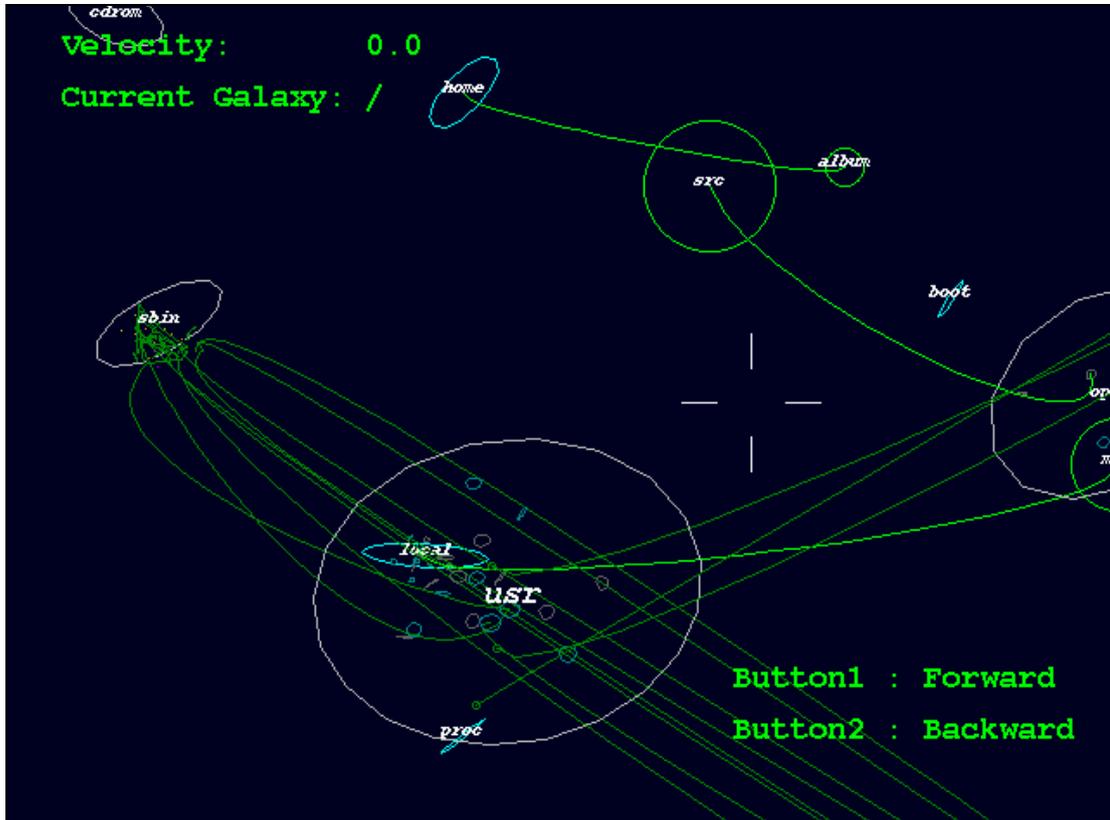
Entre as ferramentas desenvolvidas para a visualização de sistemas de arquivos utilizando técnicas semelhantes, pode-se citar a Tactile 3D (UPPER BOUNDS INTERACTIVE, 2006 apud KRAUSPENHAR, 2007, p. 27) onde é apresentado um ambiente semelhante a um jogo tridimensional. A Figura 16 demonstra a visualização do sistema de arquivos na ferramenta Tactile 3D.



Fonte: Upper Bounds Interactive (2006 apud KRAUSPENHAR, 2007, p. 28).

Figura 16 – Visualização de diretórios no ambiente Tactile 3D

A ferramenta XCruiser (SHINYAMA, 2003) aplica a metáfora de galáxias para o sistema de arquivos, onde em um ambiente 3D, arquivos são representados como planetas e diretórios por galáxias, como representa a Figura 17.



Fonte: Shinyama (2003).

Figura 17 – Visualização de uma partição Unix no ambiente XCruiser

4 DESENVOLVIMENTO

Para detalhar o processo de desenvolvimento serão abordados os temas a seguir:

- a) análise e especificação dos requisitos;
- b) especificação através de diagramas de classe e casos de uso;
- c) estratégia para geração do fractal e principal técnica.

4.1 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO

A ferramenta deverá conter os seguintes requisitos funcionais:

- a) mapear os objetos do sistema de arquivos para uma estrutura de dados baseada no modelo de fractais: a ferramenta deverá gerar um fractal onde cada ponto nó do fractal representa um arquivo ou diretório do sistema de arquivos;
- b) permitir a ampliação dos ramos do fractal: o protótipo deverá ampliar os ramos mostrando a complexidade dos fragmentos seleccionados, sendo possível ver cada nível do fractal.
- c) permitir a visualização do conteúdo de cada ponto do fractal: mostrar o conteúdo de cada ponto nó do fractal os quais representam os objetos do sistema de arquivos do sistema operacional.

A ferramenta deverá conter os seguintes requisitos não funcionais:

- a) ser implementado utilizando o ambiente de desenvolvimento Eclipse SDK versão 3.3.2;
- b) ser implementado na linguagem Java versão 1.6;
- c) ser compatível com os sistemas operacionais Windows XP e Vista.

4.2 ESPECIFICAÇÃO

Nesta seção serão apresentados os diagramas de casos de uso e diagramas de classe. Os diagramas foram desenvolvidos utilizando a ferramenta Enterprise Architect versão 7.1 da

empresa Sparx Systems.

4.2.1 Diagrama de casos de uso

Os diagramas de casos de uso apresentados nesta seção tem como ator o usuário que fará a visualização dos objetos através do fractal representativo e o próprio ambiente ao receber as mensagens do sistema operacional.

Na Figura 18 é descrito o usuário como ator que utilizará a interface e efetuará a geração do fractal selecionando o diretório e sua manipulação.

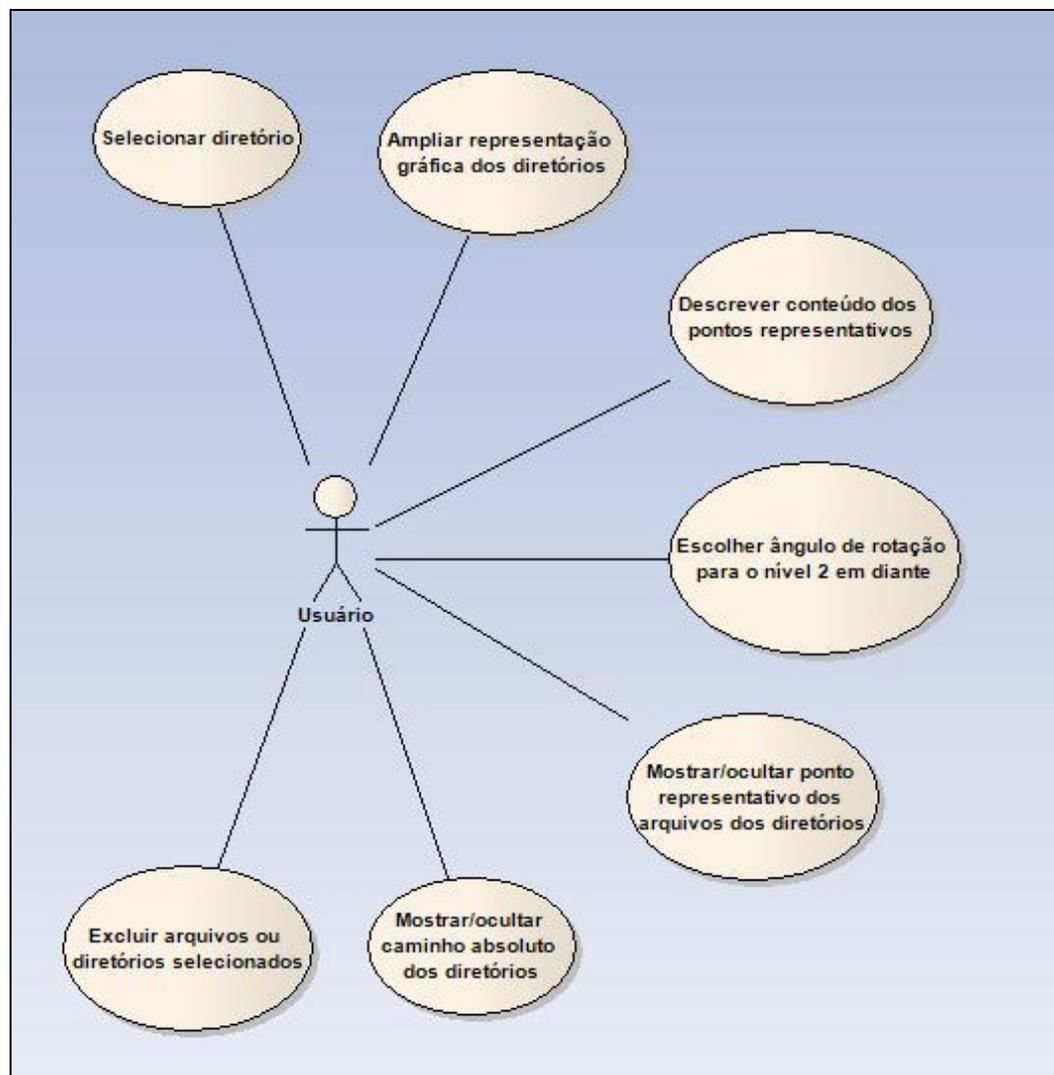


Figura 18 – Diagrama de classes de uso do ator usuário

Na Figura 19 o ator é o ambiente que receberá a mensagem do sistema operacional no momento em que o diretório for selecionado.

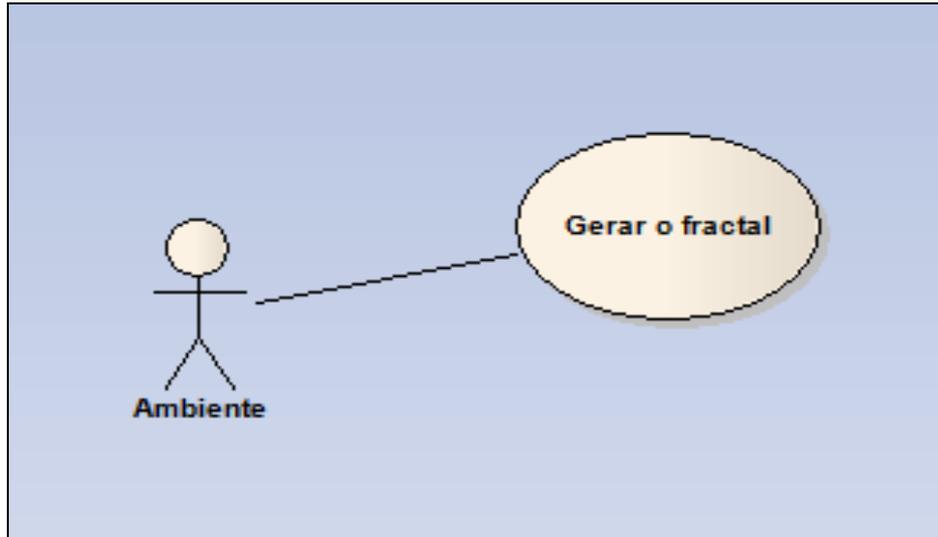


Figura 19 – Diagrama de classes de uso do ator ambiente

4.2.2 Diagrama de classes

Os diagramas de classes apresentados nesta seção demonstram a estrutura de dados para a construção da ferramenta.

Na Figura 20 é apresentado o diagrama de classes, com seus atributos e métodos, das classes utilizadas para gerar a estrutura de dados para criação do fractal.

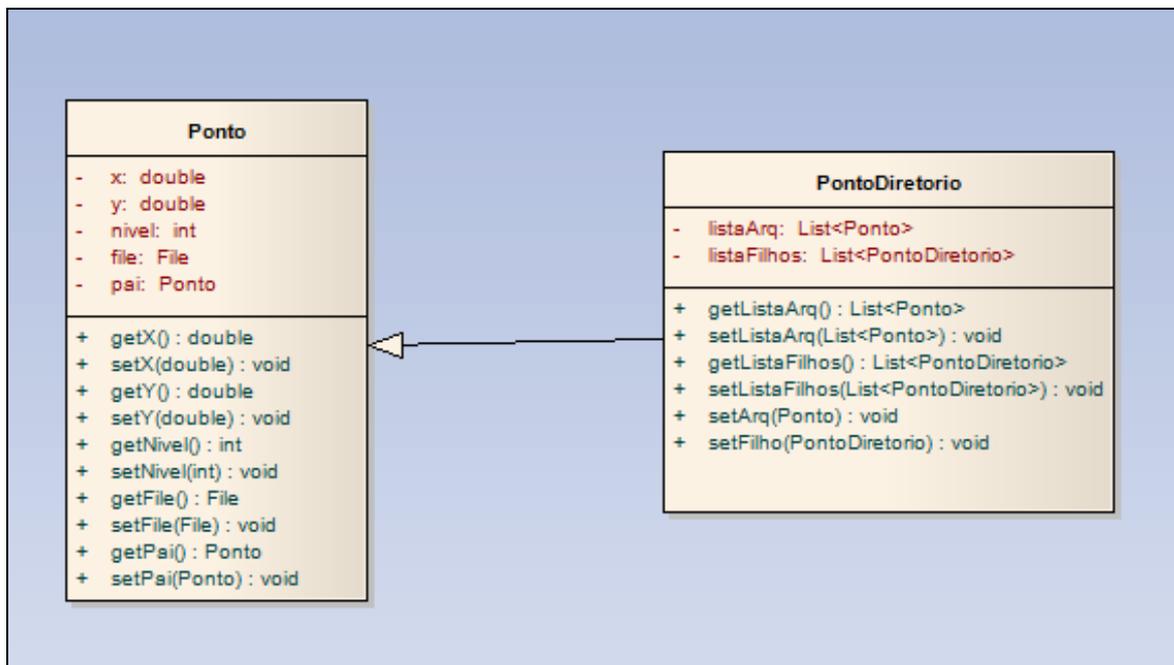


Figura 20 – Diagrama das classes responsável pela estrutura de dados

Na Figura 21 será apresentado o diagrama de classes da classe geradora do fractal, se resumindo aos principais atributos e métodos responsáveis pela geração do fractal.



Figura 21 – Diagrama das classes responsável pela geração e manipulação do fractal

A seguir cada uma das classes é apresentada em detalhes.

4.2.2.1 Classe Ponto

A classe `Ponto` é utilizada para representar os pontos base para os nodos do fractal. Nele incluem-se os arquivos que estão nos diretórios.

Possui os seguintes atributos:

- a) `x` e `y`: responsáveis pelas coordenadas dos pontos no sistema de coordenadas;
- b) `nivel`: guarda o nível do ponto;
- c) `file`: guarda o dado relativo ao objeto do sistema de arquivos;
- d) `pai`: para caso necessário buscar alguma informação do ponto pai.

Os métodos existentes nesta classe servem para acessar seus atributos, fazendo desta forma o chamado encapsulamento proveniente do conceito de programação orientada a objetos.

4.2.2.2 Classe PontoDiretorio

A classe `PontoDiretorio` é uma extensão da classe `Ponto`. Ela tem por objetivo tornar possível o encadeamento dos diretórios.

Possui os seguintes atributos:

- a) `listaArq`: responsável por armazenar os arquivos representados pela classe

Ponto.

- b) `listaFilhos`: responsável por encadear os diretórios representados pela classe `PontoDiretorio`.

Os métodos existentes na classe `PontoDiretorio` objetivam encapsular seus atributos, provendo seus devidos acessos.

4.2.2.3 Classe `FractalArvoreDir`

A classe `FractalArvoreDir` é responsável pela geração da estrutura de dados dos diretórios selecionados podendo também gerar a estrutura dos pontos com suas respectivas posições. Responsável também pela geração do gráfico do fractal e pelo texto que apresenta o conteúdo dos pontos análogos aos objetos do sistema de arquivos. Serão apresentados apenas os atributos e métodos principais no desenvolvimento do fractal por esta classe.

Possui os seguintes atributos:

- a) `raiz`: é através da raiz que é encadeado toda a estrutura de diretórios, gerando uma árvore n-ária de pontos diretórios (classe `PontoDiretorio`), conforme é a própria estrutura do sistema de arquivos;
- b) `originalDir`: este atributo é responsável por guardar o diretório selecionado pelo usuário, a partir dele será gerado o fractal;
- c) `possuiArq`: esta propriedade diz se possui arquivos ou não na geração da estrutura de dados, isto é, inclui os arquivos das pastas na geração da estrutura de dados;
- d) `mostraCamAbsDir`: responsável por mostrar ou não o caminho absoluto dos diretórios na geração do texto descritivo dos diretórios;
- e) `valorRotacao`: este valor será atribuído ao ângulo de rotação para dividir entre os pontos diretórios a partir do nível dois em diante. Cabe salientar que este valor de rotação será dividido entre o número de diretórios existentes no nível em questão, sendo possível informar ele na tela. O valor de 120° se mostrou eficaz para não colidirem as retas, tendo um bom aproveitamento da área.

Possui os seguintes métodos:

- a) `gerarEstruturaDados`: este método é responsável por gerar recursivamente a estrutura de dados do sistema de arquivos, representado por uma árvore n-ária;

- b) `desenharFractal`: através deste método o fractal é gerado recursivamente, nele é feita toda a estratégia de geração do fractal;
- c) `desenharFractal`: neste método com o mesmo nome do método anterior é aplicado o que chama-se em programação orientada a objetos de sobrecarga de métodos. O método tem por finalidade atribuir o posicionamento dos pontos dentro da estrutura de dados existente e ,assim como no método anterior, nele é feita toda a estratégia de geração do fractal;
- d) `getArvoreSelecionada`: gera texto dos arquivos e diretórios selecionados. Dentro do protótipo caso a opção “Mostrar arquivos dos diretórios” estiver selecionada, será apresentado os arquivos com as propriedades: data de modificação, tipo de dado e tamanho em bytes. Caso esta opção não esteja selecionada não será apresentado os pontos referentes aos arquivos no gráfico;
- e) `deleteDirSel`: responsável por excluir arquivos ou diretórios selecionados. Exclui também subdiretórios e arquivos subseqüentes quando existentes dentro do diretório selecionado para a exclusão.

4.3 IMPLEMENTAÇÃO

A seguir são mostradas as técnicas e a operacionalidade da implementação.

4.3.1 Estratégia para geração do fractal

A estratégia para geração do desenho fractal é baseada nas seguintes premissas.

- a) representar uma árvore n-ária com sua hierarquia, assim como funciona no sistema de arquivos;
- b) não colidir as retas que ligam um diretório ao outro;
- c) aproveitar o espaço no gráfico sem desperdício de área.

Para demonstrar a estratégia é apresentada a imagem do fractal gerada pelo protótipo (Figura 22).

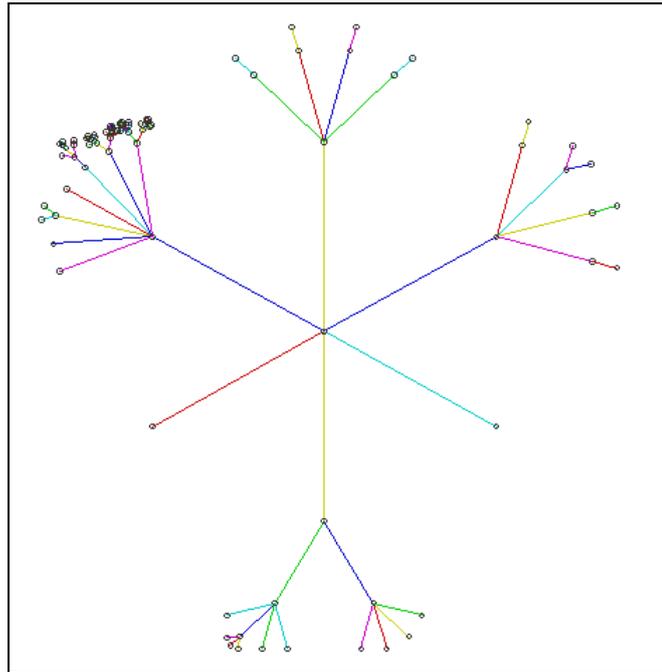


Figura 22 – Fractal representativo do diretório C:\Users\Falcone\Documents\Visual Studio 2005

Verifica-se que para o primeiro nível de iteração do fractal o ângulo total de rotação para os pontos é de 360° . Neste caso verificam-se 6 diretórios no primeiro nível de iteração. Dividindo-se 360° por 6 pontos se obtém o ângulo de 60° para cada fatia.

A partir do nível dois em diante a estratégia de geração é modificada, altera-se o ângulo total de rotação para 120° e o tamanho da reta é fragmentado. Conforme Figura 23.

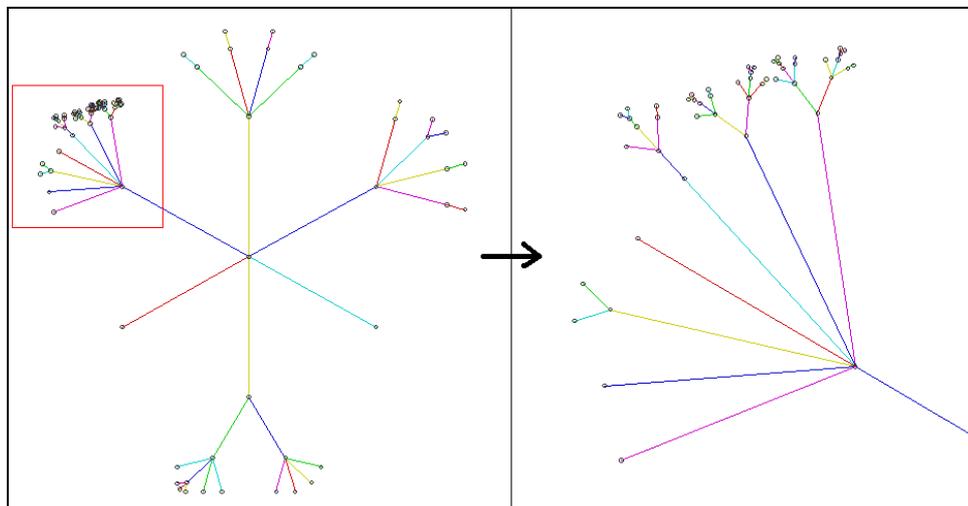


Figura 23 – Representação da fragmentação ocorrida de um nível para outro posterior

Para o novo tamanho de reta é feito um cálculo verificando o ângulo de rotação no nível anterior para medir distância entre os pontos. Os pontos a ser medida a distância são demonstrados na Figura 24.

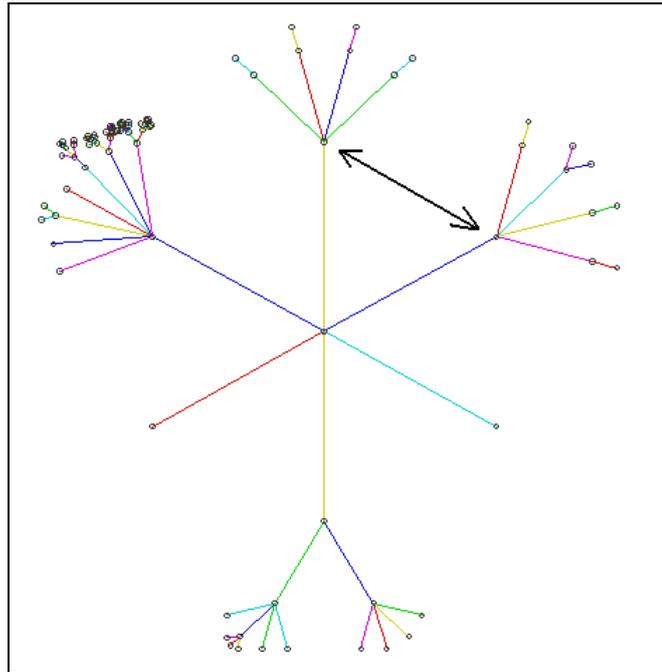


Figura 24 – Pontos do fractal usados para cálculo do tamanho das retas do próximo nível

O tamanho desta nova reta (nível 2) tem a metade do tamanho da distância demonstrada. Verifica-se que aplicando o ângulo total de rotação de 120° eficiência para a não intersecção entre as retas.

Outro ponto a ser levado em consideração na geração do fractal é a direção dos níveis subsequentes, que levam em consideração a direção do nível do ponto pai. Este que está sempre no meio do ramo.

4.3.2 Recursão

A recursividade na programação de computadores envolve a definição de uma função ou método que pode invocar a si próprio. A recursão foi utilizada nos cinco principais métodos deste trabalho, são eles:

- a) gerarEstruturaDados;
- b) desenharFractal;
- c) desenharFractal¹;
- d) getArvoreSelecionada;
- e) deleteDirSel.

¹A linguagem de programação Java permite a criação de métodos com o mesmo nome, porém com parâmetros e tipos de retorno diferentes. Este conceito é conhecido dentro da orientação a objetos por sobrecarga.

Dentre eles será apresentado o método `gerarEstruturaDados` e explicado seu funcionamento para entendimento do conceito de recursividade na programação de computadores. A Figura 25 descreve o código fonte deste método.

```

public void gerarEstruturaDados(File fileObject, PontoDiretorio ponto) {
    try {
        if (fileObject.isDirectory()) {
            int nivel = getNivel(fileObject);
            if (raiz == null) {
                ponto = new PontoDiretorio(fileObject, nivel);
                raiz = ponto;
            } else {
                PontoDiretorio filho = new PontoDiretorio(fileObject, nivel);
                ponto.setFilho(filho);
                ponto = filho;
            }

            File allFiles[] = fileObject.listFiles();
            for(File aFile : allFiles){
                gerarEstruturaDados(aFile, ponto);
            }
        } else if (fileObject.isFile() && mostraArquivos){
            int nivel = getNivel(fileObject);
            Ponto pontoArq = new Ponto(fileObject, nivel);
            ponto.setArq(pontoArq);
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}

```

Figura 25 – Código fonte do método `gerarEstruturaDados`

Este método tem o objetivo de encadear uma árvore n-ária representativa de um diretório passado como o parâmetro. Utiliza-se a variável global `raiz` para encadeamento da árvore.

Existem dois casos básicos para este método:

- a) objeto é um diretório;
- b) objeto é um arquivo.

Para o primeiro caso ele tem a possibilidade de ter mais diretórios subsequentes, já no segundo caso quando for um arquivo ele termina a busca nesse ponto.

Segundo Deitel e Deitel (2005, p. 553) o método recursivo é dividido em duas partes conceituais, uma que o ele sabe como resolver e outra que ele não sabe como resolver. No caso a que ele não sabe resolver ele divide o problema por uma versão ligeiramente mais simples ou menor. Como esse novo problema é parecido com o problema original, o método chama uma nova cópia dele próprio para trabalhar no problema menor – isso é referido como chamada recursiva ou como passo de recursão.

No caso apresentado o caso básico ocorreria quando um diretório alcançado não contivesse mais nenhum diretório ou arquivo ou quando chegasse a um arquivo, encerrando assim a busca por subdiretórios nesse ponto.

4.3.3 Operacionalidade da implementação

A ferramenta apresenta as seguintes funcionalidades verificadas na Figura 19, descritas como segue:

- a) selecionar diretório: é possível selecionar um diretório de duas formas. Uma é digitando diretamente o diretório no campo “Diretório” e depois clicando no botão “Atualizar”, outra é selecionando o diretório através do botão “Abrir Diretório”. Por padrão este campo vem com o diretório corrente. Através deste procedimento o fractal é gerado para representar este diretório;
- b) ampliar representação gráfica dos diretórios: através do uso do clique com o botão esquerdo mouse é possível selecionar parte da imagem gerada pelo protótipo. Após selecionar um pedaço da imagem, a nova imagem que ocupará a área reservada ao fractal será da região selecionada;
- c) descrever o conteúdo dos pontos representativos: como cada ponto representa um diretório ou arquivo convém mostrar o conteúdo deles. Através do clique com o botão direito do mouse é possível selecionar os pontos a serem analisados;
- d) escolher o ângulo de rotação para o nível 2 em diante: existe a opção de alterar o ângulo de rotação dos “ramos” gerados a partir do nível 2 de iteração, possibilitando uma maior interação com o usuário na criação do fractal. Por padrão este campo vem com o valor de 120°;
- e) mostrar/ocultar ponto representativo dos arquivos dos diretórios: outra opção para maior interação com o usuário, permitindo mostrar ou não os pontos referentes aos arquivos. Por padrão este campo vem selecionado;
- f) mostrar/ocultar caminho absoluto dos diretórios: pode ser conveniente mostrar o caminho absoluto ou não dos diretórios. Por padrão este campo não vem selecionado;
- g) excluir arquivos ou diretórios selecionados: para manutenção dos diretórios e arquivos do sistema operacional existe a possibilidade de excluir diretórios e arquivos selecionados no fractal, sendo que excluirá também os subdiretórios e

arquivos existentes nos mesmos caso selecionado o diretório pai.

Os campos informados/alterados na tela terão validade após clicar no botão “Atualizar”.

Na Figura 26 verifica-se a tela de execução deste trabalho.

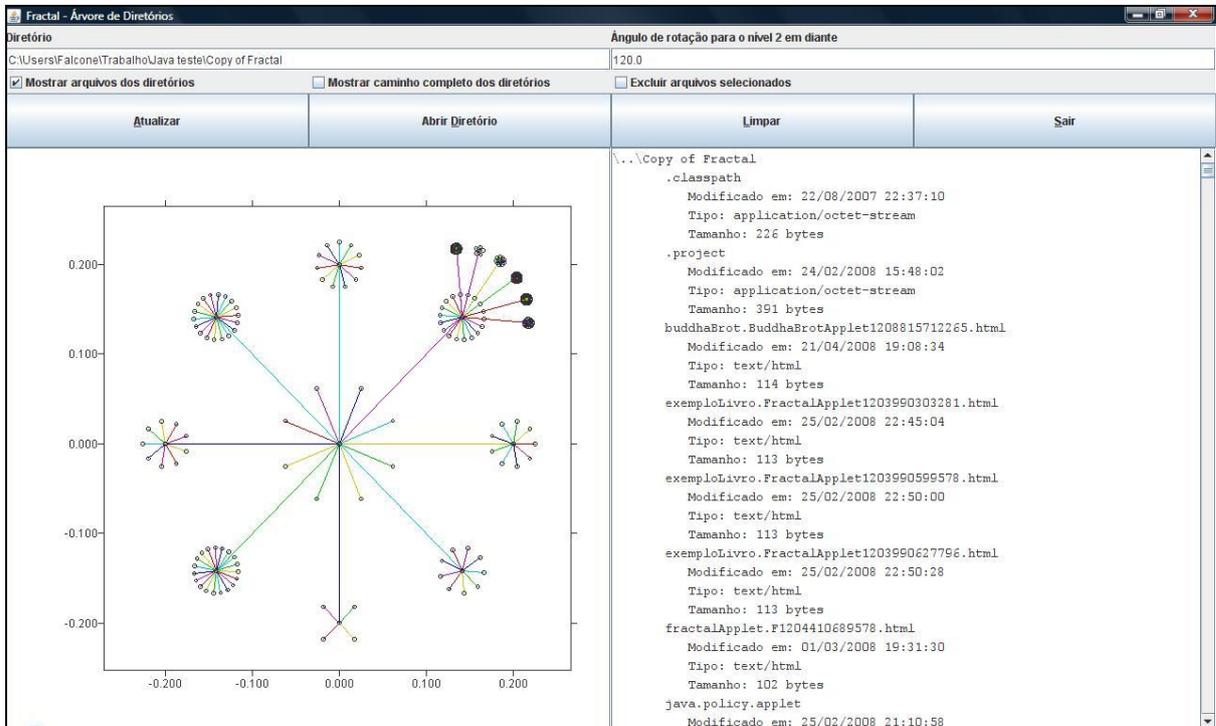


Figura 26- Tela de execução deste trabalho

4.4 RESULTADOS E DISCUSSÃO

Os estudos realizados na criação do Data Mountain por Cockburn e McKenzie (2001) demonstraram as vantagens da utilização da memória espacial em ambientes virtuais.

Entretanto, estes estudos não apresentam uma ferramenta direcionada ao gerenciamento do sistema de arquivos. Os outros trabalhos correlatos utilizados para gerência de arquivos não fazem uso dos conceitos de memória espacial.

A biblioteca Java JCKKit contribuiu para facilitar a representação e ampliação do fractal, sendo que esta biblioteca é indicada para a representação de gráficos e diagramas científicos.

A ferramenta se torna lenta ao selecionar um diretório com um número muito grande de subdiretórios ou o diretório raiz. Toda via, o protótipo consegue representar o diretório raiz quando não selecionado para mostrar os arquivos, neste caso os arquivos não são incluídos no

encadeamento da árvore.

Quando selecionado para mostrar os arquivos e escolhido o diretório raiz ou um diretório com muitos subdiretórios a ferramenta não finaliza o processamento e não mostra a imagem ocorrendo um erro de estouro de memória² da *Java Virtual Machine* (JVM), verificado na máquina³ de testes.

Verifica-se a necessidade de um melhor gerenciamento da memória para evitar problemas quando o número de arquivos for muito grande.

O fractal gerado neste trabalho pode ser considerado auto-afim (PINTO, 2001, p. 28), pois se verifica diferentes valores de dimensão fractal para escalas de observação diferentes. Isto ocorre pelo fato dos “ramos” subseqüentes a cada ponto refletirem o número de diretórios conforme organização do sistema de arquivos do usuário. Não existe um número fixo de arquivos ou subdiretórios para cada diretório, sendo assim, cada “ramo” analisado em cada nível da “árvore” possui sua dimensão fractal independente, conforme pode ser verificado na Figura 27.

² Erro mostrado no eclipse: Exception in thread "AWT-EventQueue-0" java.lang.OutOfMemoryError: Java heap space.

³ O protótipo foi testado em uma máquina com processador Intel Core 2 Duo de 1.67 GHz com 2 GB de memória RAM.

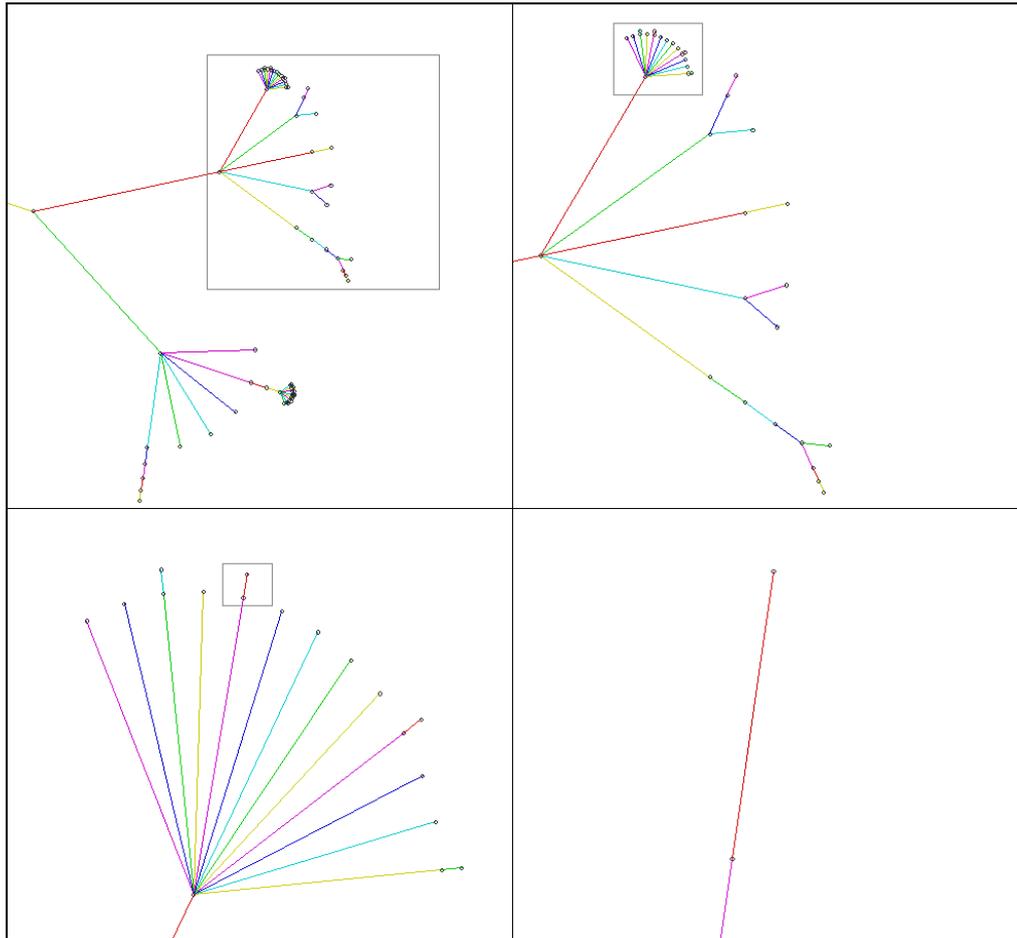


Figura 27 – Fragmentos do fractal gerado neste trabalho

5 CONCLUSÕES

O objetivo deste trabalho em criar uma infra-estrutura baseada em fractais para viabilizar a construção de um sistema de visualização de sistemas de arquivos semânticos foi concluído com êxito. Através desta infra-estrutura, será possível representar o sistema de arquivos na metáfora descrita em Krauspenhar (2007).

O grande diferencial deste projeto foi encontrar uma forma de representar os arquivos do sistema operacional através de um fractal. Fractal este que foi criado para este problema específico.

A criação da biblioteca para geração e manipulação do fractal também atingiu seu objetivo, tendo em vista a construção do protótipo deste trabalho, toda via foi implementado apenas a exclusão de arquivos/diretórios, sendo que a criação e a movimentação de arquivos/diretórios não foram feitos devido a restrições da biblioteca JCKKit. Pretender-se-ia fazer estas operações diretamente no gráfico, sendo totalmente visuais.

A manipulação dos arquivos ainda não está madura devido ao uso da biblioteca utilizada neste projeto, onde a mesma em sua concepção foi direcionada para representar gráficos e diagramas científicos. Seria interessante a construção de uma biblioteca com o objetivo específico para representação e manipulação dos arquivos do sistema operacional.

Este trabalho iniciado por Krauspenhar (2007) é uma prova de conceito, ou seja, buscou demonstrar a aplicabilidade de uma nova metáfora na forma de interação com os objetos do sistema de arquivos. Certamente muito trabalho tem de ser realizado de forma a possibilitar a utilização desta forma de interação por usuários acostumados a gerenciar arquivos na estrutura hierárquica.

5.1 EXTENSÕES

Como se verificou, a biblioteca JCKKit não se mostrou adequada para os propósitos do presente projeto. Assim sendo, uma possível extensão deve ser a construção de uma biblioteca específica para suportar o projeto descrito em Krauspenhar (2007). Outra possibilidade de extensão é o aprofundamento dos estudos utilizando a metáfora proposta como forma de interação do usuário com o sistema de arquivos.

Para uma melhor interação do usuário com o protótipo desenvolvido neste trabalho, sugere-se o uso de cores nos nomes dos diretórios/arquivos que estão representados graficamente, com intenção de uma visualização e associação com os pontos representativos do fractal mais eficaz.

Verifica-se a necessidade de uma gerência mais eficiente de memória, sendo assim propõe-se um estudo para este fim.

Para a visualização do gráfico do fractal sugere-se que a cada nova ampliação do desenho este não se distorça, devido ao campo selecionado não se adequar ao espaço de visualização. Para este problema propõe-se o uso do diagrama de Voronoi.

Sugere-se a aplicabilidade desta infra-estrutura na aplicação desenvolvida por Krauspenhar (2007) como uma forma de organizar a metáfora da cidade.

Contudo Krauspenhar (2007) levantou as seguintes necessidades para que possa ser melhorada. Aplicar novas funcionalidades que facilitem a operacionalidade, como controle de colisão entre os objetos aplicando conceitos da física, múltiplos ambientes, possibilidade de modificar a aparência do ambiente em tempo de execução, utilizar conceitos de arquivos semânticos e estender a aplicação para ser utilizada em outros sistemas operacionais. Contudo, o aspecto mais importante a ser trabalhado refere-se ao detalhamento das novas formas de interação com o ambiente metafórico criado e todas as conseqüências desta interação.

REFERÊNCIAS BIBLIOGRÁFICAS

- AGARAWALA, A.; BALAKRISHNAN, R. **Keepin' it real**: pushing the desktop metaphor with physics, piles and the pen. Toronto, 2006. Disponível em: <<http://honeybrown.ca/Pubs/BumpTop.pdf>>. Acesso em: 30 ago. 2006.
- BIRCHAL, M. A. S. Caos, fractais e o método computacional. **Revista Micro Sistemas**, Rio de Janeiro, ano XIII, n. 142, p. 12-15, set. 1994.
- BUFFONI, S. **A teoria do caos**. [S.l.], 2005. Disponível em: <<http://www.professores.uff.br/salete/caos.htm>>. Acesso em: 15 nov. de 2008.
- CHERQUES, H. R. T. O caos e a confusão. **Revista da ESPM**, São Paulo, v. 31, p. 5-21, 1999.
- DEITEL, H. M.; DEITEL, P. J. **Java**: como programar. 6. ed. Tradução Edson Furmankiewicz. São Paulo: Pearson, 2005.
- DUPONT, A.; ROSSETTI, H. R.; SIQUEIRA, R. A. **Janelas para o infinito**: exposição de fractais. [S.l.], 2008. Disponível em: < <http://www.fractarte.com.br/> >. Acesso em: 15 novembro 2008.
- ELMER, F. **JCKKit**: chart construction kit for the Java platform. [S.l.], 2005. Disponível em <<http://jckkit.sourceforge.net/>>. Acesso em: 23 nov. 2008.
- FARHOOMAND, A. F.; DRURY, D. H. Managerial information overload. **Communications of the ACM**, Nova Iorque, v. 45, n. 10, p. 22-36, out. 2002.
- GLEICK, J. **Caos**: a criação de uma nova ciência. 4. ed. Tradução Waltensir Dutra. Rio de Janeiro: Campus, 1991.
- HENDERSON, S. Personal digital document management. In: CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS, 24., 2004, Vienna. **Proceedings...** Vienna: [s.n.], 2004. p. 20-25.
- KRAUSPENHAR, V. C. R. **Ferramenta de visualização 3d de um sistema de arquivos**. 2007. 56 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.
- LECHY-THOMPSON, G. W. **Infinite game universe**: mathematical techniques. Hingham, Massachusetts: Charles River Media. 2001. 306 p.
- MANDELBROT, B. B. **The fractal geometry of nature**. New York: W. H. Freeman, 1983.

- MATTOS, M. M.; TAVARES, A. C.; KRAUSPENHAR, V. C. R. Construção de um sistema de visualização de arquivos 3D. In: WORKSHOP DE SISTEMAS OPERACIONAIS, 27., 2007, Rio de Janeiro. **Anais...** Rio de Janeiro: Congresso da SBC, 2007. p. 910-915.
- MESQUITA, M.; MOTA, M. G. Fractais: a linguagem do caos. **Anais eletrônicos...** Lisboa, jul./set. 1991. Disponível em: <<http://www.albertomesquita.net/am/fractais/capa.asp>>. Acesso em: 8 novembro 2008.
- MCKAY, J; MARSHALL, P. **The dual imperatives of action research**. Churchlands, 2001. Disponível em: <http://www.utas.edu.au/.../research/Qual_Research/mckay%20&%20marshall_dual%20imperatives%20of%20ar.pdf>. Acesso em: 7 set. 2006.
- OLIVEIRA, L. H. A matemática do delírio. **Revista Superinteressante**, São Paulo, v. 85, n. 10, p. 22-27, out. 1994.
- PINTO, S. C. D. **Estimação da dimensão fractal de imagens de SPM**. 2001. 133 f. Dissertação (Mestrado em Ciências Física Aplicada) - Instituto de Física de São Carlos, Universidade de São Carlos, São Carlos.
- PLOTZE, R. O.; BRUNO, O. M. Análise de formas e reconhecimento de padrões por meio da assinatura fractal multiescala. **INFOCOMP - Journal of Computer Science**, Lavras, v. 6, n. 2, p. 91-100, jun. 2007.
- REZENDE, S. M. A dança dos Spins. **Revista Ciência Hoje**, Rio de Janeiro, v. 14, n. 80, p. 29-32, mar./abr. 1992.
- ROBERTSON, G. et al. **Data Mountain**: using spatial memory for document management. Vermont, 1998. Disponível em: <<http://www.microsoft.com/usability/UEPostings/p153-robertson.pdf>>. Acesso em: 22 nov. 2008.
- ROBERTSON, G. et al. **Task Galley**: a 3D window manager. Vermont, 1999. Disponível em: <<http://research.microsoft.com/~marycz/tg2000.pdf>>. Acesso em: 22 nov. 2008.
- SANTOS, J. C. **Protótipo de uma ferramenta baseada em sistemas de Lindenmayer para Modelagem de estruturas fractais**. 1999. 128 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.
- SHINYAMA, Y. **XCruiser**. [S.l.], 2003. Disponível em: <<http://xcruiser.sourceforge.net>>. Acesso em: 22 nov. 2008.
- SIQUEIRA, R. A. **Fractais**: arte e ciência. [S.l.], 2006. Disponível em: <<http://www.insite.com.br/rodrigo/misc/fractal/>>. Acesso em: 16 novembro de 2008.
- TANENBAUM, Andrew S.; WOODHULL, Albert S. **Sistemas operacionais**: projeto e implementação. 2. ed. Tradução Edson Furmankiewicz. Porto Alegre: Bookman, 2000.

UNIVERSITY OF BRISTOL. **Neural basis of spatial memory**. Bristol, [2001?]. Disponível em: <<http://www.bris.ac.uk/depts/Synaptic/research/projects/memory/spatialmem.htm>>. Acesso em: 28 ago. 2006.