

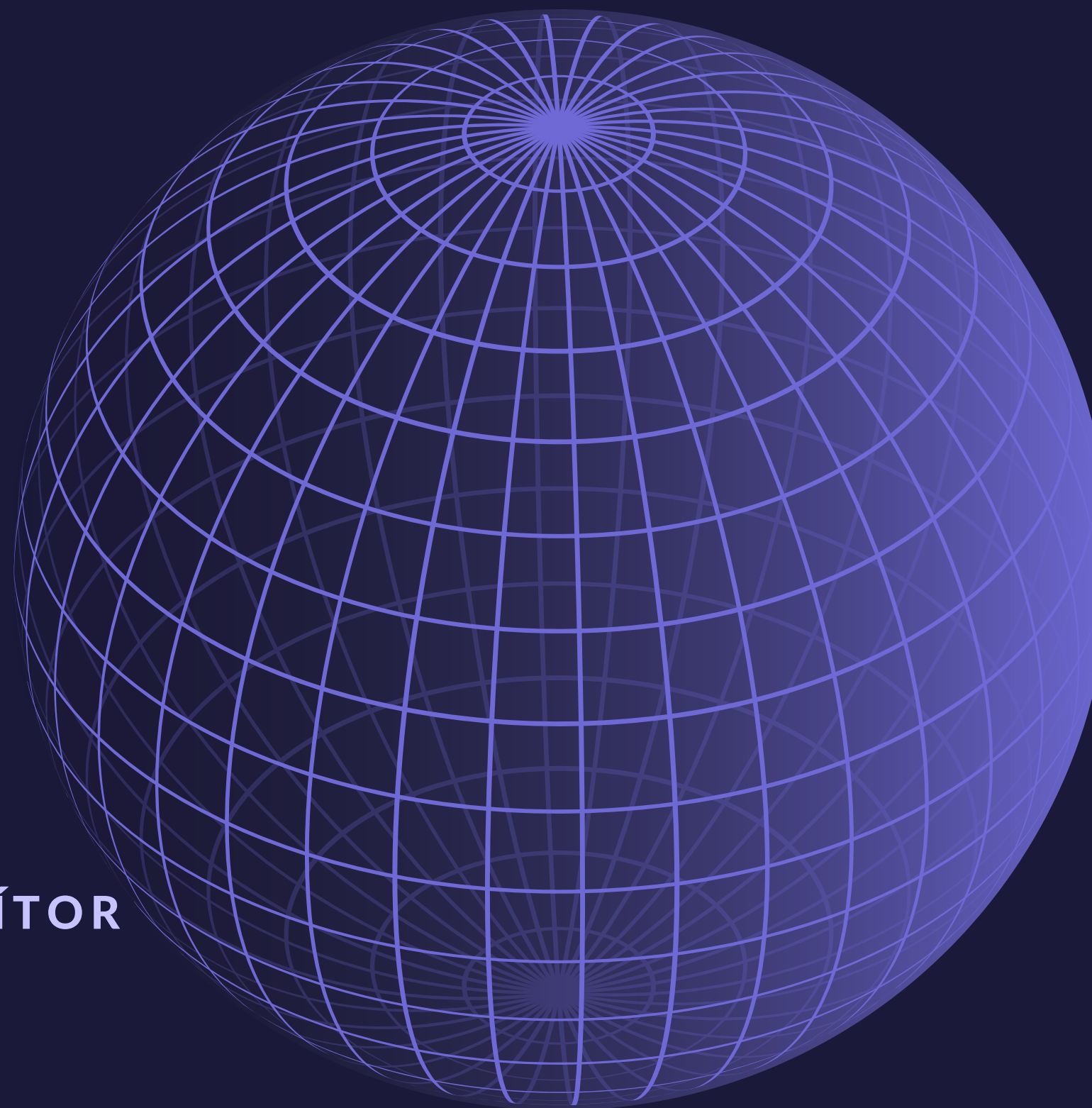


000

ESTRUTURA DE DADOS ESPACIAIS: GRID

ESTRUTURA DE DADOS II
GUILHERME TAVARES DE ASSIS

DANILO CÉSAR, ENYA SANTOS, JOÃO VÍTOR
VAZ, NATHANN ZINI, VITÓRIA BISPO





SUMÁRIO

- Introdução
- O que é ?
- Aplicação
- Código
- Vantagens e Desvantagens
- Grid Files



INTRODUÇÃO

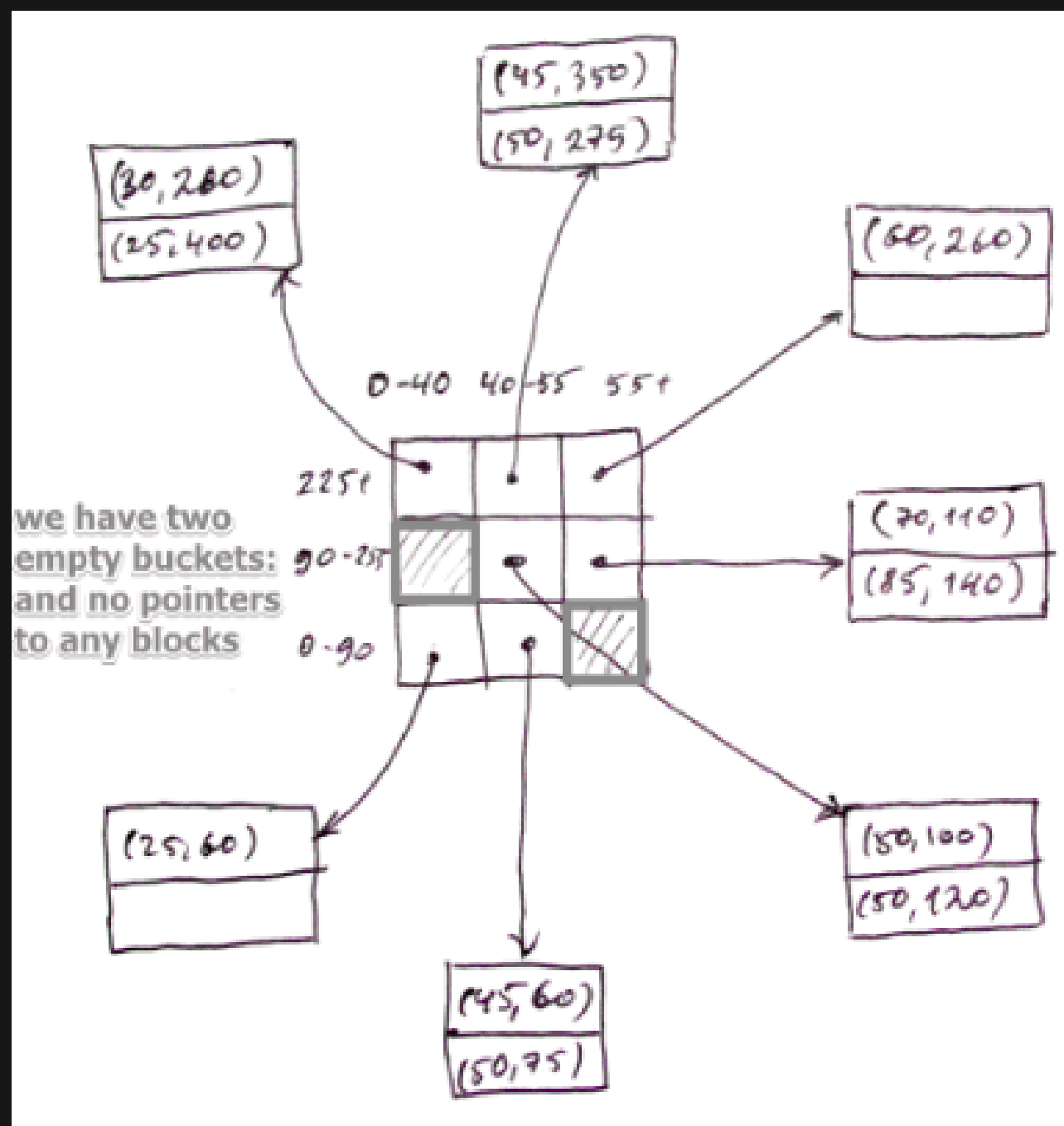
- Definição de Dados Espaciais;
- Definição de Estrutura de Dados Espaciais;
- Origem do Grid;

O QUE É?

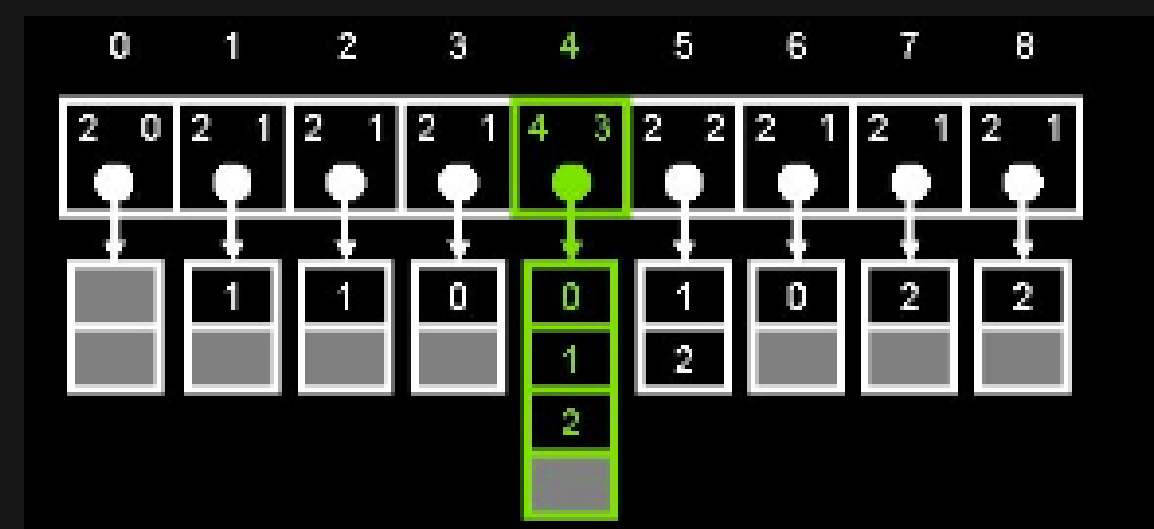
- Possibilita a divisão de dados espaciais em repositórios (*buckets*) de tamanho fixo em cada dimensão;
- Dados localizados em um mesmo repositório ou célula são armazenados numa área contígua de armazenamento;
- Portanto, um repositório deve ser capaz de armazenar uma quantidade de dados espaciais que possam ser armazenados em um mesmo bloco de disco;
- Os dados são recuperados através da “localização” de seus *buckets*.



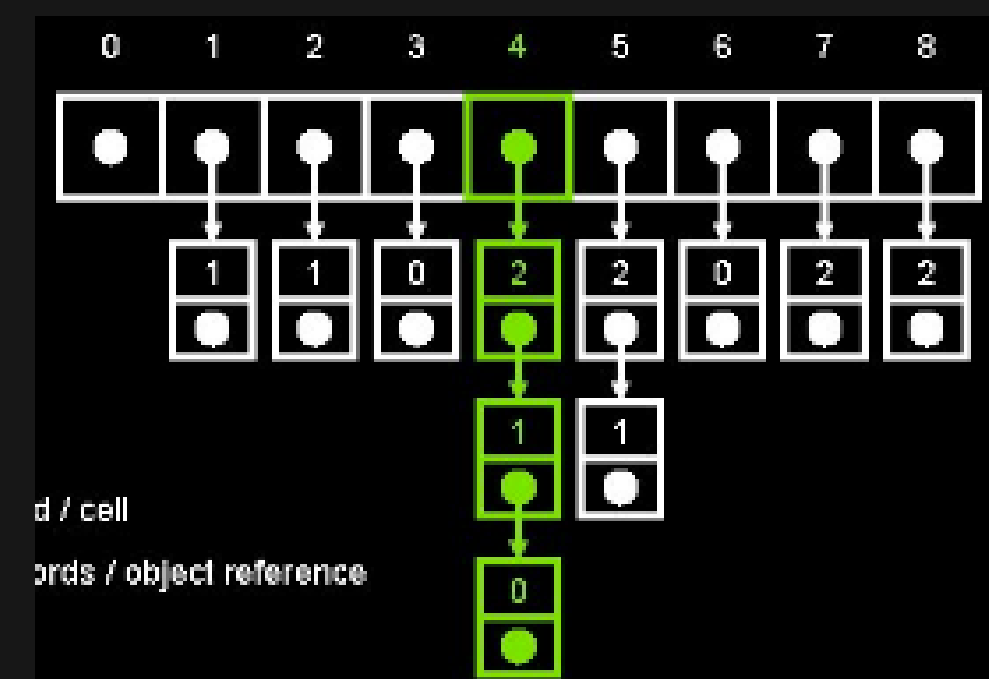
004



Representação da Grid



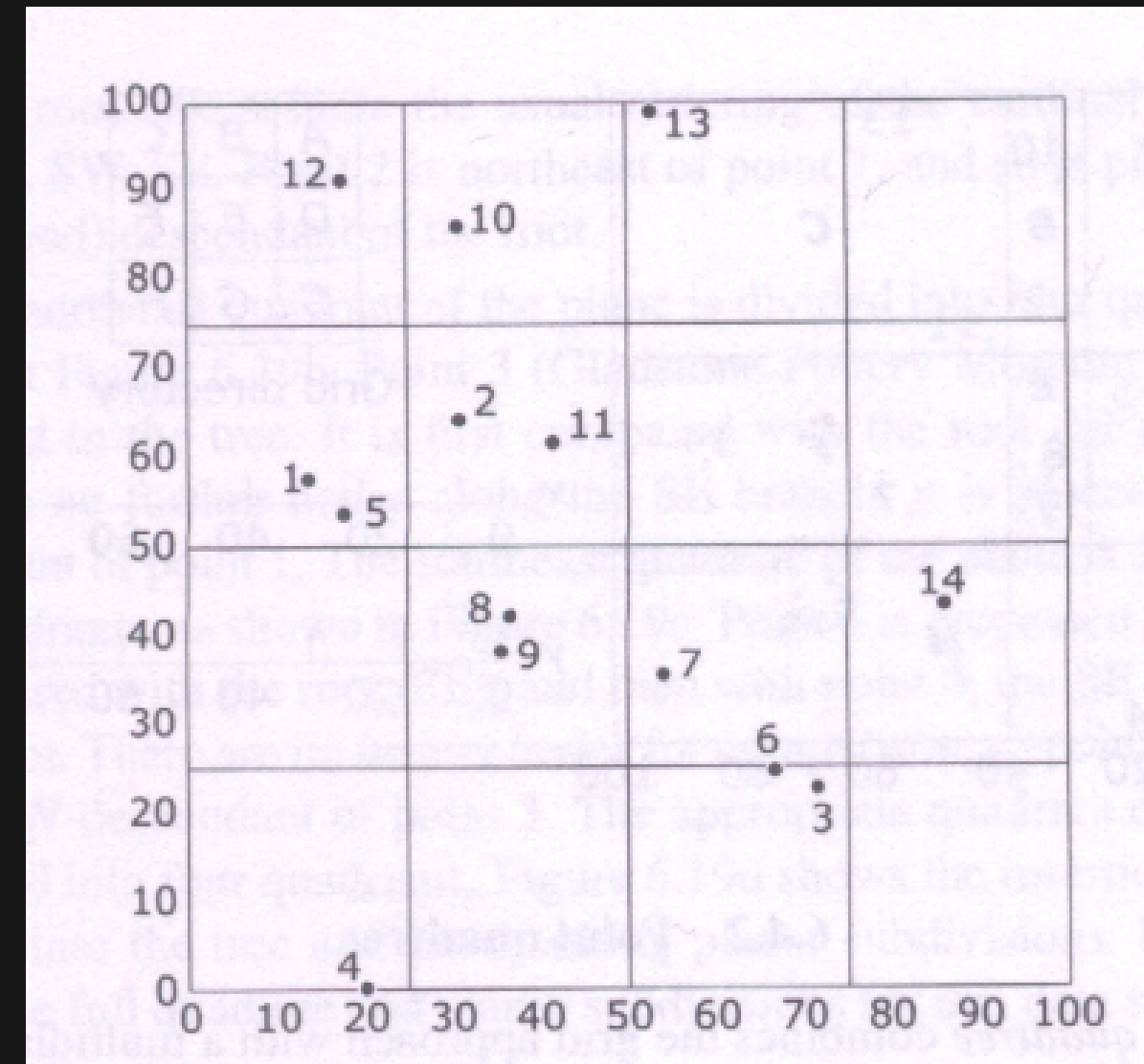
Grid com array dinâmico



Grid com lista encadeada

EXEMPLO

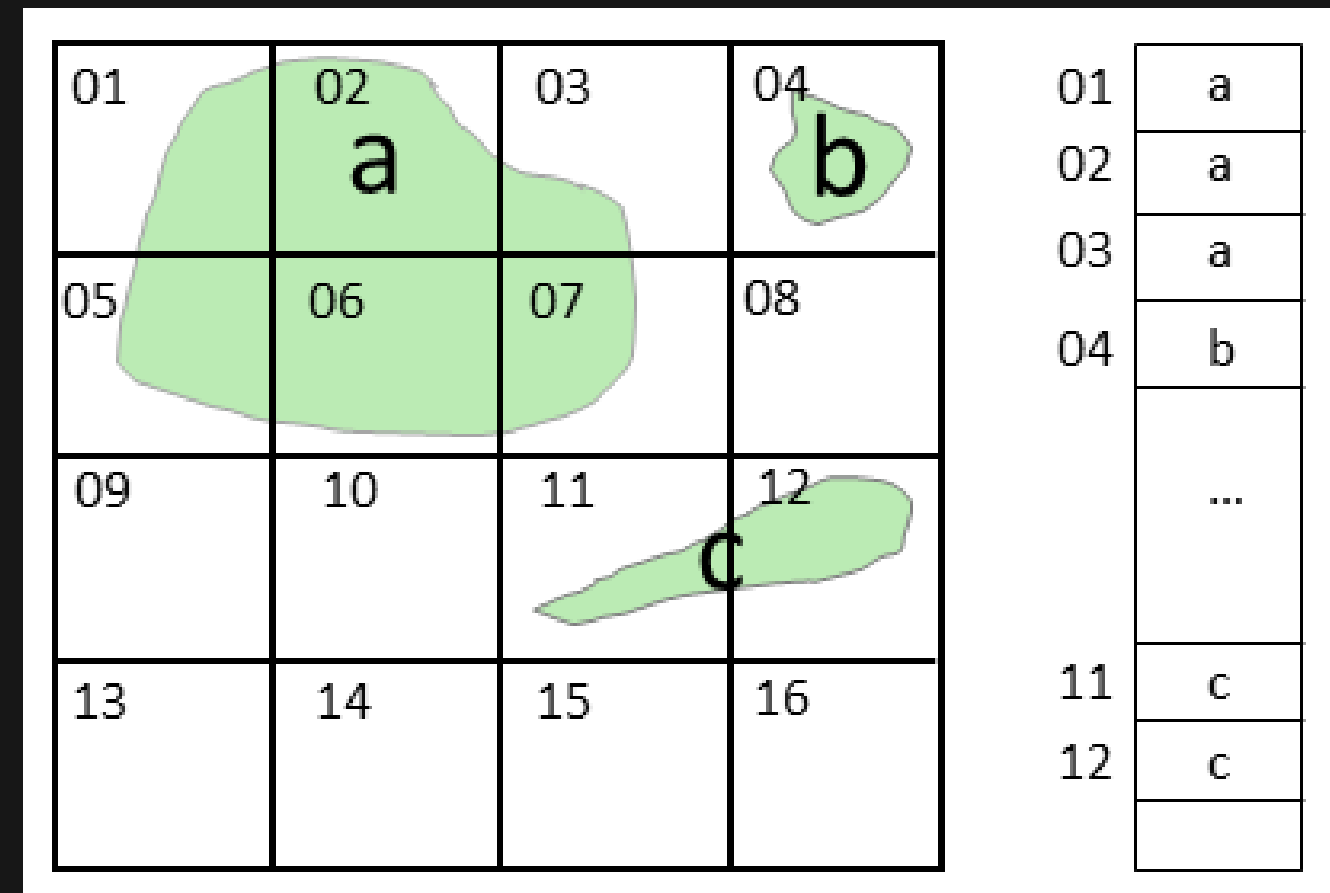
- Pontos como (8 e 9), (1 e 5), (2 e 11).
- O ponto 6 intercepta a linha da grid para resolver esses problemas e o desenvolvedor determina a conversão que irá usar.



Representação da Grid

EXEMPLO

- Há objetos espaciais que serão representados na grid que podem ocupar mais de uma célula.
- As representações mostraram exemplos em representação de espaços 2D, mas também pode ser usado para 3D / 4D.

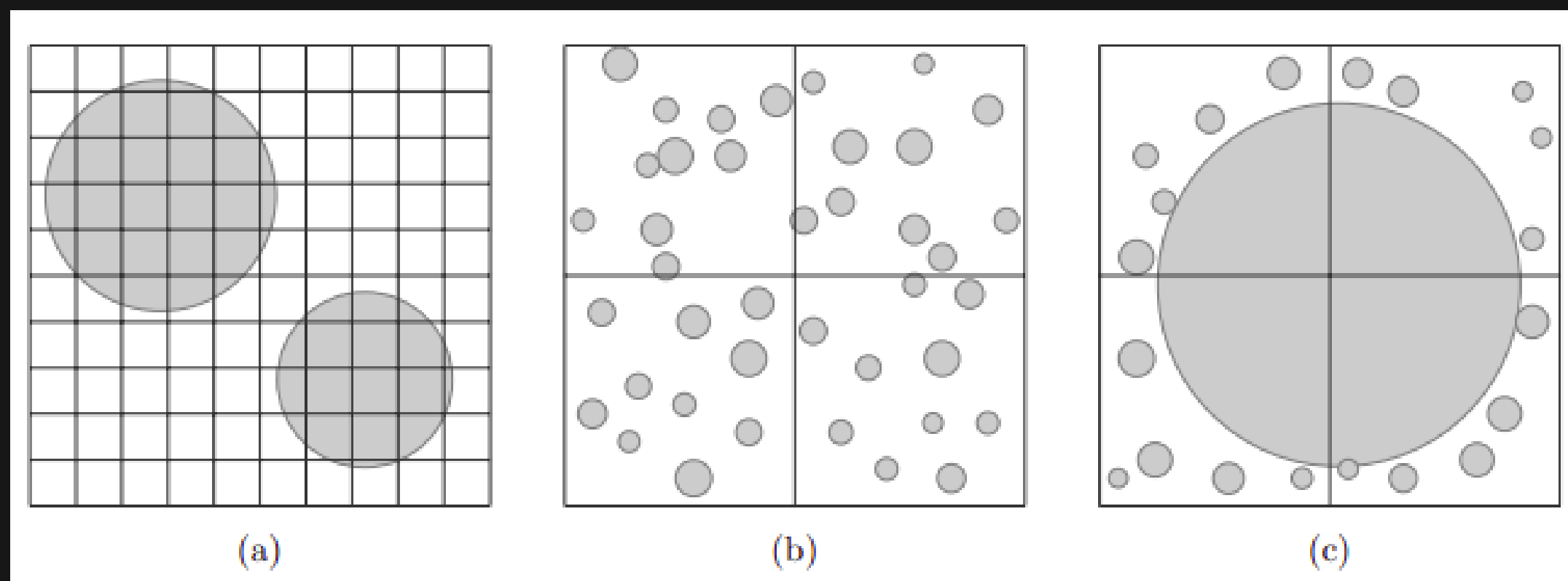


Representação da Grid

ESCOLHA DO TAMANHO E NÚMERO DE CÉLULAS

- A parte mais crítica do particionamento espacial da grade uniforme é escolher um apropriado tamanho da célula.
- Este valor é importante pois define o desempenho da busca espacial na estrutura, também, pode ser problemático em certas situações.

ESCOLHA DO TAMANHO DE NÚMERO DE CÉLULAS



(a) a grade é muito pequena em relação ao tamanho do objeto.

(b) a grade é muito grande em relação ao tamanho do objeto.

(c) a grade é muito grande para os objetos pequenos, mas é muito pequena para os objetos grandes.

LOGO O TAMANHO IDEAL DO BUCKET É DEPENDENTE DE 2 CONDIÇÕES:

- Quanto maior o número total de pontos, maior o número de células requeridas
- O tamanho da célula é dependente na magnitude da faixa média das range query suportadas pelo sistema



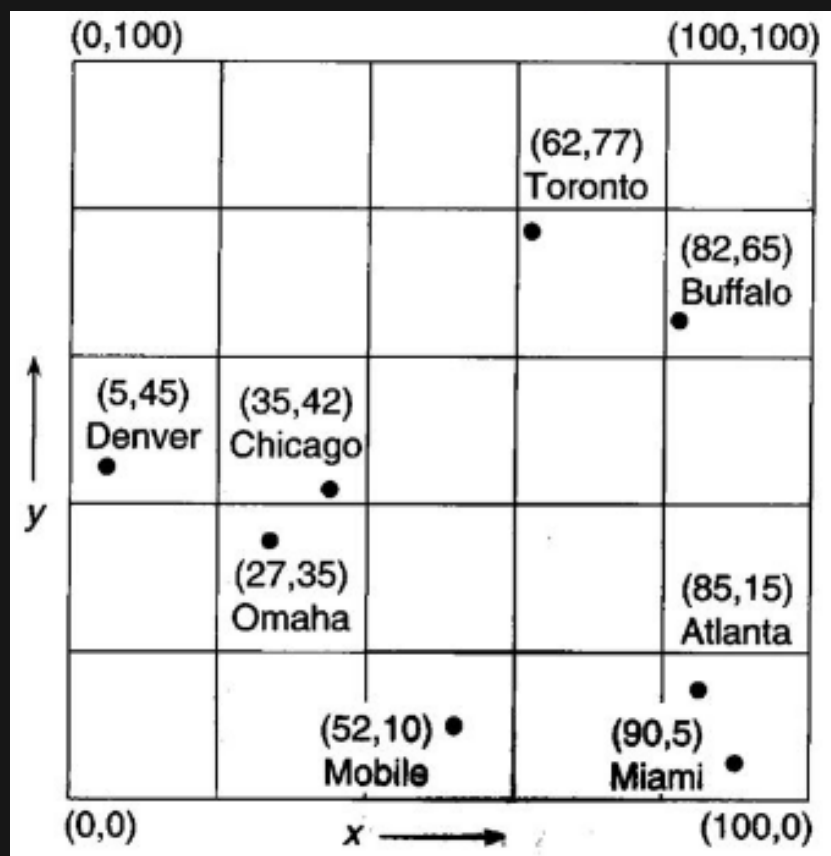
O GRID

Grid é útil para solucionar dois tipos de problemas:

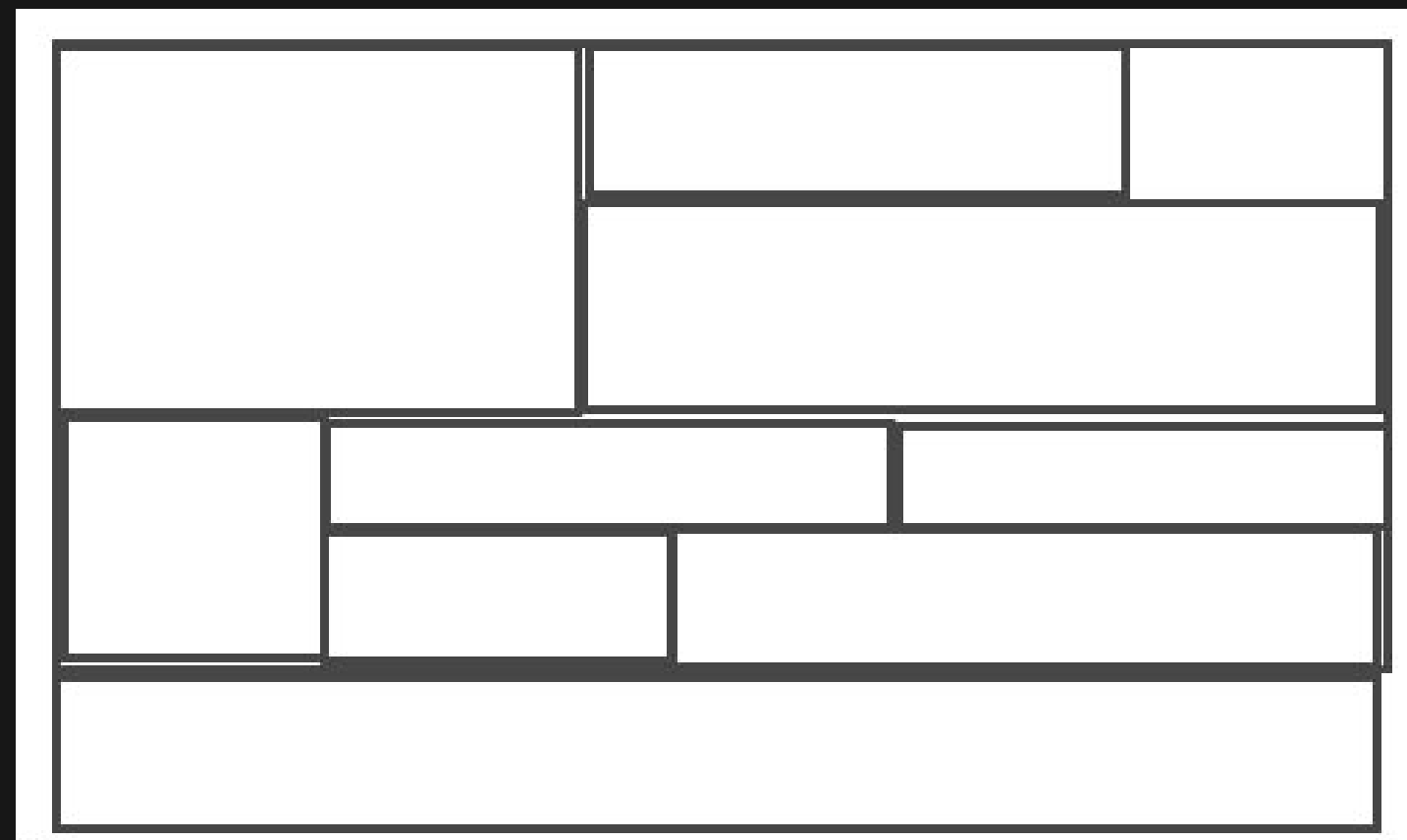
1. **Point Query:** encontrar um ponto que atende uma condição específica.
2. **Range Query:** encontrar um conjunto de pontos que se encaixam em um determinado range.

GRID REGULAR/UNIFORME E GRID IRREGULAR

Ao contrário do Grid uniforme, o Grid Irregular particiona o espaço em um número distinto de divisões para cada eixo de coordenadas ($N_x \times N_y \times N_z$).



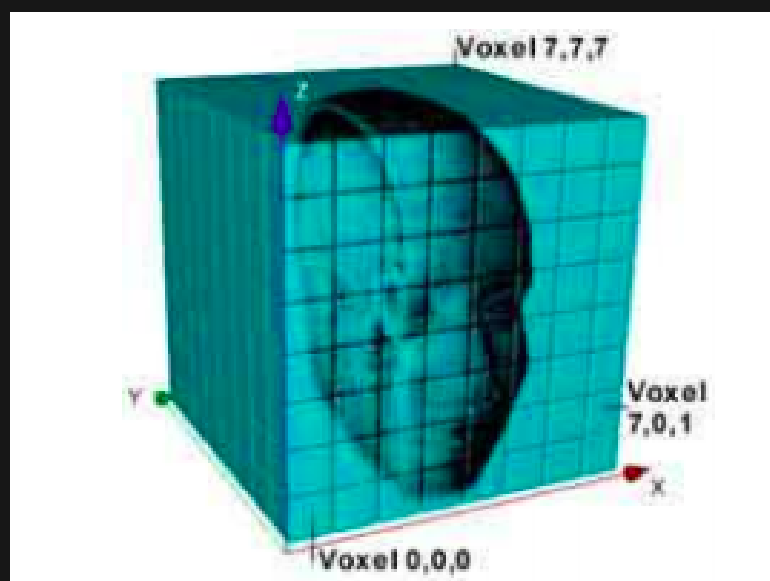
Exemplo de Grid Uniforme



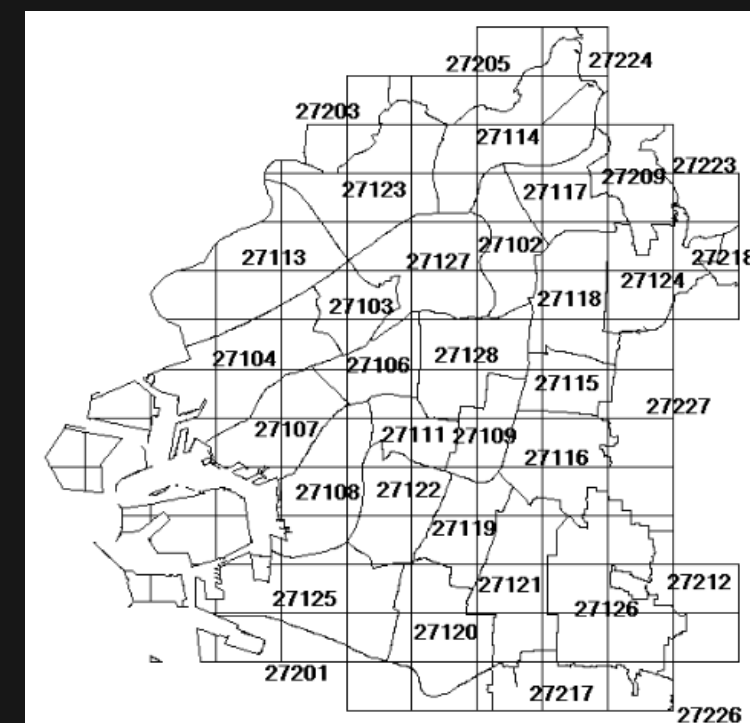
Exemplo de Grid Irregular

APLICAÇÃO

- Computação gráfica
 - Modelagem 3D
- SIG (Sistema de informação Geográfica)
- Jogos



Volume Graphics, 2005)



Exemplo da aplicação em SIG



Jogo de RPG



CÓDIGO

Estrutura

```
#include <stdbool.h>
#define GRID_SIZE 4

//Estrutura com a definição das coordenadas dos itens
typedef struct{
    double x, y;
    //Outros atributos da estrutura
}Item;

//Estrutura que representa os Buckets
typedef struct{
    Item *itens;
    Item *prox; //Endereço do próximo
}Bucket;

// Estrutura que representa o grid com GRID_SIZE*GRID_SIZE buckets
typedef struct{
    Bucket grid[GRID_SIZE][GRID_SIZE];
}Grid;
```

CÓDIGO

Inserção

```
void adicionarItem(Grid *grid, Item *item){  
  
    int insX, insY;  
    //Encontrar posição de inserção  
    acharPos(&insX, &insY, *item);  
  
    //Adicionar no bucket da posição [insX][insY]  
    adicionarGr(&grid->grid[insX][insY], item);  
}  
  
void acharPos(int *posX, int *posY, Item item){  
    //Para essa implementação, utilizamos o valor arredondado para baixo  
    *posX = floor(item.x);  
    *posY = floor(item.y);  
}
```



CÓDIGO

Remoção

```
bool removerItem(Grid *grid, Item *item){
    //Verificação para saber se o item existe dentro do GRID
    if(pesquisa(grid, item)){
        int posX, posY;
        //Encontrar posição do item no GRID para realizar a remoção
        acharPos(&posX, &posY, *item);

        //Remover o item no bucket encontrado [posX][posY]
        retirada(&grid->grid[posX][posY], item);

        return true;
    }
    return false;
}
```




CÓDIGO

Pesquisa

```
bool pesquisa(Grid *grid, Item *item){  
    int posX, posY;  
    //Encontrar posição interia de onde o item desejado está localizado  
    acharPos(&posX, &posY, *item);  
  
    //Buscar dentro do bucket  
    bool result = pesquisarGr(&grid->grid[posX][posY]);  
  
    //Encontrou ou não  
    return result ? true : false;  
}
```

Vantagens

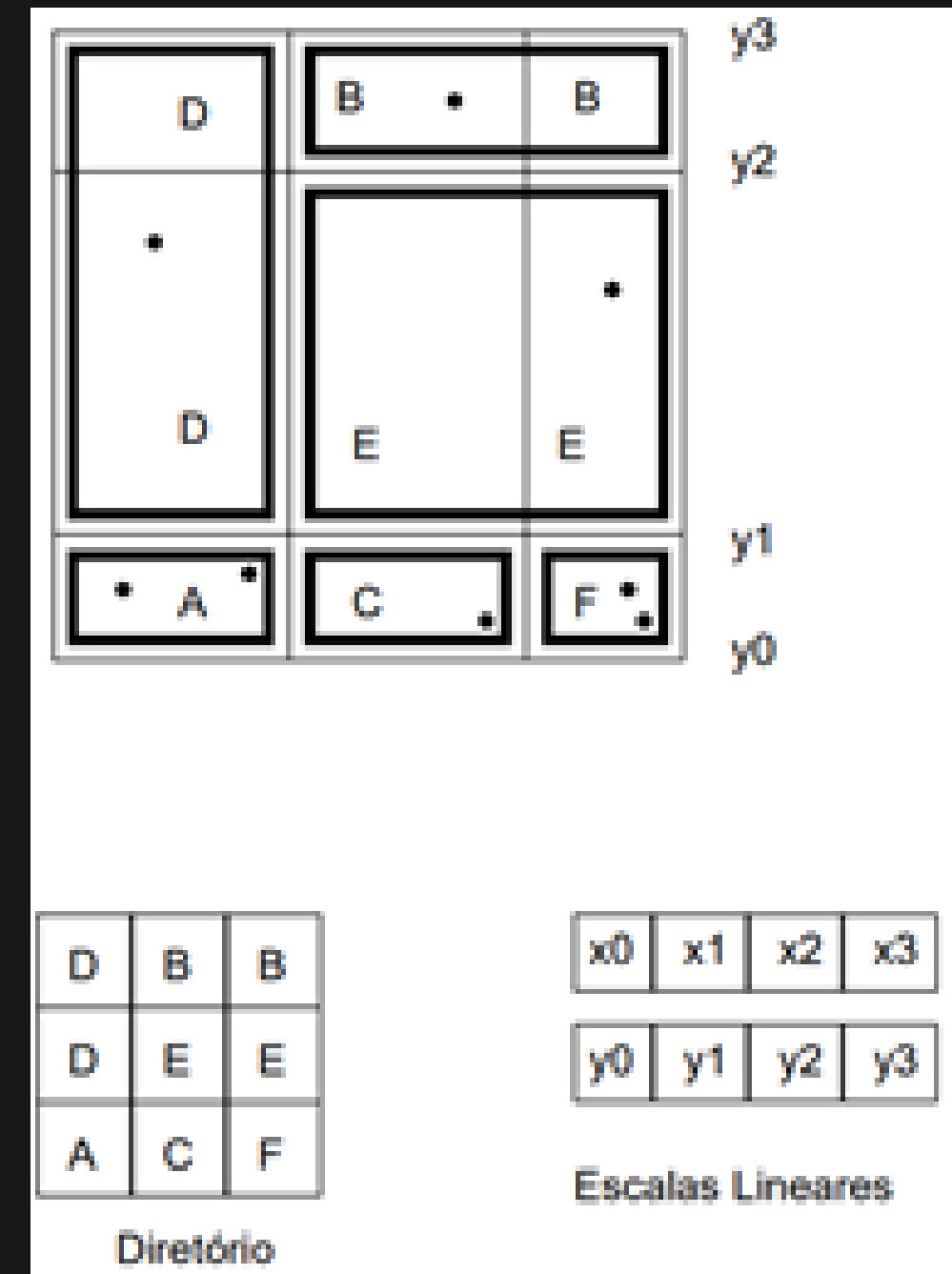
- Pesquisa de chaves:
 - Pesquisa de uma única chave;
 - Retorno de pesquisa;

Desvantagens

- Limite de utilização do método com base no espaço;
- Cálculo do tamanho do bucket/célula (hash);
- Quantidade e distribuição dos pontos.

GRID FILES

- O que é;
- Estrutura;
- Exemplo.





ALGUMA PERGUNTA?



REFERÊNCIAS

Disponível em: <<https://www.youtube.com/watch?v=gZZDIQ2RZao/>>.
Acesso em: 20 de Agosto de 2021.

ESPERANÇA, Cláudio. Processamento Geométrico Bancos de Dados Espaciais Sistemas de Informações Geográficos (GIS). In: Estruturas de Dados Espaciais: Processamento Geométrico Bancos de Dados Espaciais Sistemas de Informações Geográficos (GIS). Disponível em: <http://orion.lcg.ufrj.br/gc/download/ede.pdf>. Acesso em: 20 ago. 2021.

GRID. FERREIRA, Arthur, 2014. Disponível em:
<<https://prezi.com/1wfvudsxrein/grid/>> Acesso em 20 de Agosto
2021.

TADDEO, Leandro. 2005. Detecção de Colisão utilizando Grids e
Octrees Esféricas para Ambientes Gráficos Interativos. FUNDAÇÃO
EDSON QUEIROZ UNIVERSIDADE DE FORTALEZA

DM-66 - Spatial Indexing. 2017. Disponível em:
<<https://gistbok.ucgis.org/bok-topics/spatial-indexing>> Acesso em 22
de Agosto de 2021.

Disponível em:<http://mlwiki.org/index.php/Grid_File_Index>Acesso em 22 de Agosto de 2021.

Compact, Fast and Robust Grids for Ray Tracing. LAGAE, Ares; DUTRÉ,Philip. 2018. Disponível em:<<https://slidetodoc.com/19-th-eurographics-symposium-on-rendering-compact-fast/>>Acesso em 22 de Agosto de 2021.

Jonathan. 2017. Disponível em:<<https://www.jonathanyu.xyz/2017/01/14/data-structures-for-tile-based-games/>> Acesso em 22 de Agosto de 2021.