

**CETM**  
Curso de Especialização em  
Tratamento de Minérios

# Modelamento e simulação de processos

5. Autômato Celular

Prof. Dr. André Carlos Silva

# 1. INTRODUÇÃO

- Na década de 1940, Stanislaw Ulam estudou o crescimento de cristais no Laboratório Nacional de Los Alamos, modelando-o usando uma matriz.
- Ao mesmo tempo, John von Neumann, colega de Ulam em Los Alamos, trabalhava em sistemas auto-replicativos e encontrava dificuldades para explicitar o seu modelo inicial de um robô que fosse capaz de se copiar sozinho a partir de um conjunto de peças separadas.

# 1. INTRODUÇÃO

- Ulam sugeriu-lhe que se inspirasse em seus trabalhos, o que conduziu von Neumann a conceber um modelo matemático abstrato para seu problema.
- O resultado foi o “*copiador e construtor universal*”, o primeiro autômato celular (ou simplesmente AC), baseado numa matriz com duas dimensões onde cada célula podia estar em um de 29 estados possíveis.

# 1. INTRODUÇÃO

- Em 1969, Konrad Zuse publicou o livro *Rechenender Raum* (“Calcular o espaço”) onde adiantou a hipótese de as leis físicas serem discretas e que o universo era o resultado de um gigantesco autômato celular.

# 1. INTRODUÇÃO

- Nos anos 1970, um autômato celular de duas dimensões e dois estados, chamado de o “*Jogo da Vida*”, inventado por John Horton Conway, atingiu grande sucesso, particularmente entre a comunidade nascente da informática.

# 1. INTRODUÇÃO

- Foi popularizado por Gardner (1970) num artigo para a revista *Scientific American*.
- O *Jogo da Vida* é um autômato celular que simula processos de evolução de células biológicas e ainda segundo o autor é um autômato “computacionalmente universal”, ou seja, potencialmente seria capaz de simular qualquer sistema possível.

# 1. INTRODUÇÃO

- As regras deste autômato celular são as seguintes:
  - Qualquer célula viva com menos de dois vizinhos vivos morre de solidão;
  - Qualquer célula viva com mais de três vizinhos morre por superpopulação;
  - Qualquer célula morta com exatamente três vizinhos se torna uma célula viva;
  - Qualquer célula com dois vizinhos vivos continua no mesmo estado para a próxima geração.

# 1. INTRODUÇÃO

- O *Jogo da Vida* em si consiste em escolher uma configuração inicial de células vivas tais que elas acabem por sobreviver ao final da simulação.
- Em 1983, Stephen Wolfram publicou a primeira de uma série de publicações em que o autor analisou de maneira sistemática autômatos celulares muito simples.

# 1. INTRODUÇÃO

- A complexidade do seu comportamento, induzida por regras elementares, levou-o a conjecturar que mecanismos similares poderiam esclarecer fenômenos físicos complexos, idéias estas que desenvolveu em seu livro (Wolfram, 2002).
- Rucker (2005) expandiu as teorias de Wolfram para uma teoria do Automatismo Universal, que usa os autômatos celulares como um modelo para explicar como regras simples podem gerar resultados complexos.

# 1. INTRODUÇÃO

- Segundo esta teoria, tudo que existe no universo (o tempo meteorológico, a forma das folhas das árvores ou dos continentes, o movimento das estrelas, os processos da mente, etc.) tem por base algoritmos simples capazes de gerar a complexidade que vemos nos mundos da física, biologia, sociedade, cultura e até da psicologia.

# 1. INTRODUÇÃO

- Os autômatos celulares são modelos computacionais onde o tempo e o espaço são ambos discretizados e a evolução do sistema obedece a regras específicas pré-determinadas.
- Esses modelos permitem estudar sistemas de grande tamanho em tempo e custo computacionais aceitáveis.

# 1. INTRODUÇÃO

- Um autômato celular é um modelo discreto, consistindo em uma matriz finita e regular de células, cada uma podendo estar em um número também finito de estados, que variam de acordo com regras determinísticas.
- O tempo também é discreto e o estado de uma célula no tempo  $t_i$  é uma função do estado no tempo  $t_{i-1}$  de um número finito de células na sua vizinhança.

# 1. INTRODUÇÃO

- Essa vizinhança corresponde a uma determinada seleção de células próximas (podendo eventualmente incluir a própria célula).
- Todas as células evoluem segundo a mesma regra de atualização, baseada nos valores das suas células vizinhas e em alguns casos dela própria.

# 1. INTRODUÇÃO

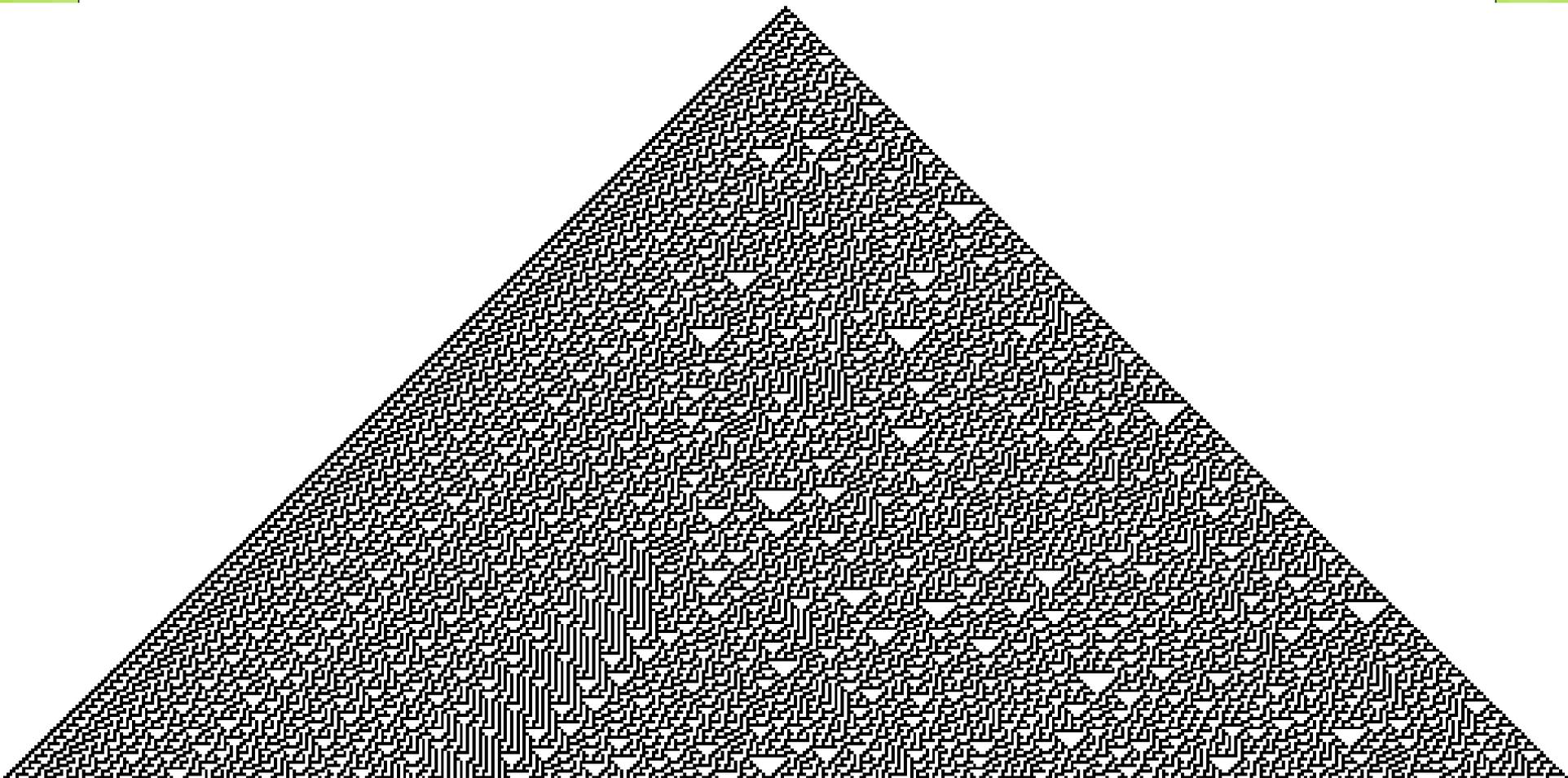
- Cada vez que as regras são aplicadas à matriz completa, uma nova matriz é produzida.
- Assim sendo qualquer sistema com muitos elementos idênticos que interagem local e deterministicamente pode ser modelado usando autômatos celulares.

# 1. INTRODUÇÃO

- O autômato celular mais simples e não trivial é unidimensional, com dois estados possíveis por célula e sendo os vizinhos de uma célula as células adjacentes de cada lado desta.
- Uma célula e as suas duas vizinhas formam uma vizinhança de 3 células, por isso existem  $2^3 = 8$  padrões possíveis para uma vizinhança. Há, então,  $2^8 = 256$  regras possíveis.

# 1. INTRODUÇÃO

- Os autômatos são referidos usualmente pelo número decimal que, em binário, representa a tabela da regra.
- A representação gráfica da evolução de um autômato com a regra 30 (em binário 11110) começando com um padrão de entrada inicial com apenas um 1 no centro é mostrada na figura abaixo.



<b>Padrão atual</b>	111	110	101	100	011	010	001	000
<b>Novo estado para a célula central</b>	0	0	0	1	1	1	1	0

# 1. INTRODUÇÃO

- Uma tabela define a regra de um autômato porque indica o que vai acontecer para cada um dos padrões possíveis para uma vizinhança.
- Por exemplo, a tabela da regra 30 diz que se três células adjacentes têm atualmente o padrão 100 (célula da esquerda tem valor 1, e as outras 0) ou 001 (célula da direita com valor 1, e as outras 0) então a célula do meio tornar-se-á 1 na próxima iteração.

# 1. INTRODUÇÃO



Padrão atual	111	110	101	100	011	010	001	000
Novo estado para a célula central	0	0	0	1	1	1	1	0

**Se três células adjacentes têm atualmente o padrão 100 (célula da esquerda tem valor 1, e as outras 0) ou 001 (célula da direita com valor 1, e as outras 0) então a célula do meio tornar-se-á 1 na próxima iteração.**

# 1. INTRODUÇÃO

***A regra 30 gera uma seqüência aparentemente aleatória.***

- Wolfram propôs usar a sua coluna central como um gerador de números pseudo-aleatórios e este usa esta regra no seu programa *Mathematica* para criar números inteiros aleatórios.

# 1.1. AC naturais

- Os desenhos de certas conchas de moluscos são gerados como autômatos celulares naturais.
- As células responsáveis pela pigmentação estão situadas sobre uma banda estreita ao longo da boca da concha.

# 1.1. AC naturais

- Cada célula segrega pigmentos de acordo com a segregação (ou ausência de segregação) das suas vizinhas e o conjunto das células produz o desenho da concha à medida que ela cresce.
- Como exemplo tem-se a espécie *conus textile* que apresenta um desenho parecido com o de um autômato celular de regra 30.



Espécie *conus textile* e o desenho da sua concha.

## 2. Rede e célula

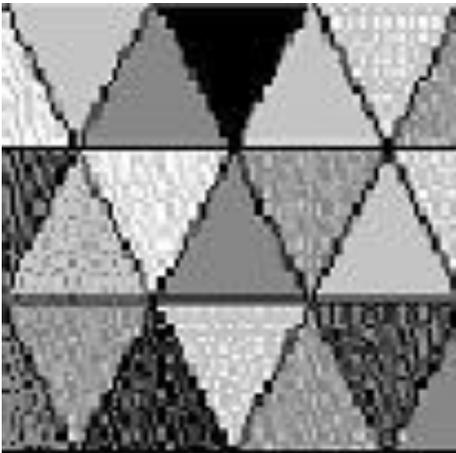
- Nos autômatos celulares tanto o **tempo** quanto o **espaço** são discretos.
- Desta forma os estados das células na rede são bem definidos.
- Regras de evolução (ou atualização) de uma dada célula dependem da vizinhança.

## 2. Rede e célula

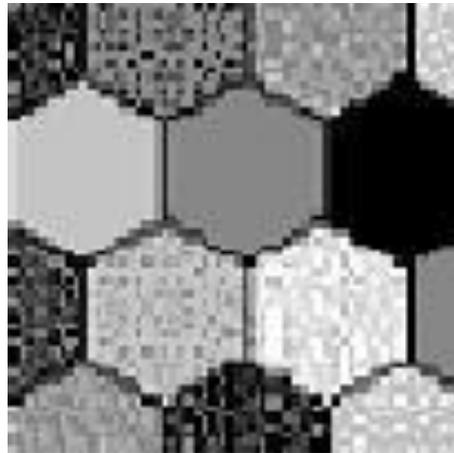
- **Rede** é toda a região que será usada na simulação.
- Uma rede é dividida em células.
- Uma **célula** (ou sítio) por sua vez é o menor pedaço da rede.

## 2. Rede e célula

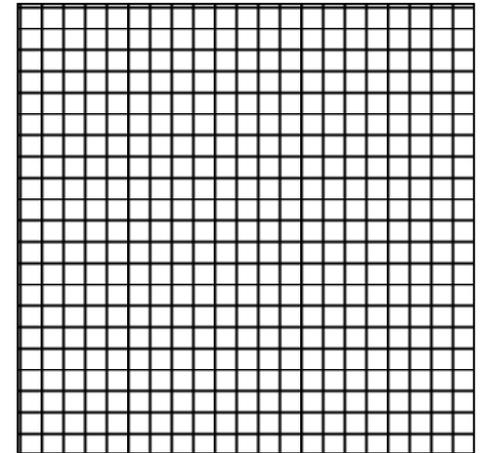
- As células de uma rede podem ter de várias formas, desde que num mesmo autômato todas as células sejam de uma mesma forma.



Triangular



Hexagonal



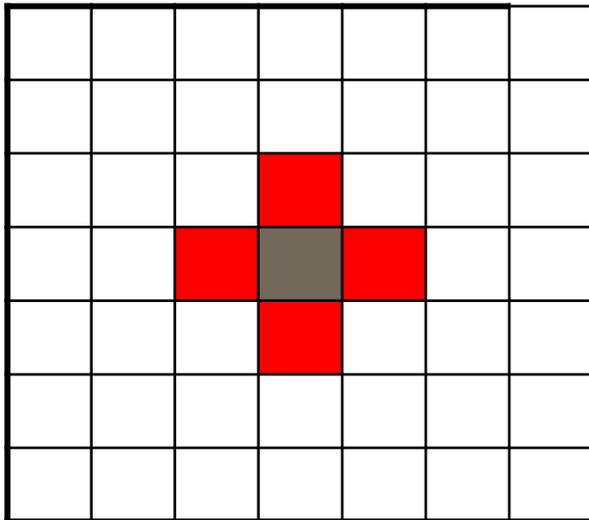
Quadrangular

## 2. Rede e célula

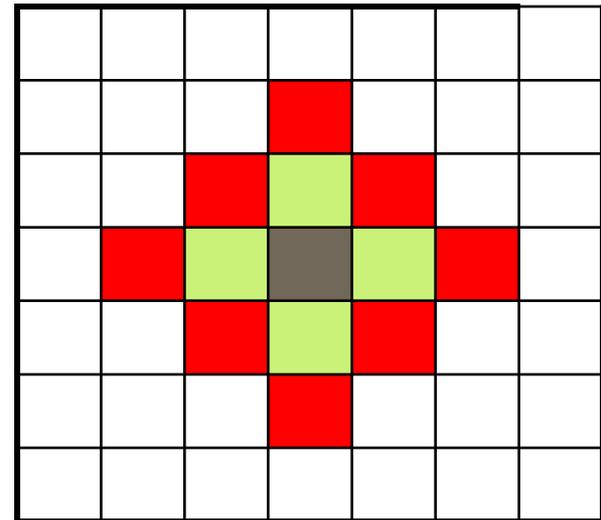
- Defini-se como **cluster** um conjunto de células vizinhas, que estejam no mesmo estado.
- Um **cluster percolante** é um cluster que atravessa toda a rede.

# 3. Vizinhança

- *Vizinhança de Neumann*



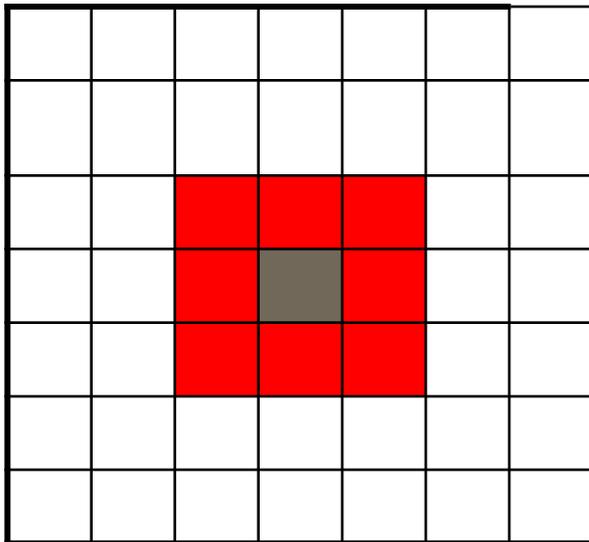
Primeiros vizinhos



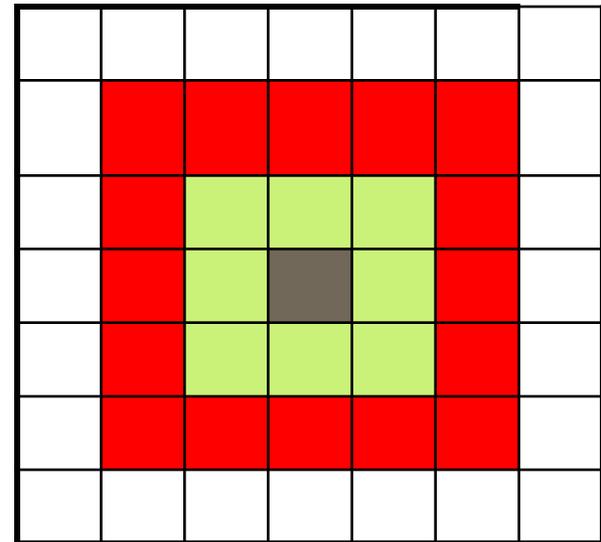
Segundos vizinhos

# 3. Vizinhança

- *Vizinhança de Moore*



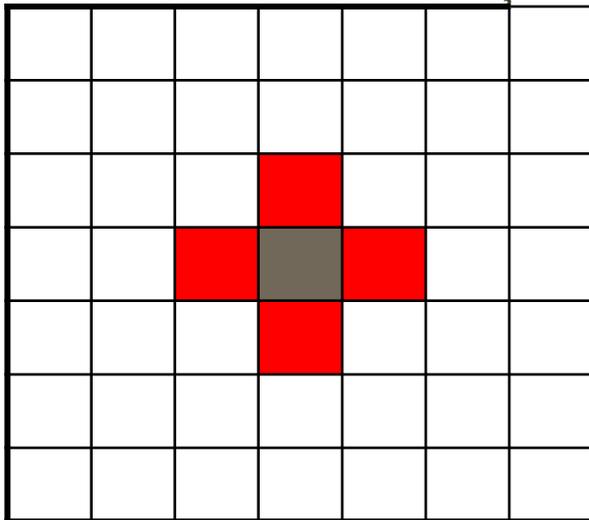
Primeiros vizinhos



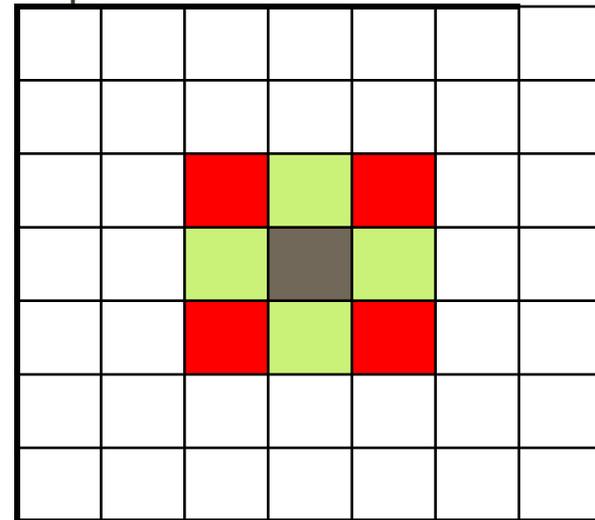
Segundos vizinhos

# 3. Vizinhança

- **Vizinhança arbitrária**
  - A vizinhança é determinada pelo modelo.



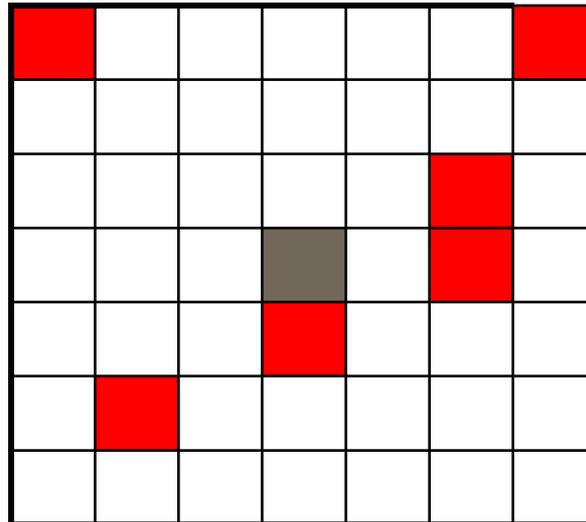
Primeiros vizinhos



Segundos vizinhos

# 3. Vizinhança

- o *Vizinhança aleatória*



## 4. Estados das células

- Os estados das células tem que ser bem **definidos** e **finitos**.
- Exemplos:**
  - 0 ou 1 – ocupado ou vazio, vivo ou morto, *spin up* ou *spin down*, etc...
  - 0, 1 ou 2 – referente a espécie que ocupa cada célula, número de indivíduos.

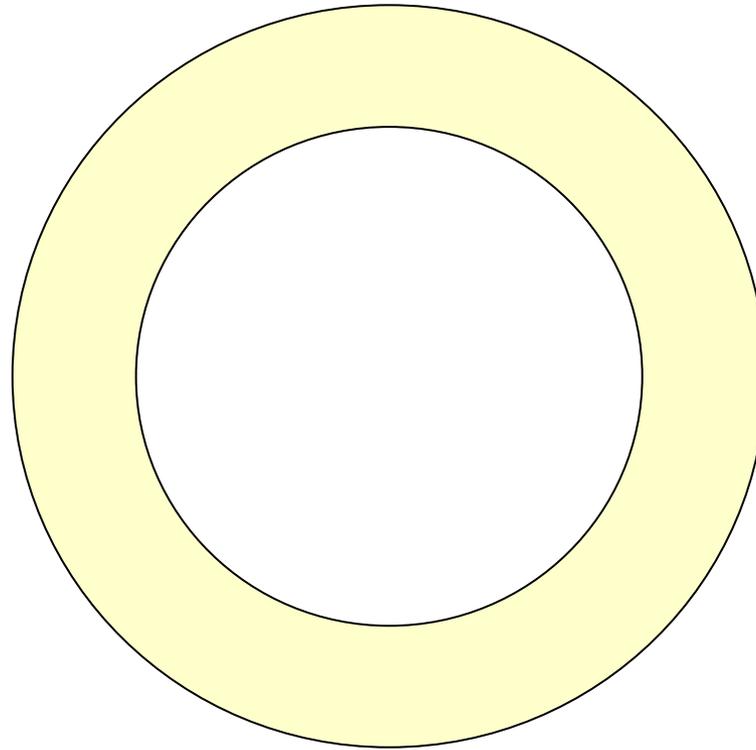
## 5. Condições de contorno

- Quando se utiliza uma rede para simular um dado sistema (discretização do espaço) é necessário adotar um critério a respeito de como serão consideradas as fronteiras (ou bordas) do sistema.

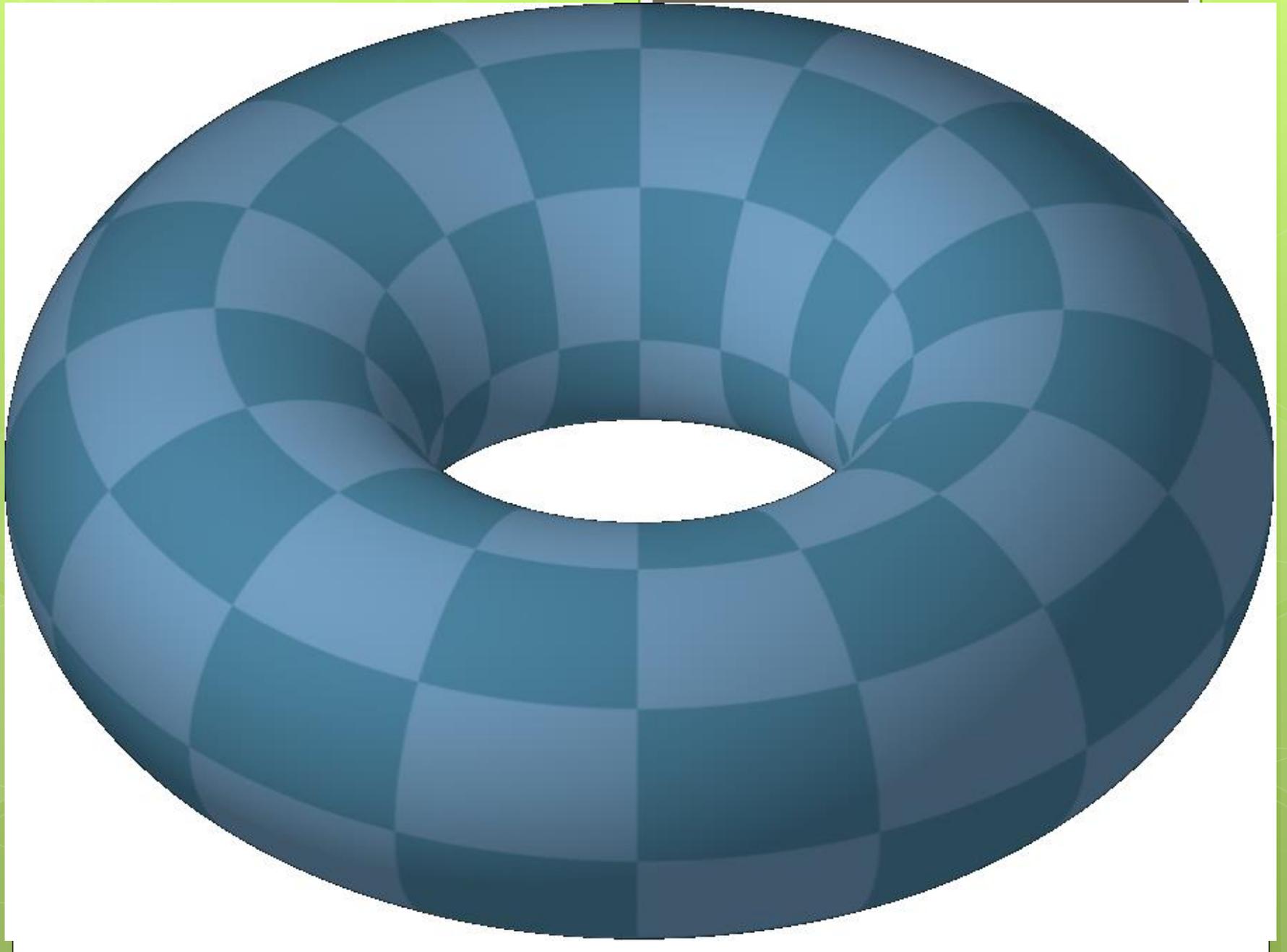
## 5. Condições de contorno

- As condições mais comuns são:
  - Condições fechadas (sistema fechado);
  - Condições abertas (sistema aberto);
  - Condições periódicas de contorno (1D – anel ou 2D – toro ou toroíde)

# 5. Condições de contorno



1D – anel



2D – toro ou toróide

## 6. Regras de atualização

- As regras podem depender do estado do próprio sítio e dos sítios vizinhos.
- Pode-se criar campos e influência da geografia do sistema.
- As regras podem ser **determinísticas** ou **probabilísticas**.

## 6. Regras de atualização

- **Processo Markoviano**

- Depende apenas do estado atual dos sítios (memória de curto alcance, também chamada de memória de 1<sup>ª</sup> ordem).

## 7. Agentes

- “Agentes autônomos são sistemas computacionais que habitam algum ambiente de dinâmica complexa, sentem e agem autonomamente neste ambiente e, fazendo isso, realizam uma série de objetivos ou tarefas para os quais foram designados” (MAES, 1995).

# 7. Agentes

- **Autonomia:**

- Agentes operam sem intervenção direta de humanos ou outros, e tem algum controle sobre suas ações e estados internos.

- **Habilidade social:**

- Interação entre agentes.

# 7. Agentes

- **Reatividade:**

- Agentes percebem seu ambiente e respondem as mudanças que ocorrem nele.

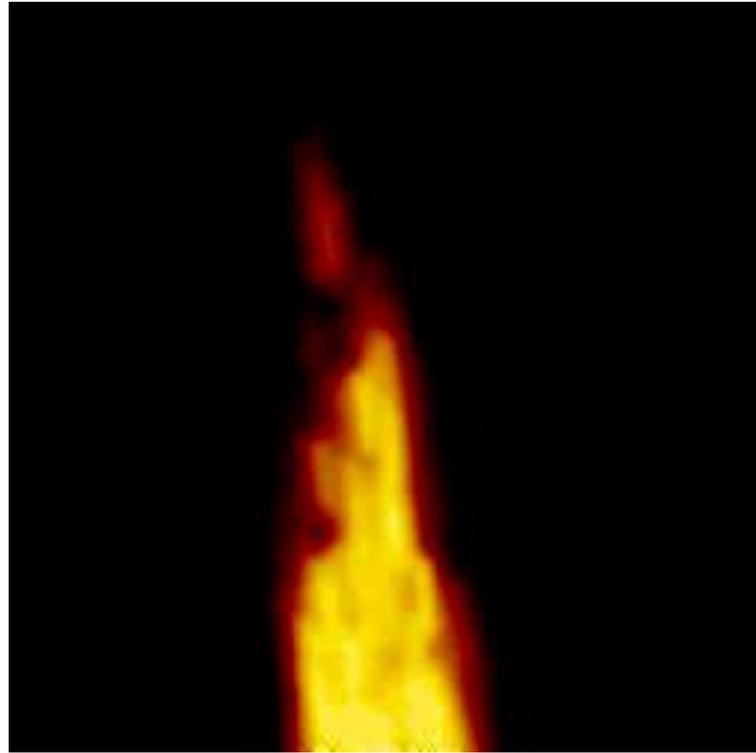
- **Pró-atividade:**

- Agentes não agem simplesmente em resposta ao seu ambiente, são capazes de tomar uma iniciativa para atingir determinado objetivo.



**CETM**  
Curso de Especialização em  
Tratamento de Minérios

## 8. AC gráficos

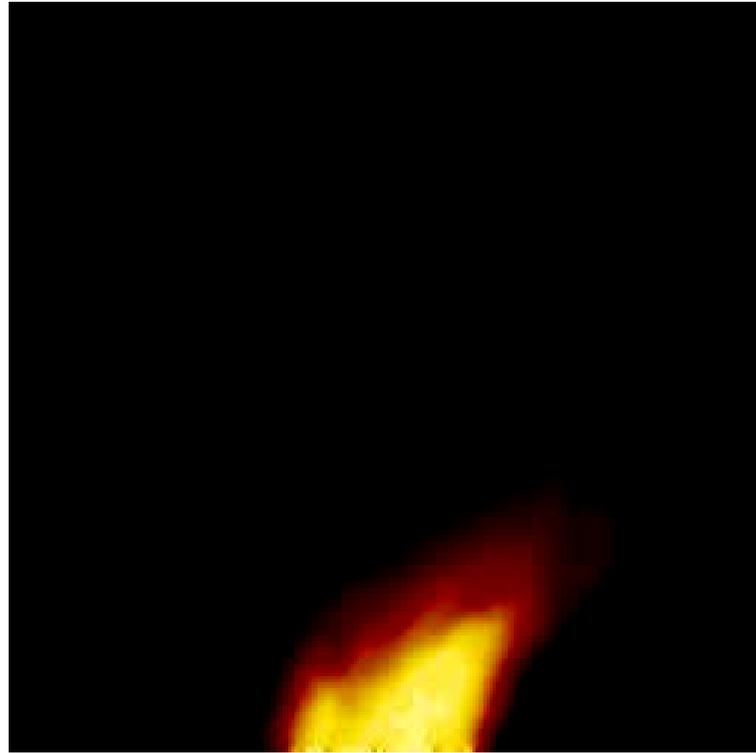


Fogo



**CETM**  
Curso de Especialização em  
Tratamento de Minérios

## 8. AC gráficos



Fogo com vento



## 8. AC gráficos



Queda d'água

## 8.1. Algoritmo de *Burning* ou Fogo na floresta

- Faz uso de uma **rede quadrada**;
- Os estados possíveis são:
  - Branco – sítio vazio;
  - Verde – sítio ocupado por uma árvore;
  - Vermelho – sítio ocupado por árvore queimando;
  - Preto – sítio ocupador por uma árvore queimada.

# 8.1. Algoritmo de *Burning* ou Fogo na floresta

## ○ **Condição inicial**

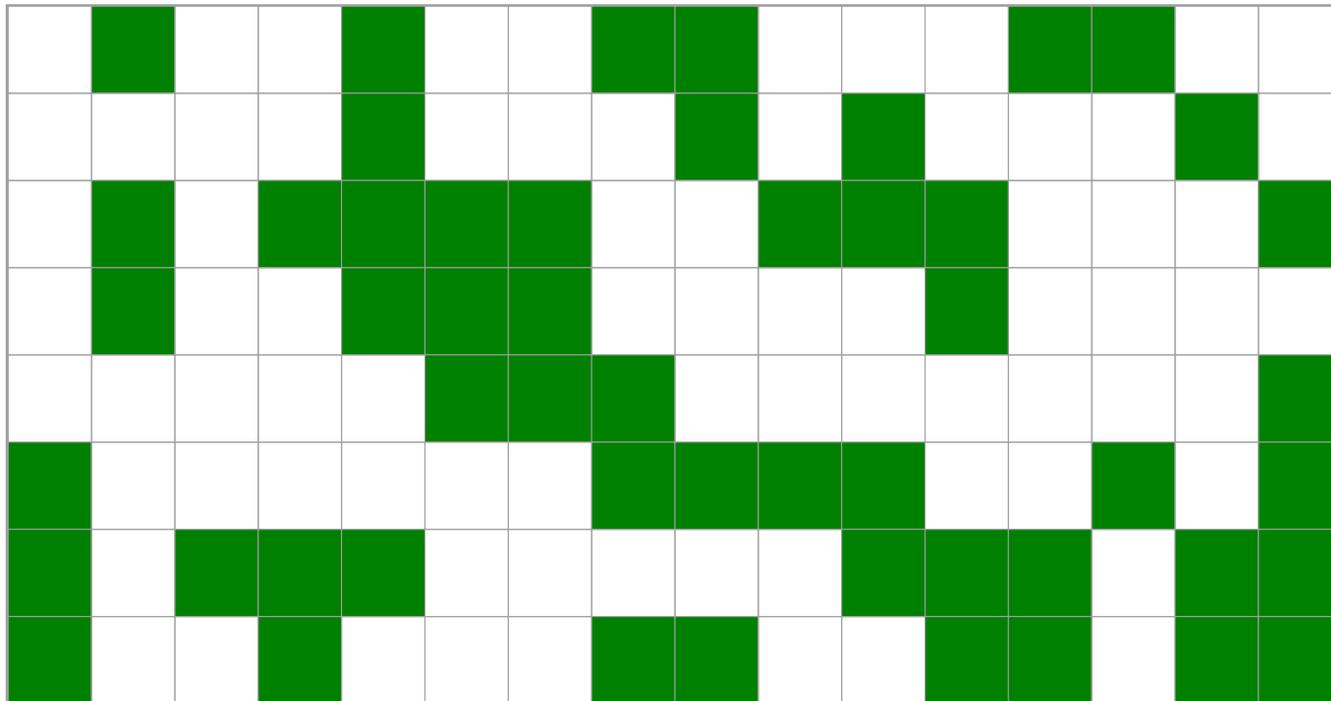
- Cada sítio tem a probabilidade  $p$  de estar ocupado (**verde**);
- Escolhe-se arbitrariamente os focos de incêndio (**vermelho**).

## ○ **Dinâmica**

- Os sítios **verdes** que possuírem um sítio **vermelho** na vizinhança queimarão no passo seguinte;
- Os sítios **vermelhos** ficarão pretos no passo seguinte.

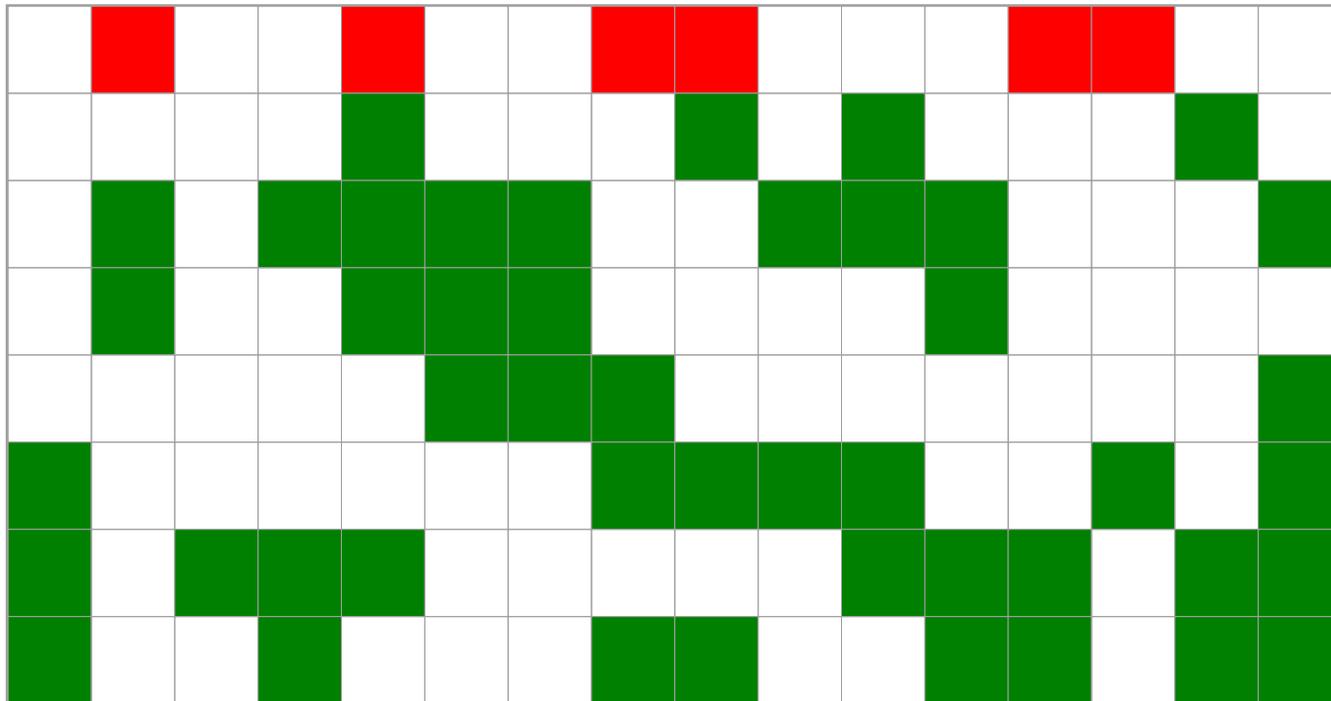


# 8.1. Algoritmo de *Burning* ou Fogo na floresta



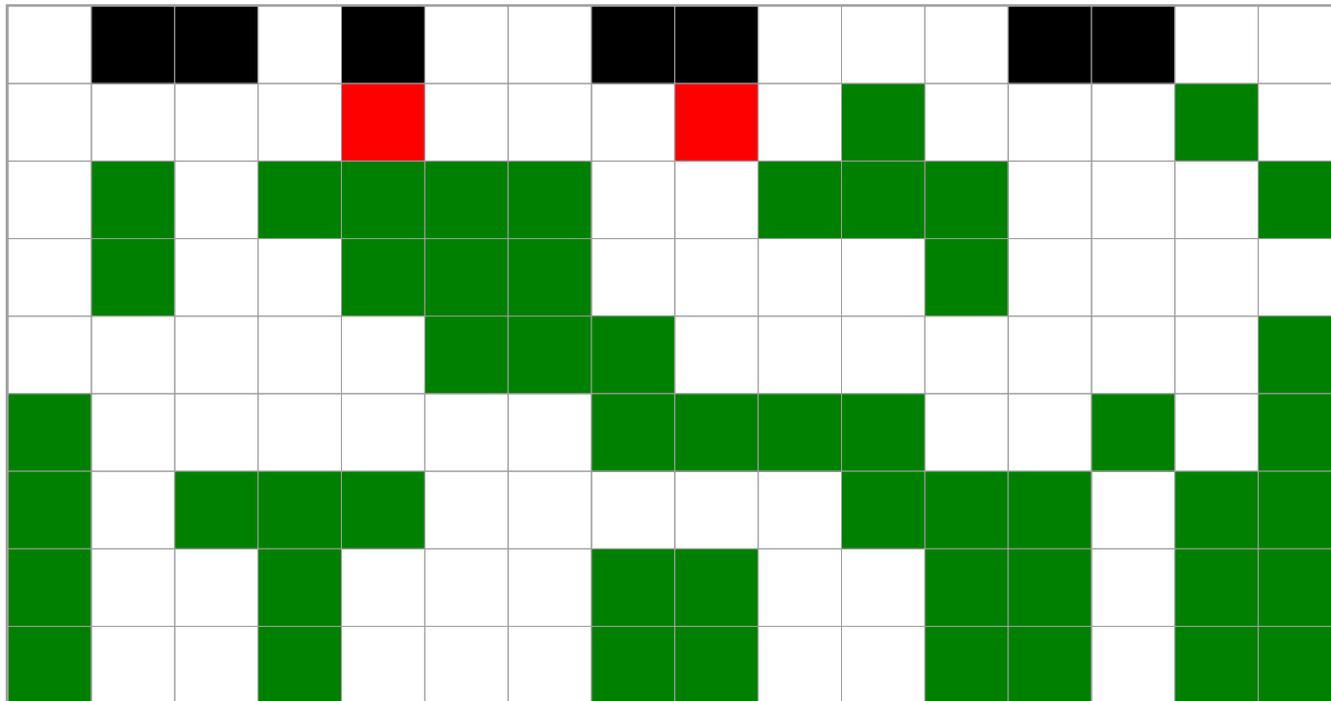


# 8.1. Algoritmo de *Burning* ou Fogo na floresta



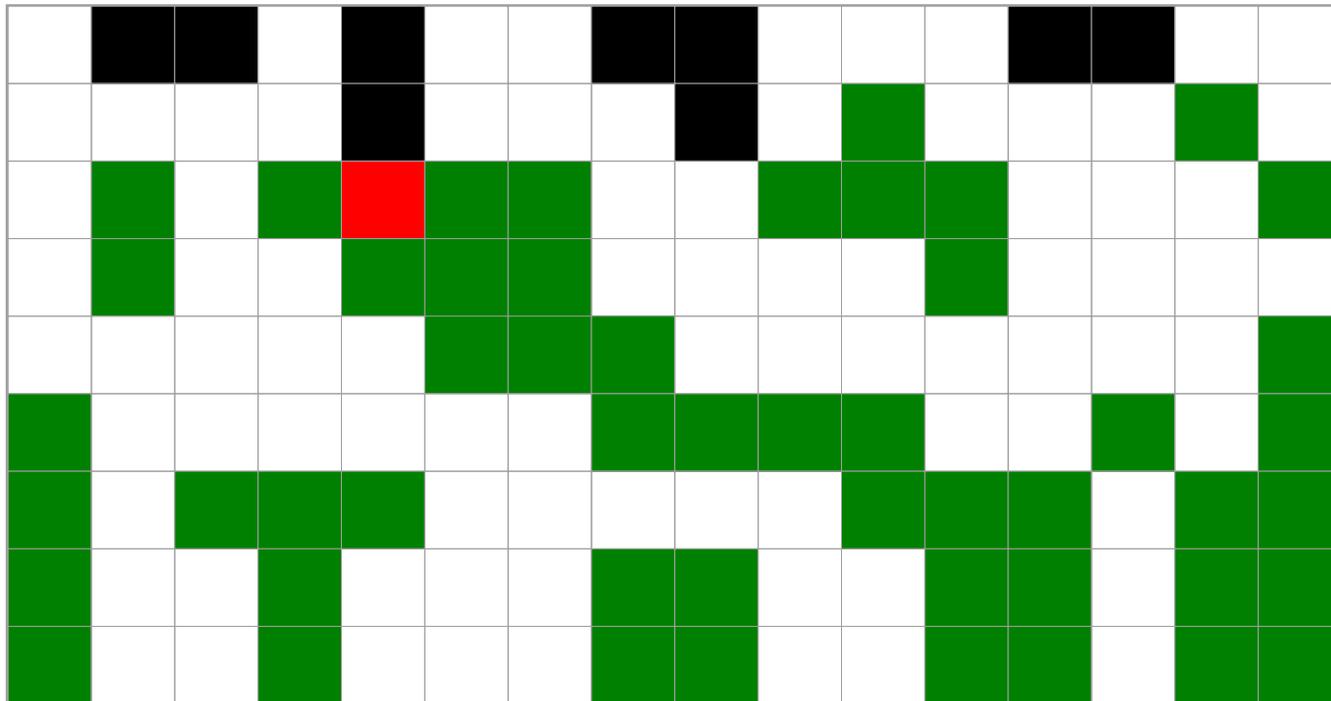


# 8.1. Algoritmo de *Burning* ou Fogo na floresta



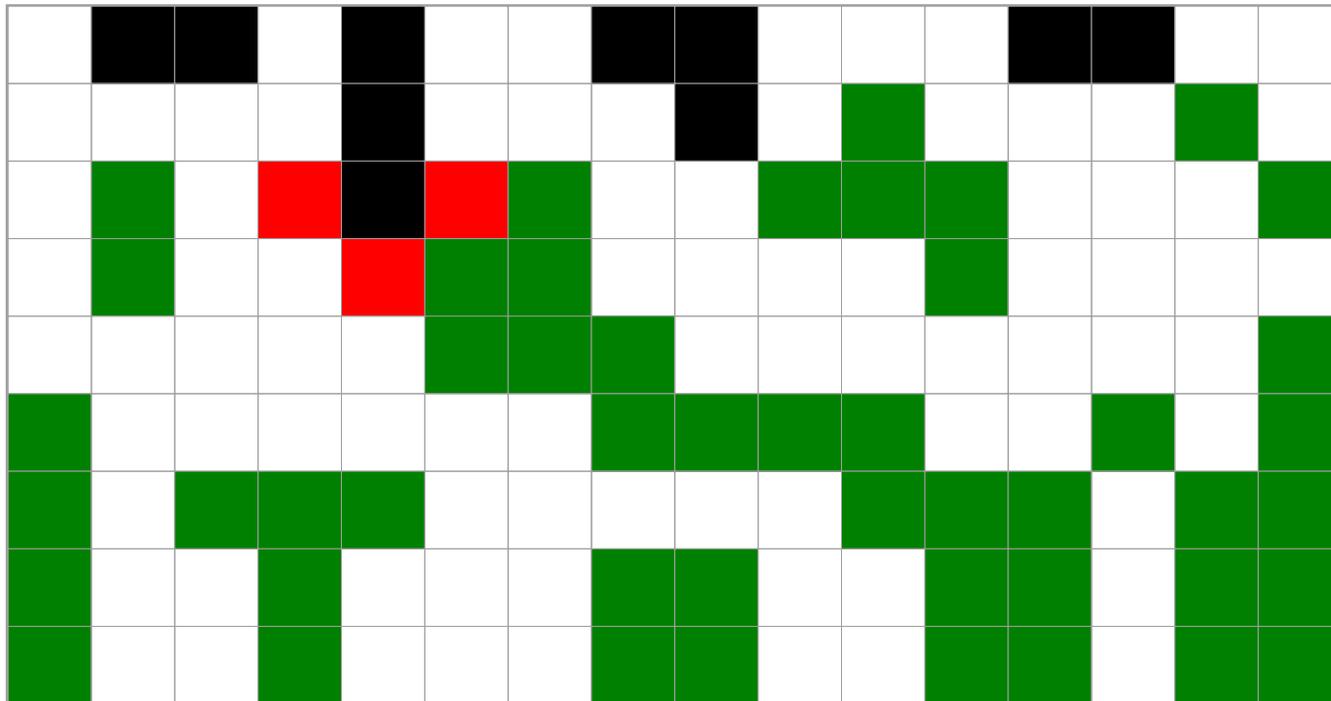


# 8.1. Algoritmo de *Burning* ou Fogo na floresta



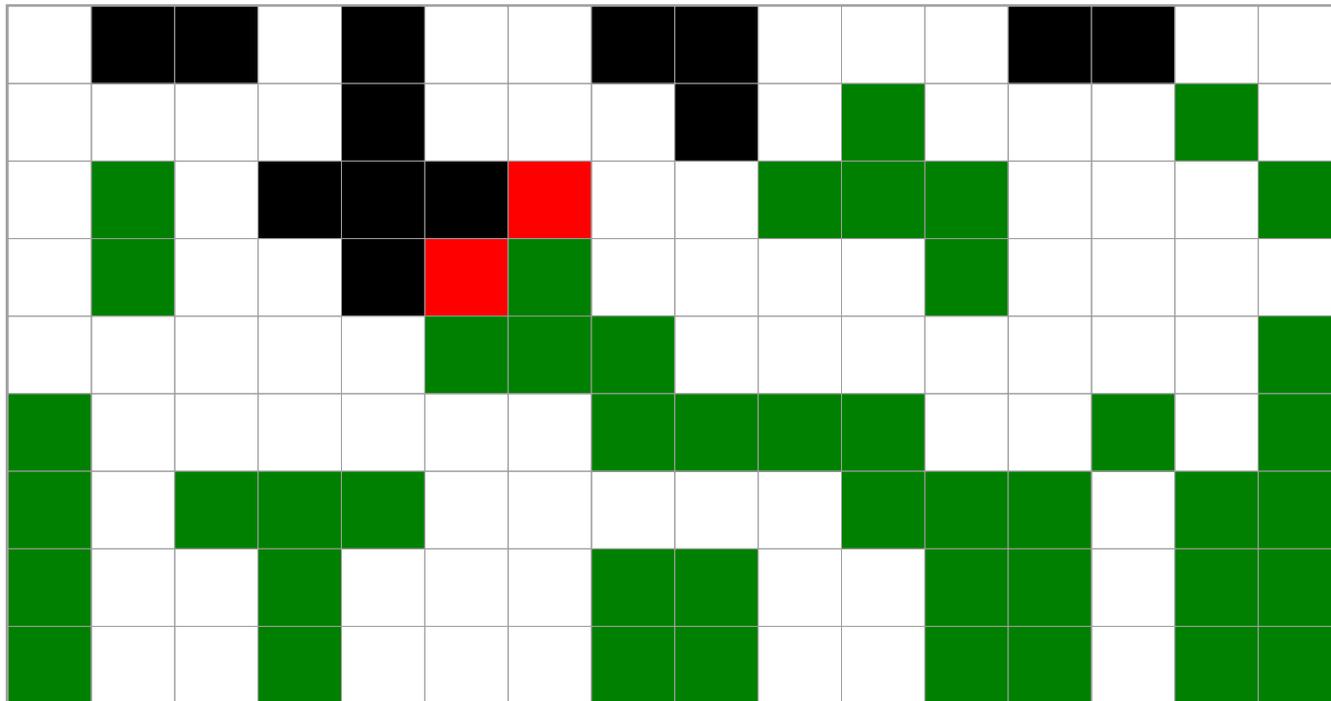


# 8.1. Algoritmo de *Burning* ou Fogo na floresta



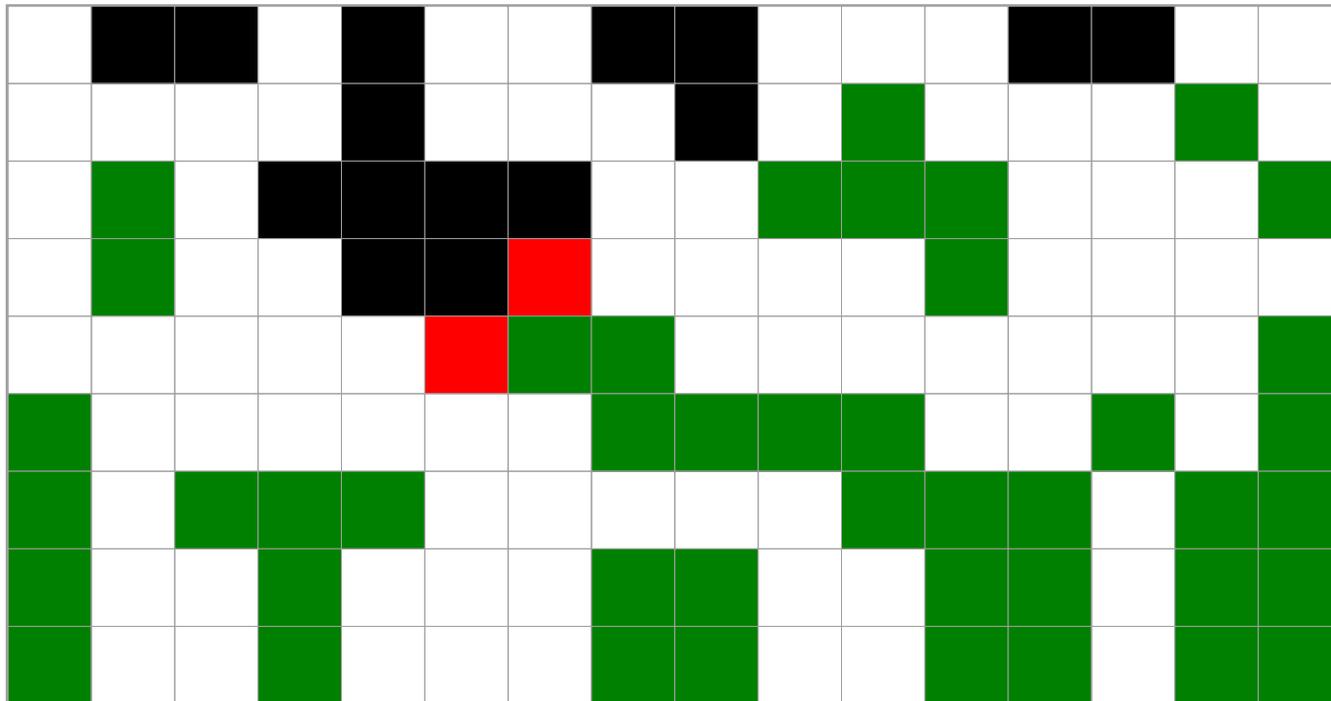


# 8.1. Algoritmo de *Burning* ou Fogo na floresta



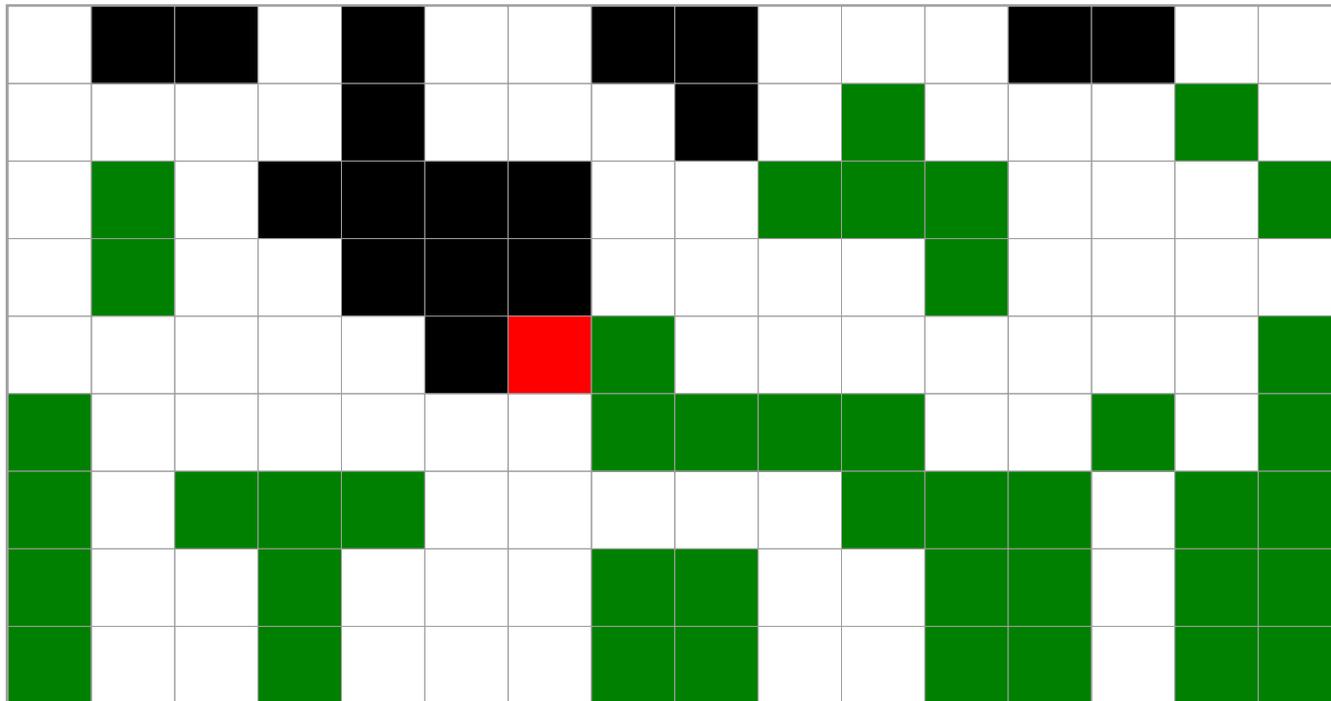


# 8.1. Algoritmo de *Burning* ou Fogo na floresta



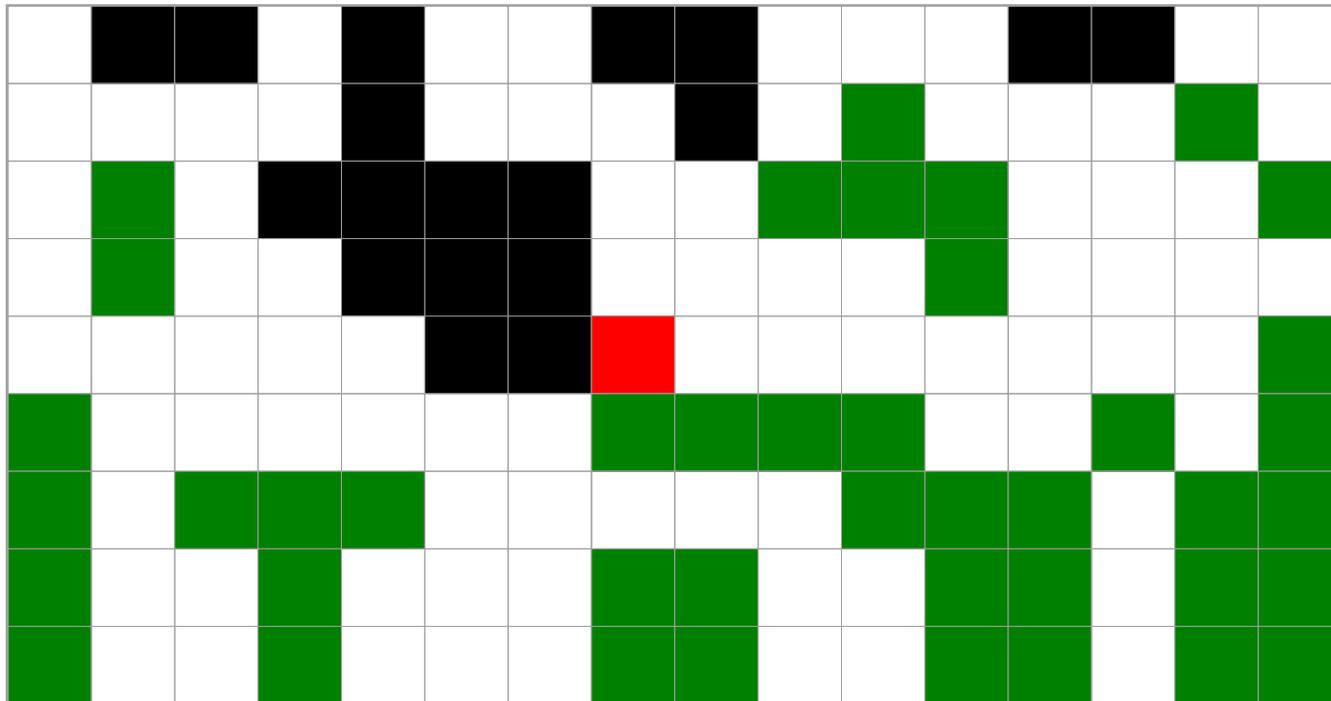


# 8.1. Algoritmo de *Burning* ou Fogo na floresta



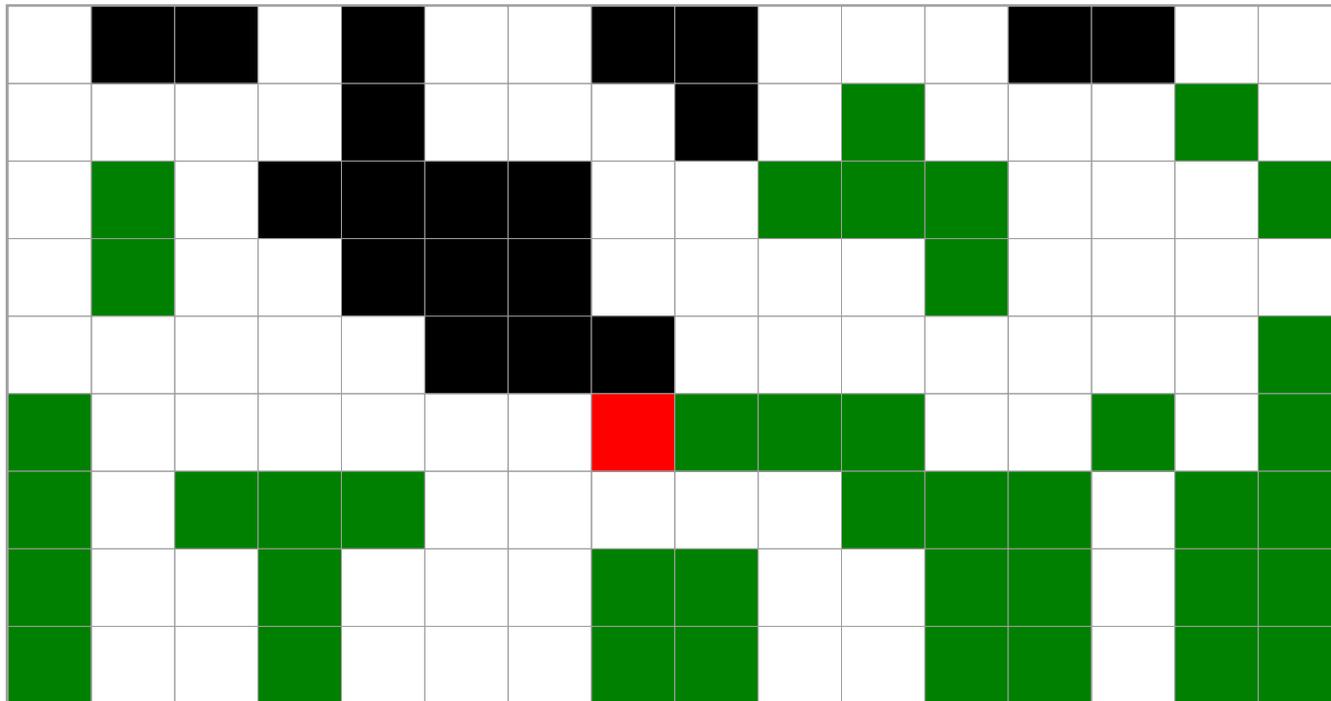


# 8.1. Algoritmo de *Burning* ou Fogo na floresta



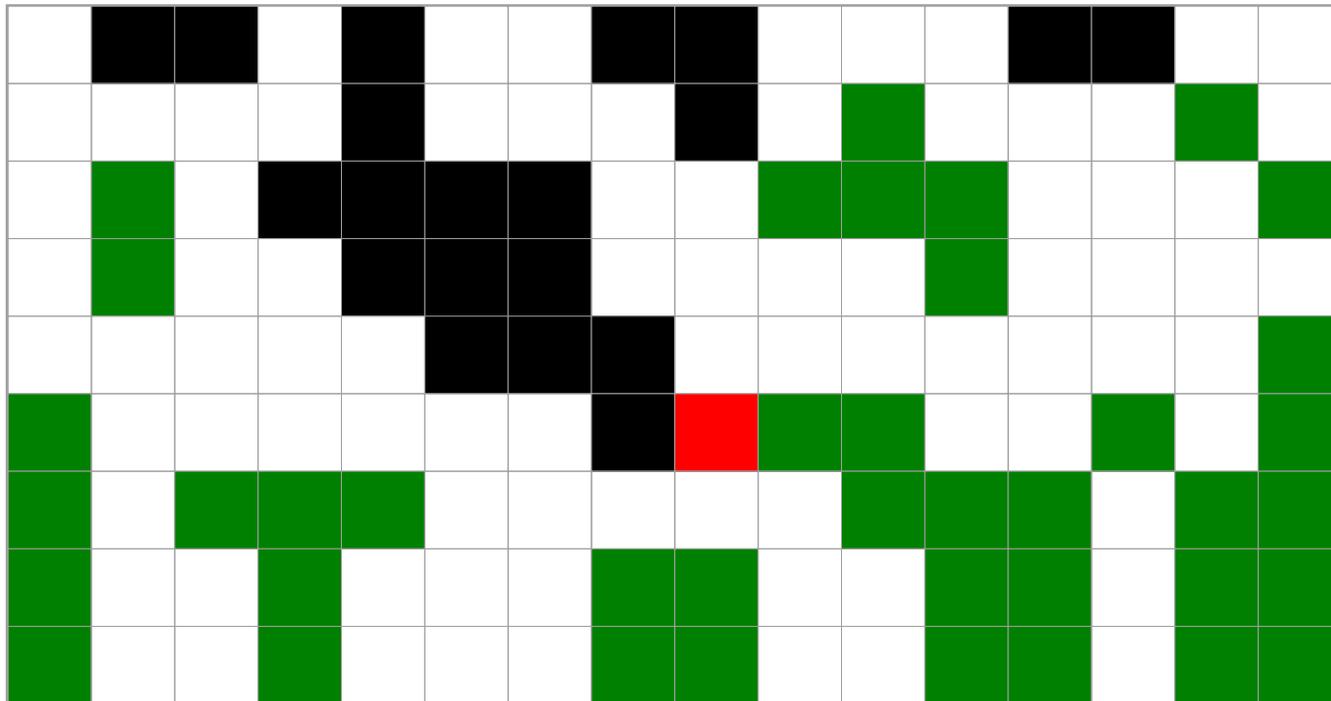


# 8.1. Algoritmo de *Burning* ou Fogo na floresta





# 8.1. Algoritmo de *Burning* ou Fogo na floresta

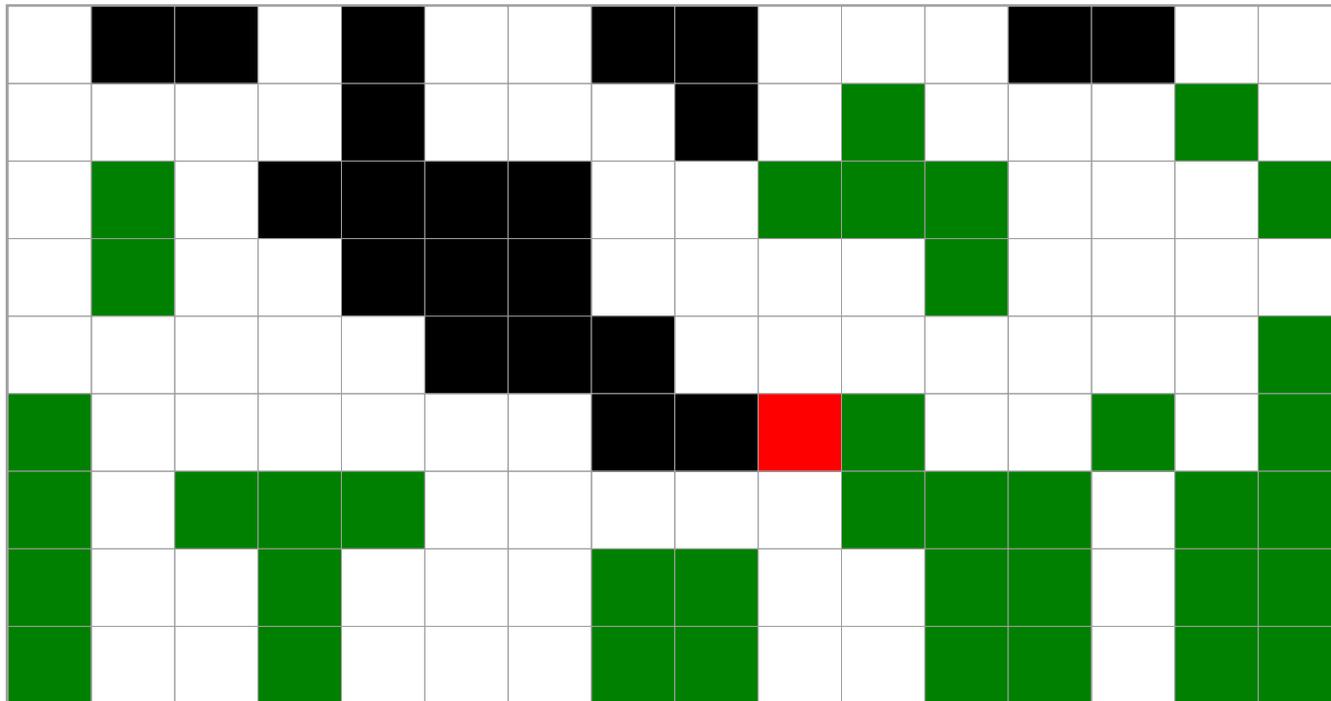




**CETM**

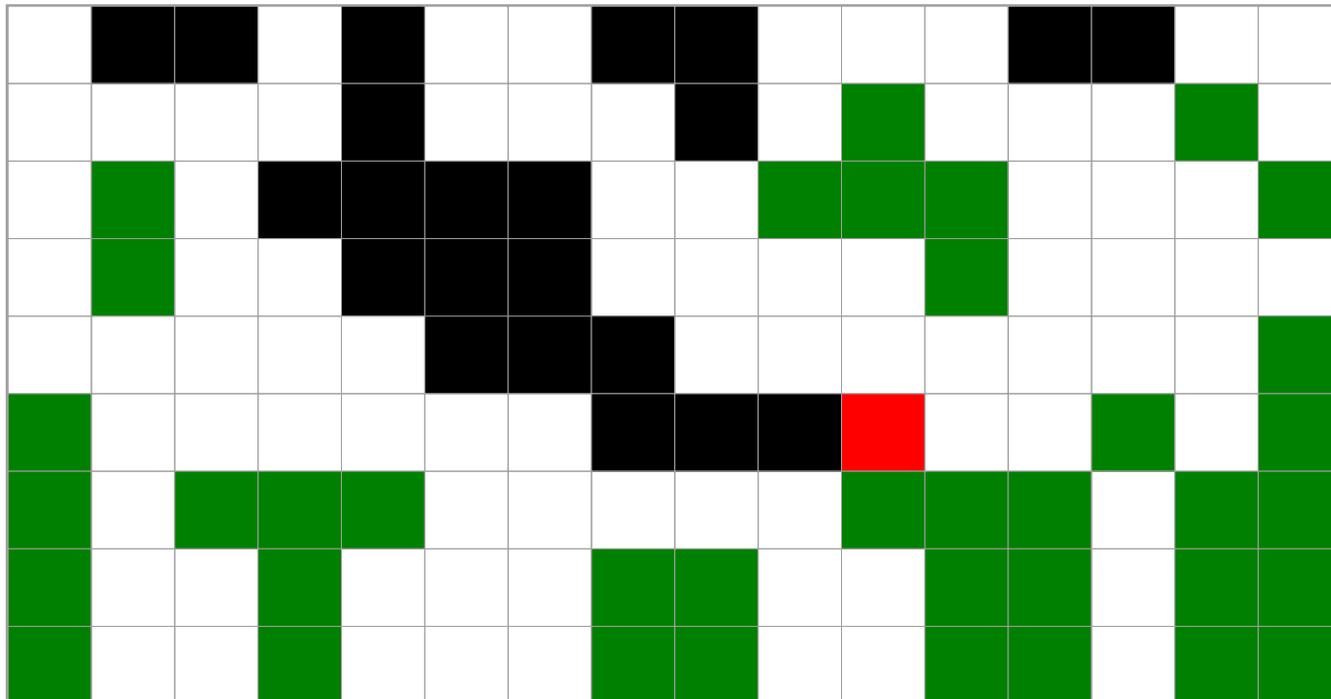
Curso de Especialização em  
Tratamento de Minérios

# 8.1. Algoritmo de *Burning* ou Fogo na floresta



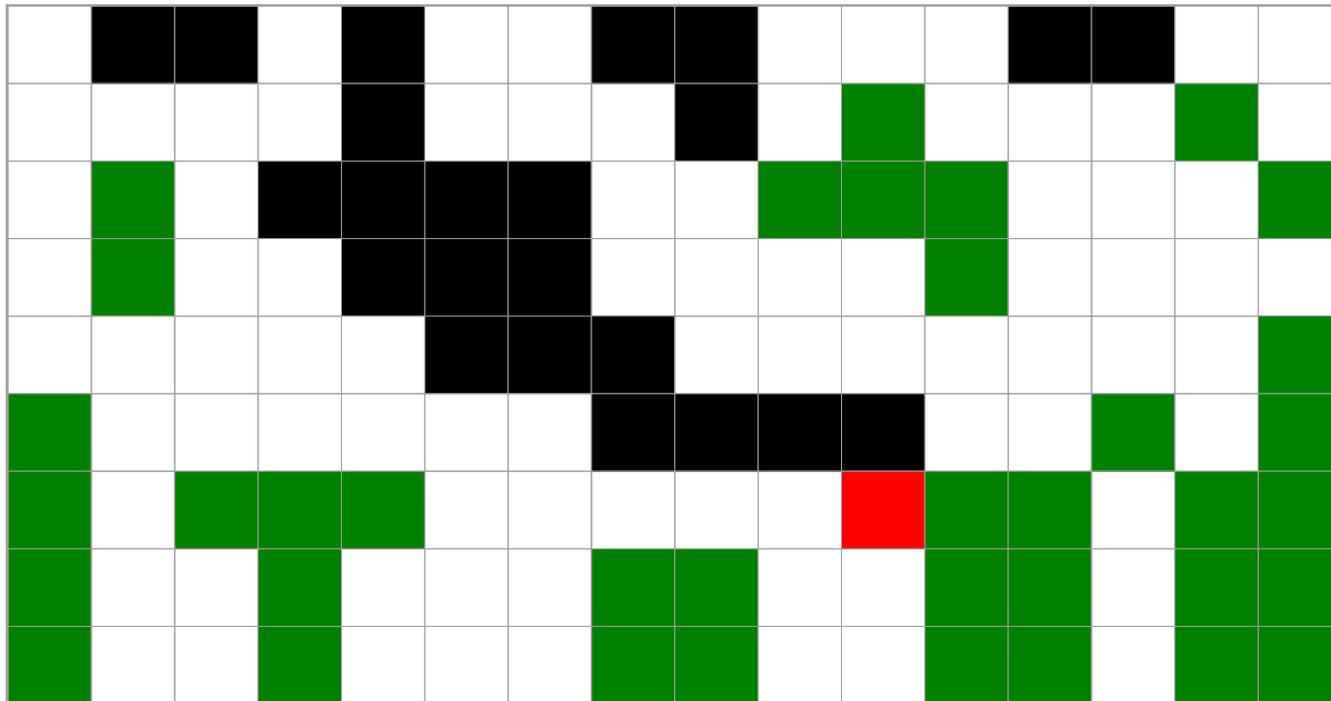


# 8.1. Algoritmo de *Burning* ou Fogo na floresta



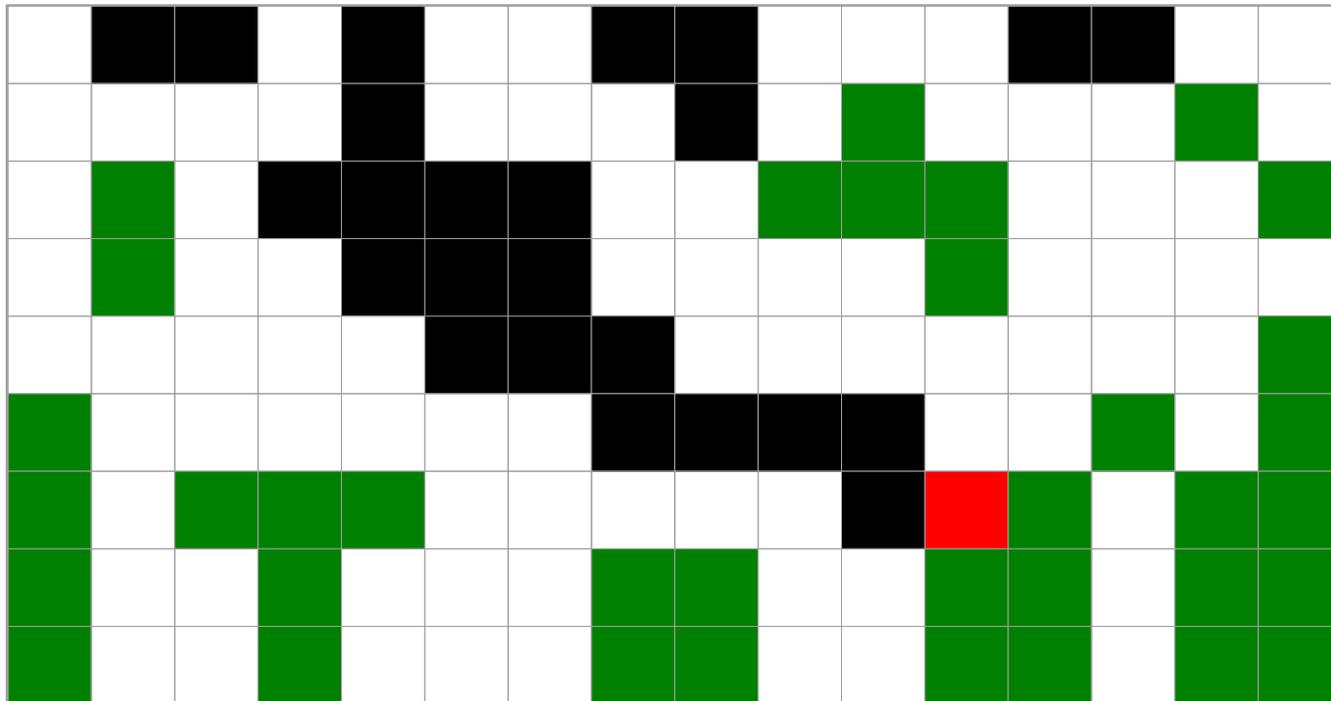


# 8.1. Algoritmo de *Burning* ou Fogo na floresta



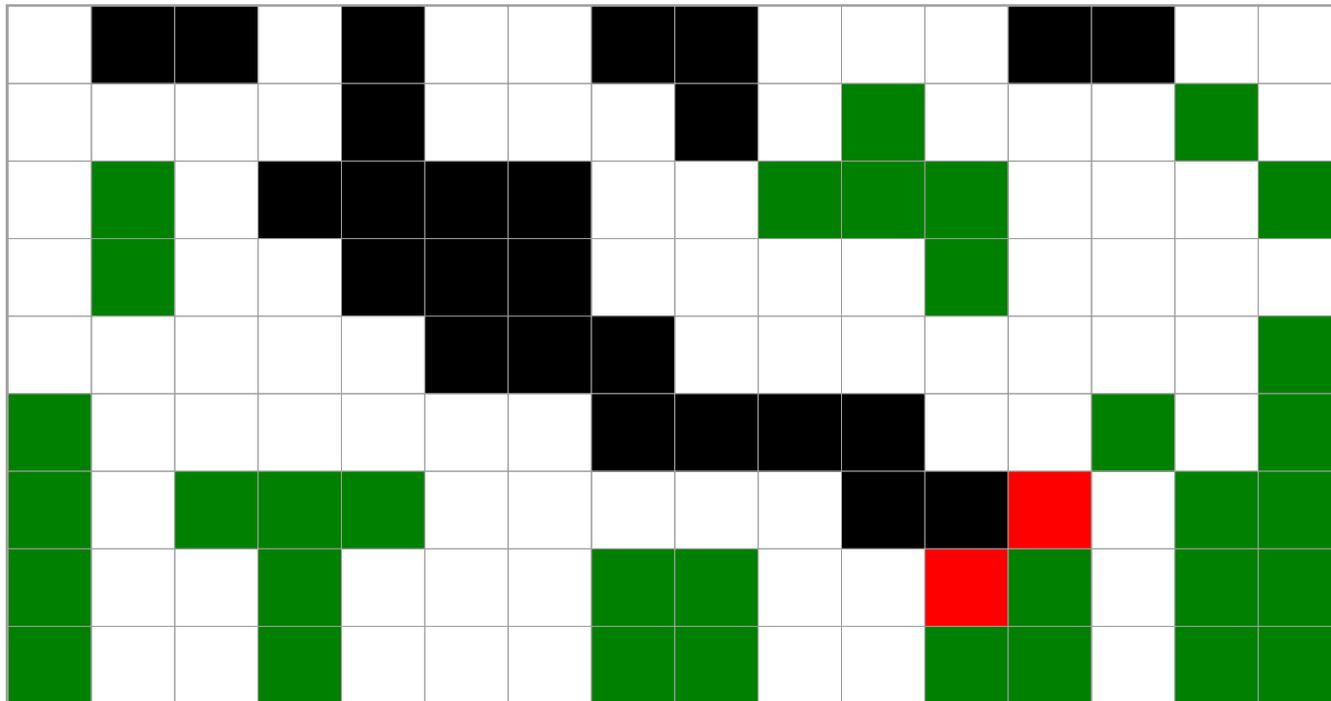


# 8.1. Algoritmo de *Burning* ou Fogo na floresta



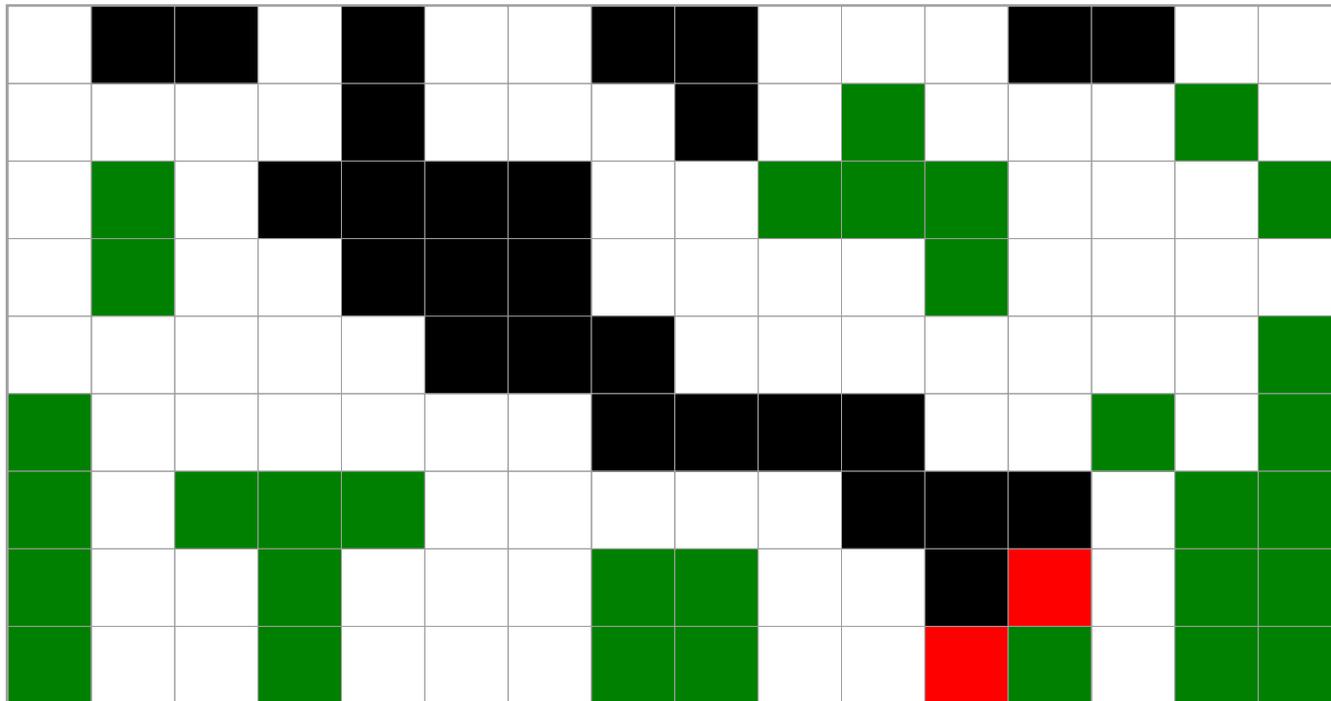


# 8.1. Algoritmo de *Burning* ou Fogo na floresta





# 8.1. Algoritmo de *Burning* ou Fogo na floresta

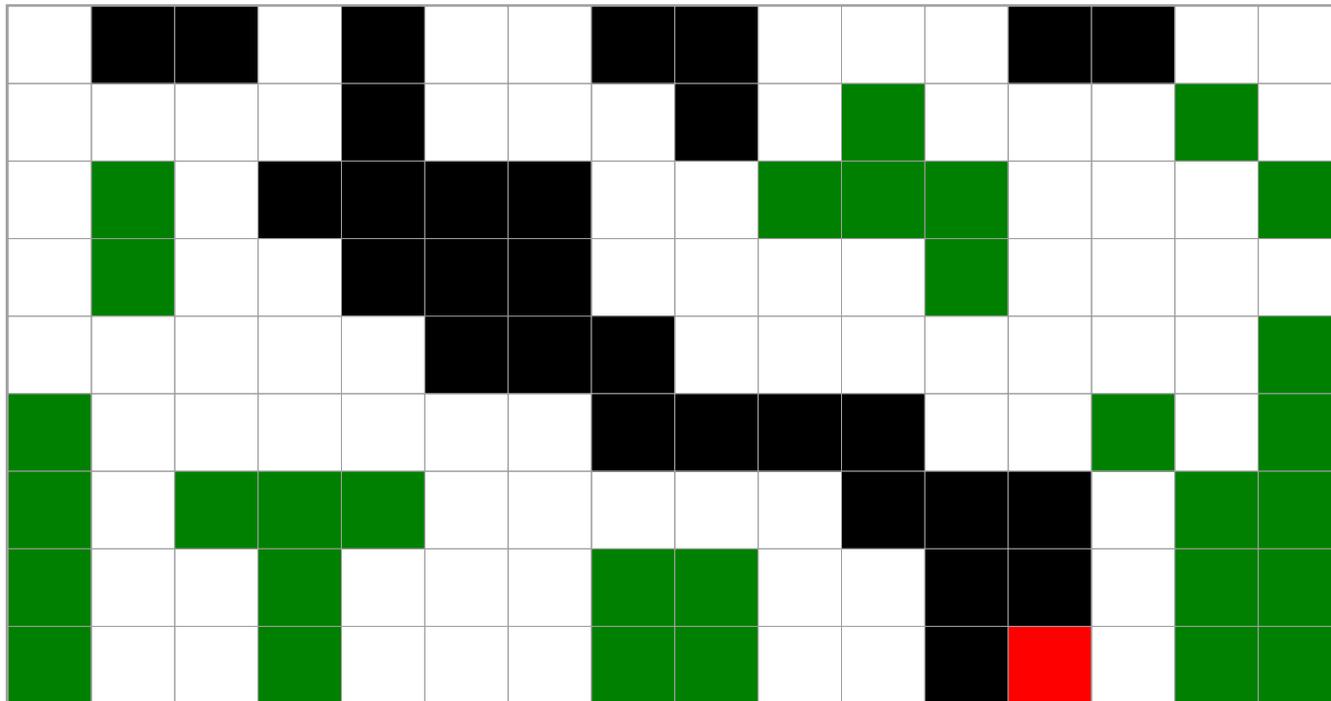




**CETM**

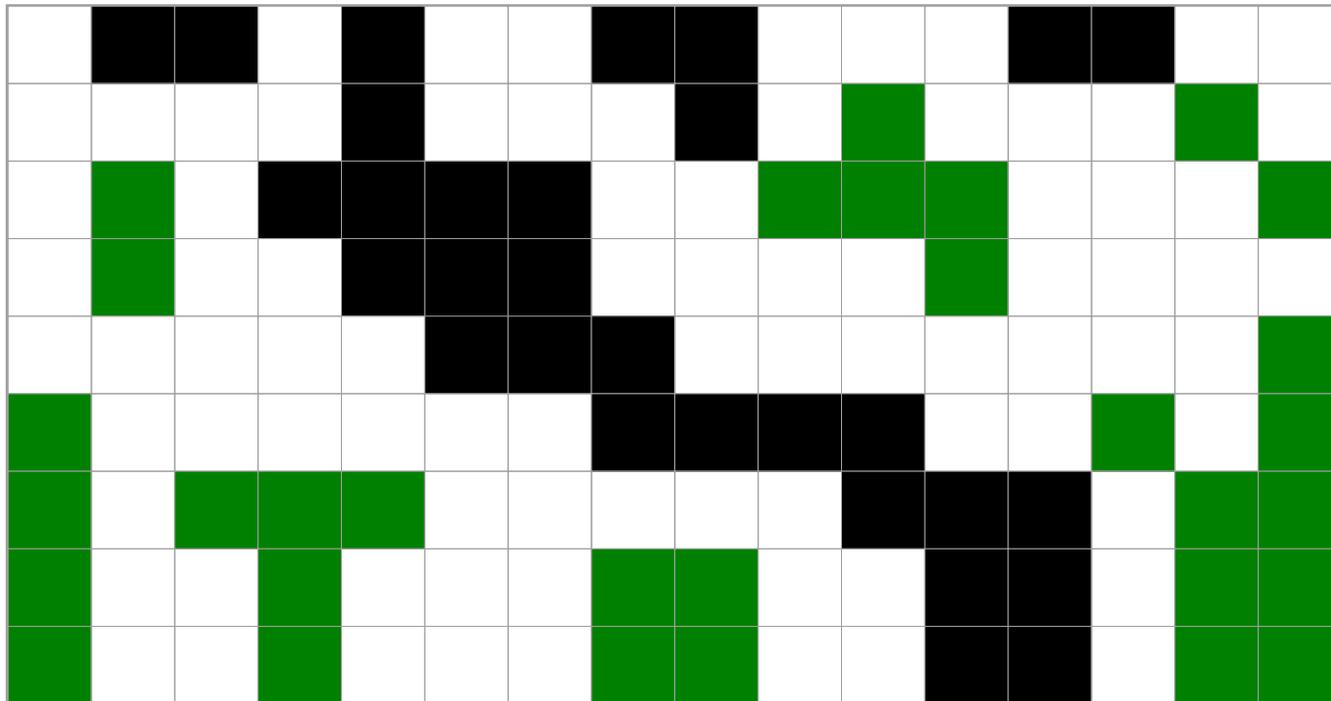
Curso de Especialização em  
Tratamento de Minérios

# 8.1. Algoritmo de *Burning* ou Fogo na floresta



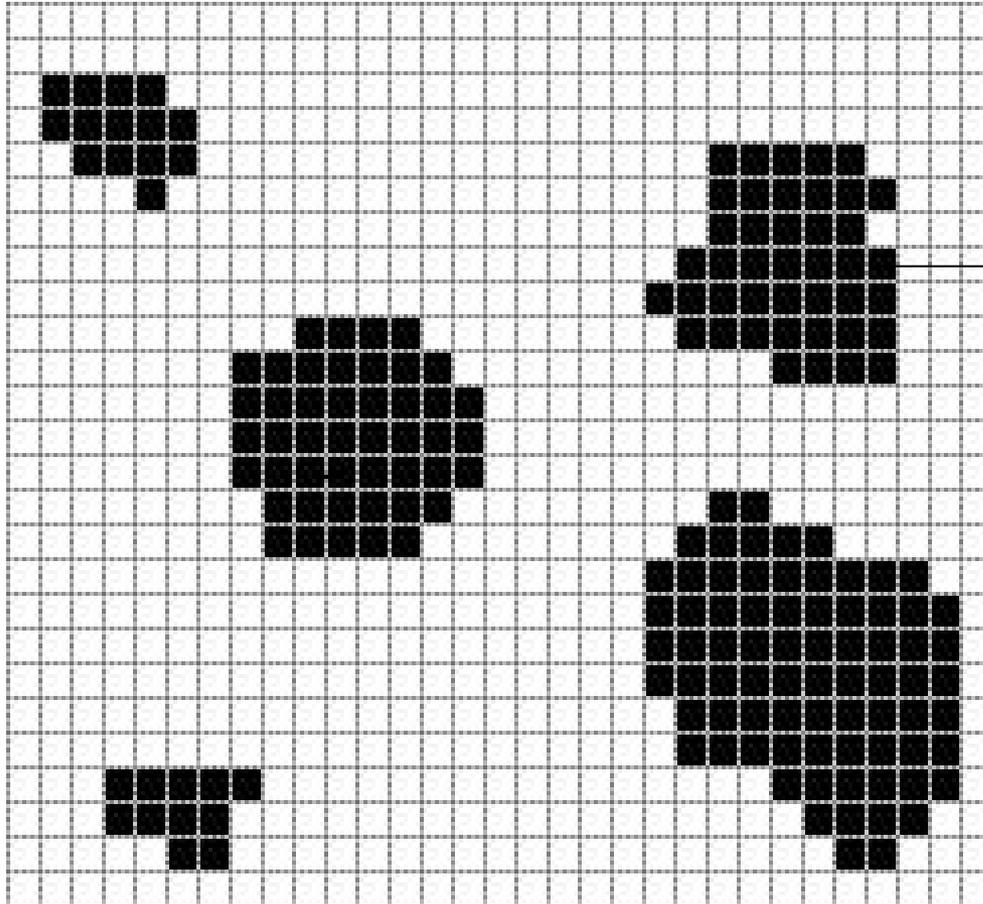


# 8.1. Algoritmo de *Burning* ou Fogo na floresta



## 8.2. Cluster

- Um cluster em uma imagem digital é um conjunto de pixels adjacentes, que se encontram “ilhados”.
- Ou seja, um conjunto de pixels que se distingue dos demais pela união de pixels com características semelhantes, tais como a sua cor.



Exemplo de um cluster  
com tamanho igual a  
42 pixels

## 8.3. Percolação

- A teoria da percolação foi introduzida por Broadbent e Hammersley, nos anos 50, como um modelo matemático de propagação em meios aleatórios.
- Nestes meios a propagação é contínua, ou não existe, e a transição entre os dois regimes é muito rápida.

## 8.3. Percolação

- Modelos utilizando a **Teoria da Percolação** aparecem nas mais diversas áreas do conhecimento, tais como:
  - Astronomia
  - Física dos materiais;
  - Biologia;
  - Economia;
  - Sociologia;
  - Geologia;
  - Dentre outros.



**CETM**  
Curso de Especialização em  
Tratamento de Minérios

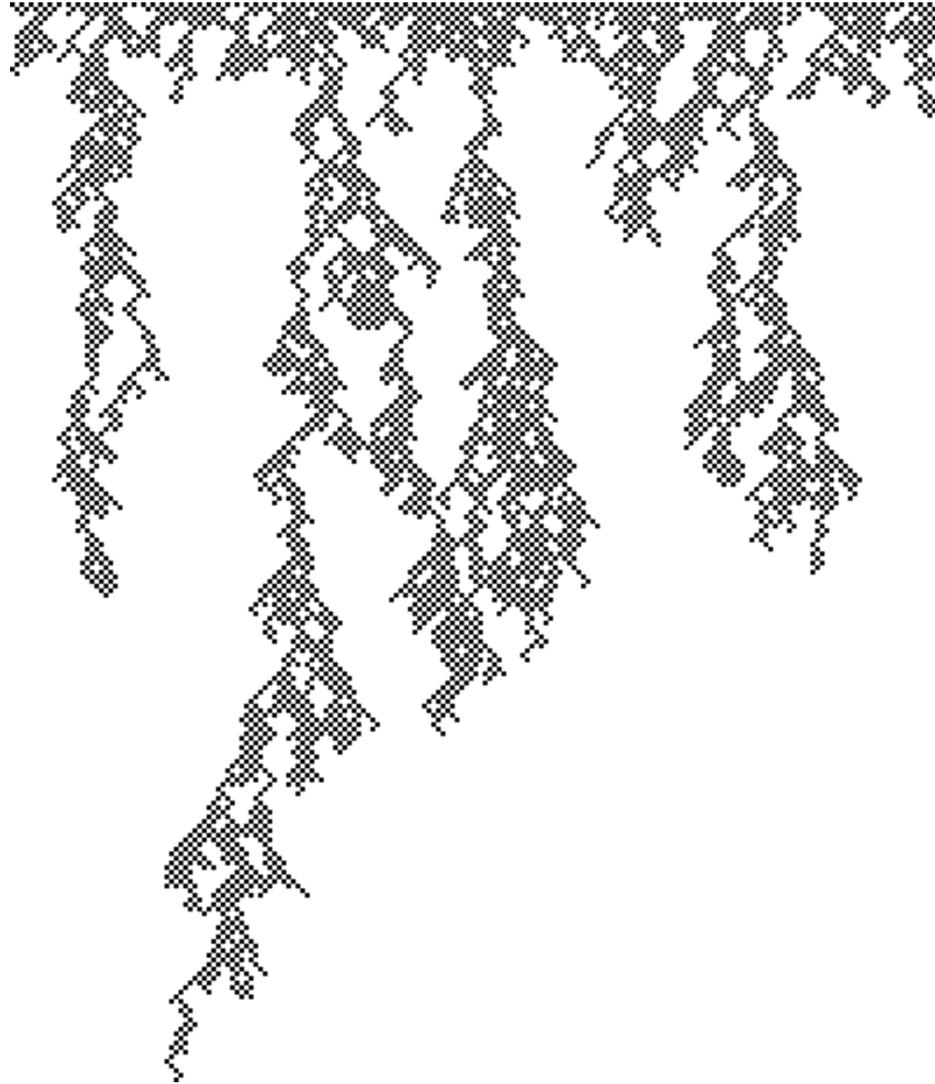




Photo F. Troya ©2007

## 8.4. Algoritmo Silva e Buonfioli

- Dada uma imagem raster qualquer e a cor desejada o algoritmo proposto percorre tal imagem até encontrar um pixel que possua cor igual à desejada e, a partir deste pixel, aplica-se a técnica de percolação.
- Para percolar entre os pixel adjacentes utilizam-se duas estruturas de dados do tipo fila **FIFO** (do inglês *First In First Out*, ou seja, o primeiro elemento a entrar na fila é o primeiro elemento a sair desta) onde os pixels a serem processados serão armazenados, bem como uma fila para armazenar os já processados.

## 8.4. Algoritmo Silva e Buonfioli

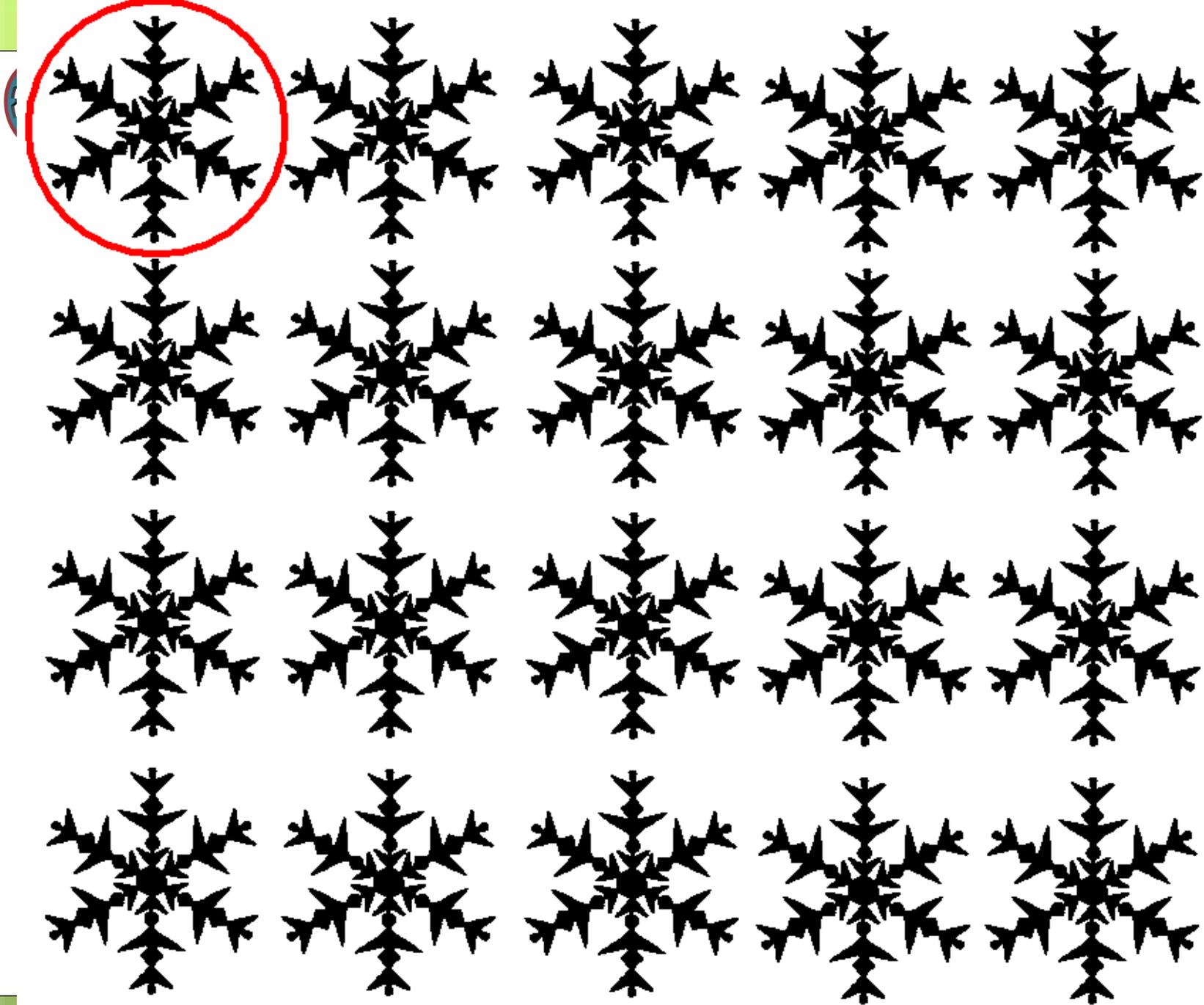
- O algoritmo consulta os pixels vizinhos do pixel atual: superior e inferior, esquerda e direita.
- Todo pixel adjacente ao pixel central que for igual à cor selecionada para análise é adicionado na fila de pixels a processar.
- A cada iteração a fila de pixels a processar é consultada para saber qual o próximo pixel a ser processado.

## 8.4. Algoritmo Silva e Buonfioli

- Quando o pixel já foi processado este é retirado da fila de pixels a processar e é inserido na fila de pixels processados.
- Repete-se tais passos até que a fila de pixels a processar esteja vazia.

## 8.4. Algoritmo Silva e Buonfioli

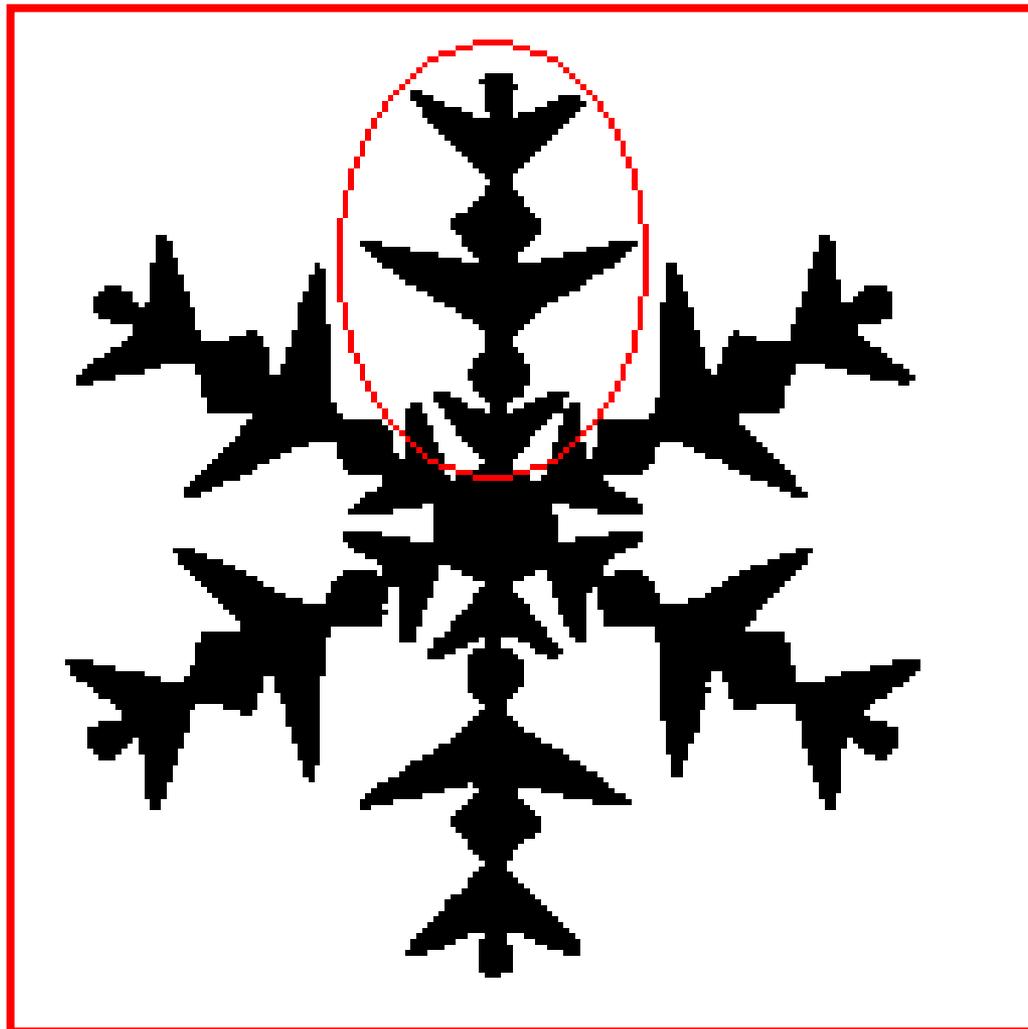
- Ao final do processo os pixels que compõem o cluster encontrado estarão armazenados na fila de pixels processados.
- Percebe-se que o algoritmo de percolação acaba por gerar um grafo.





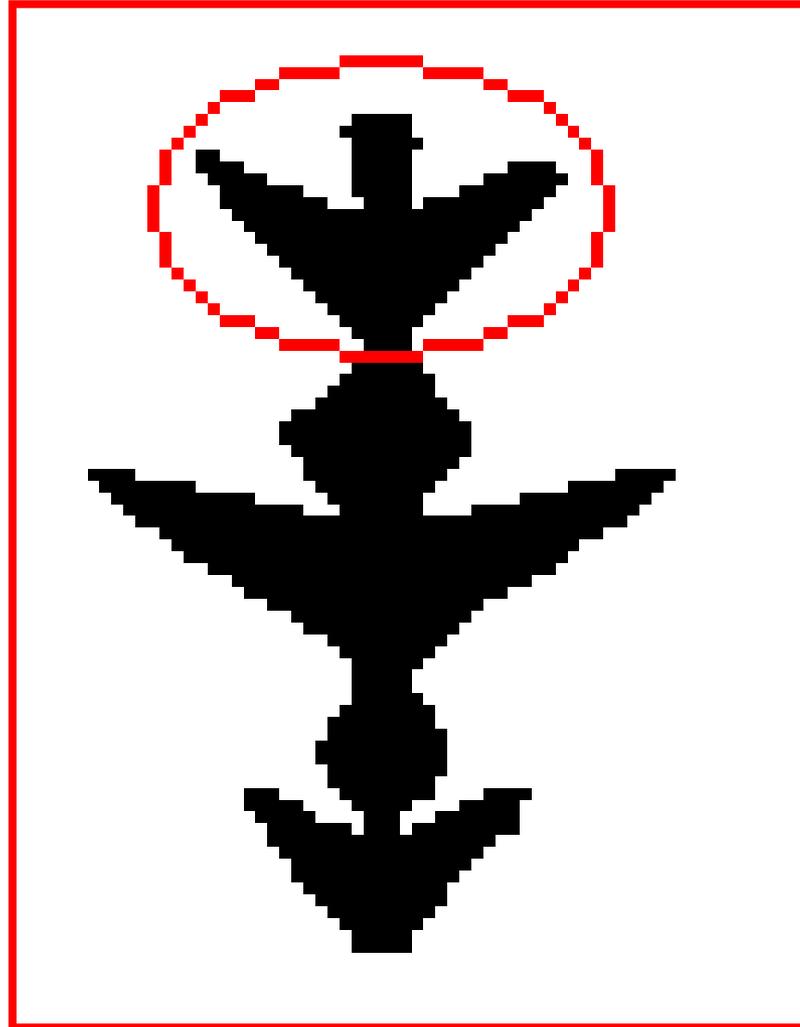
**CETM**

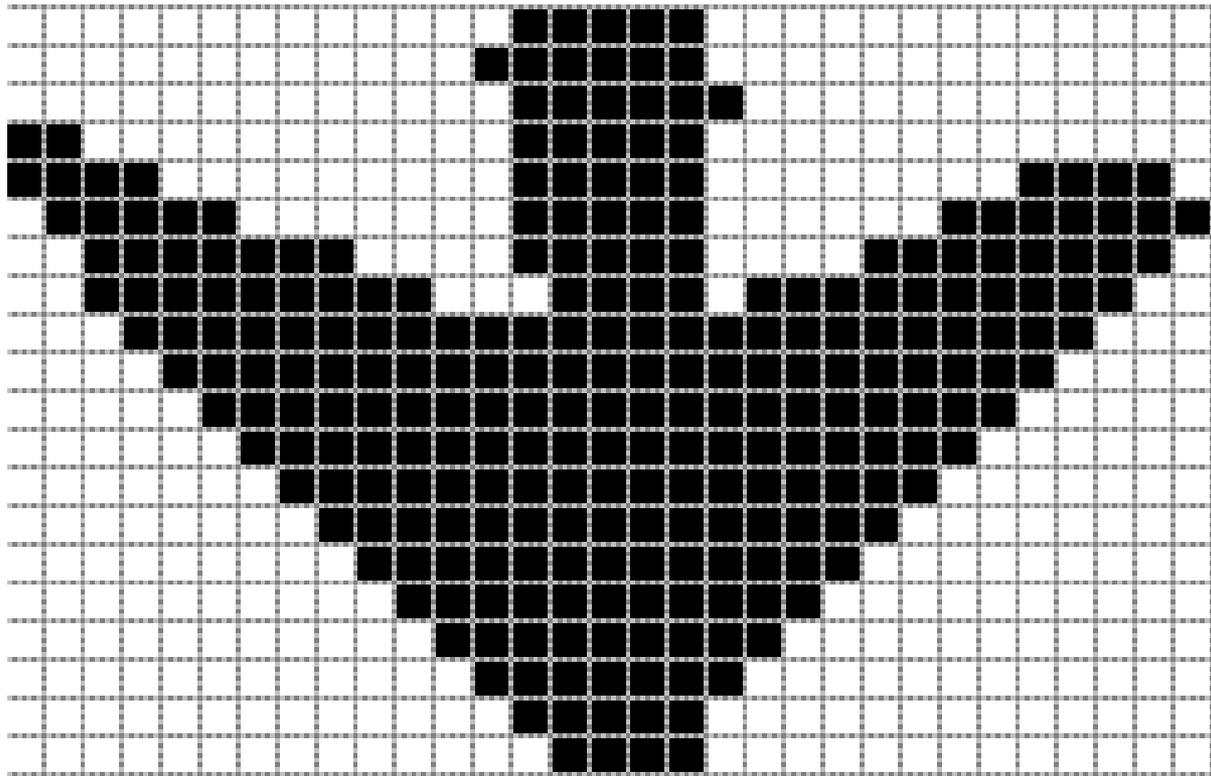
Curso de Especialização em  
Tratamento de Minérios

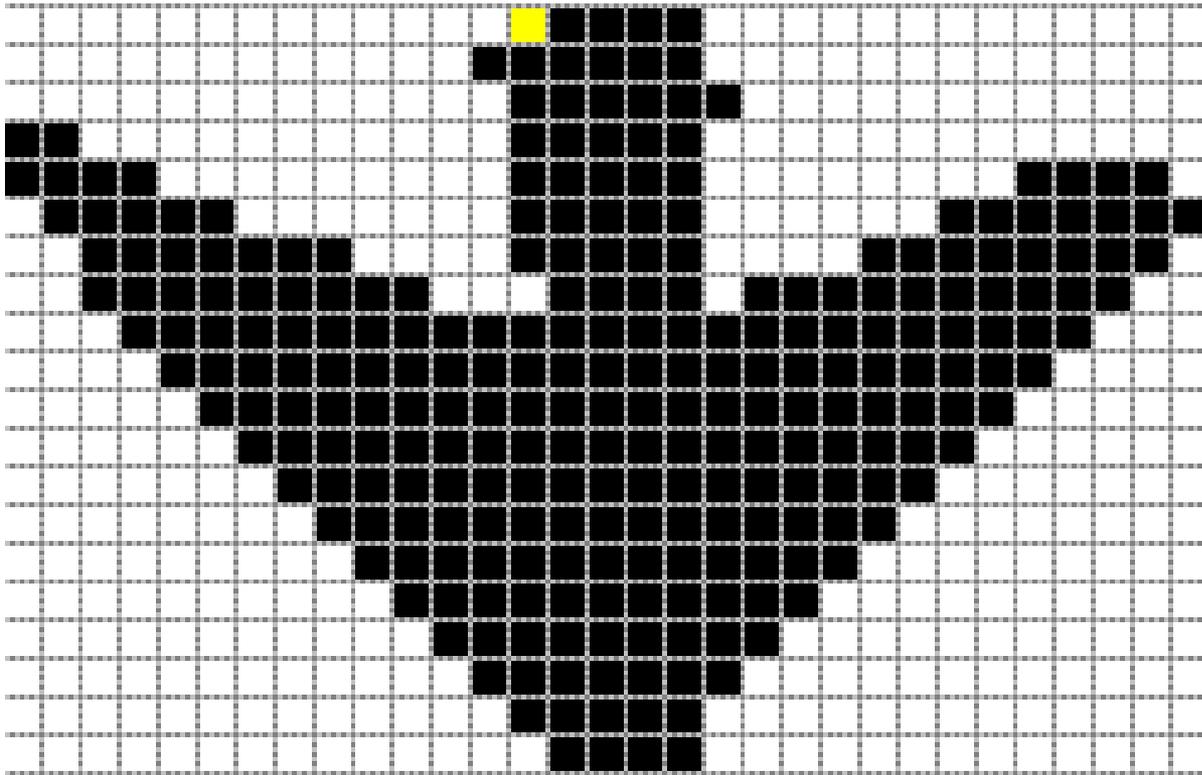


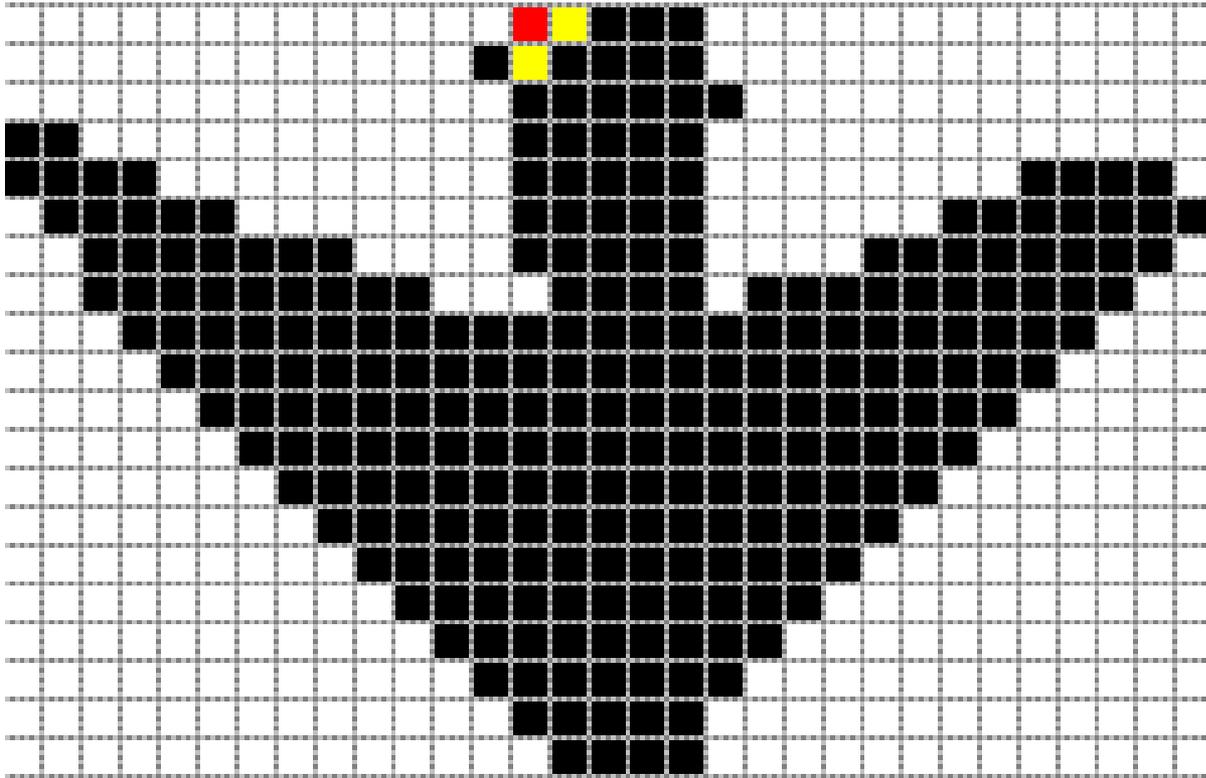


**CETM**  
Curso de Especialização em  
Tratamento de Minérios

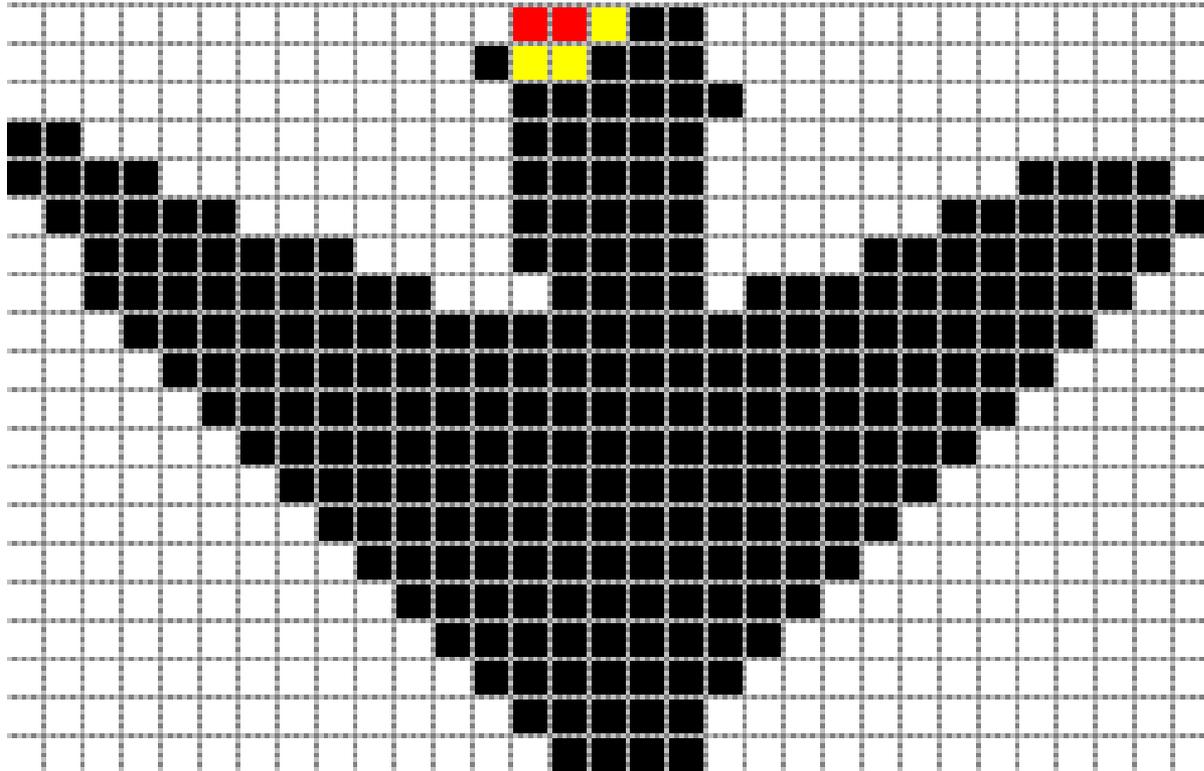




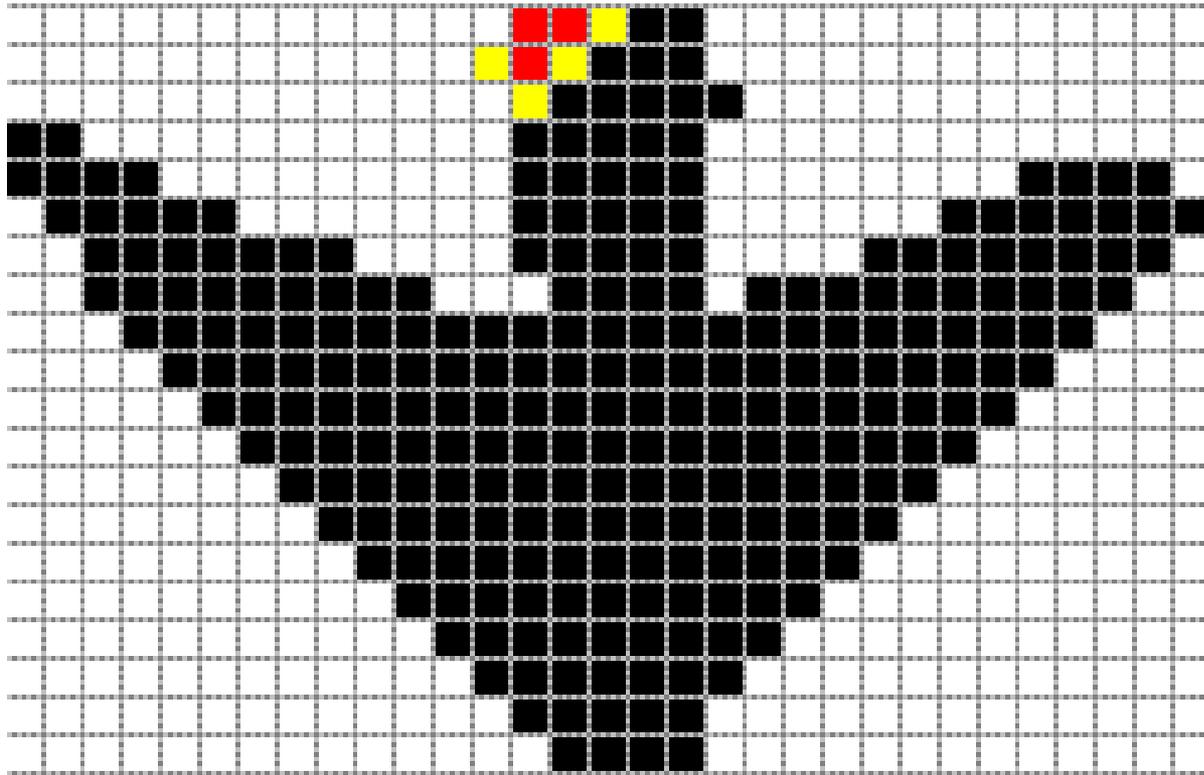




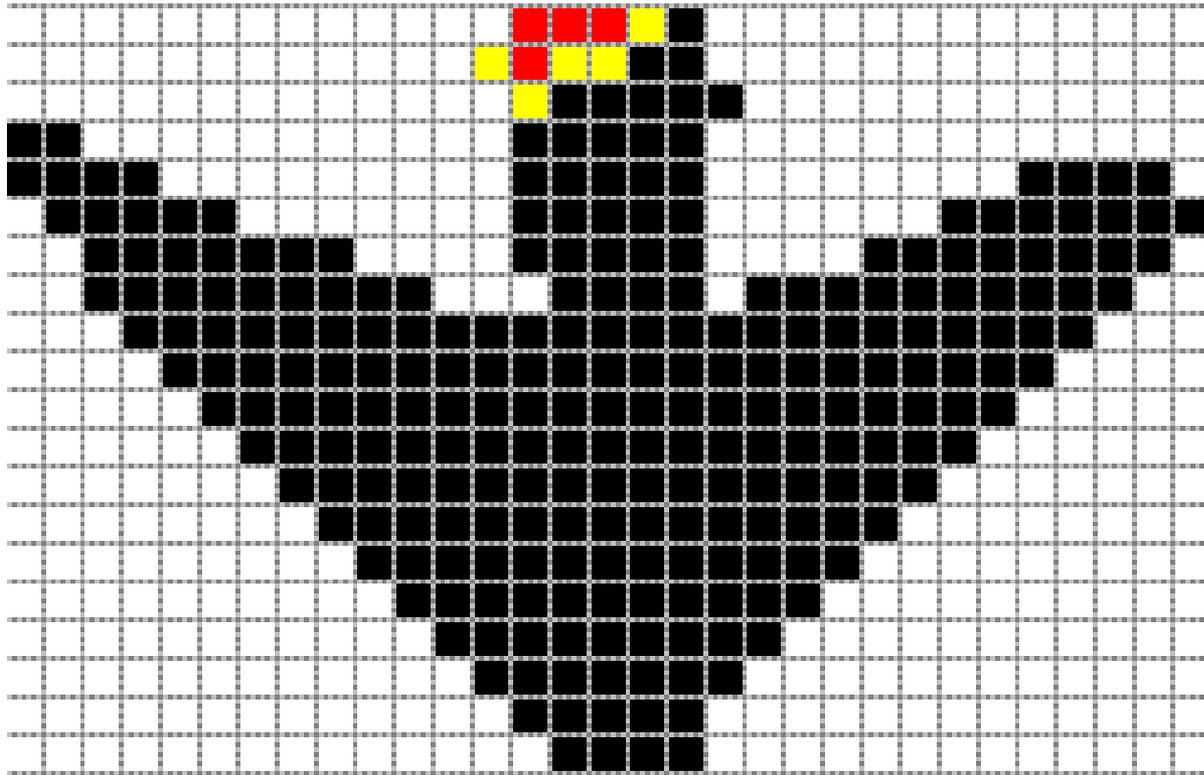
	45
14	15



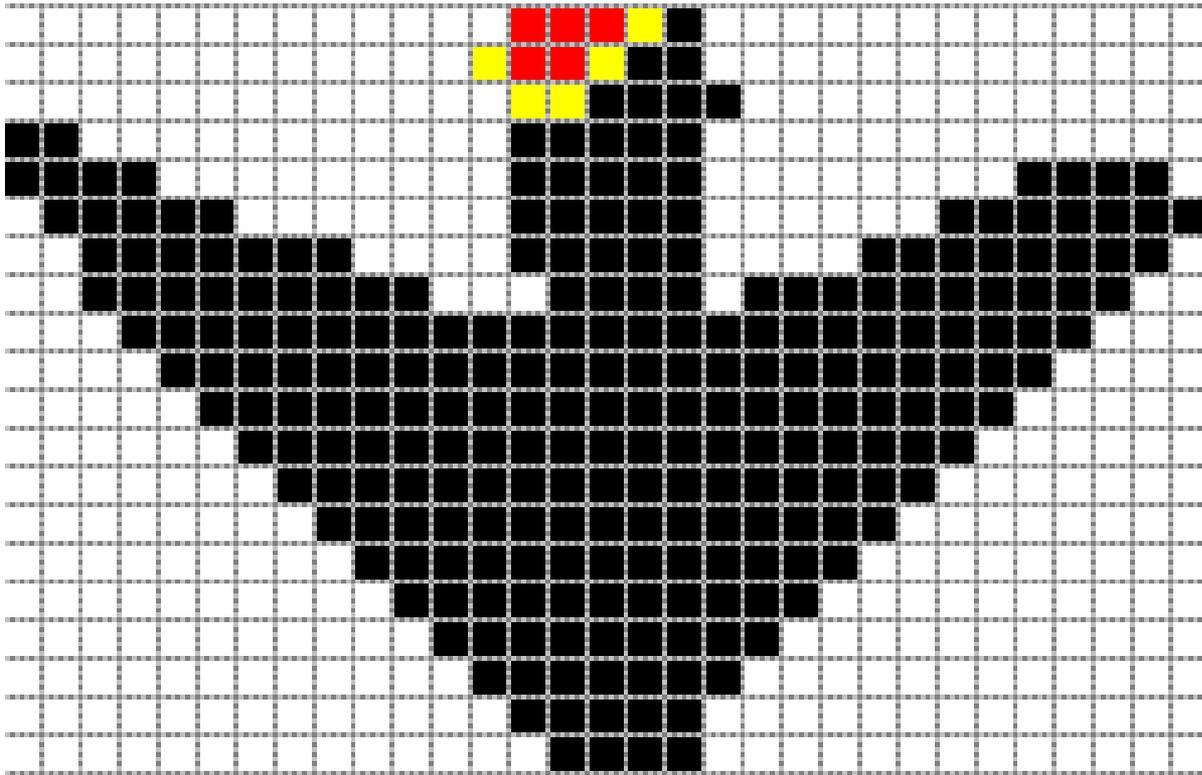
		46
15		16
14		45



	76
45	44
15	46
14	16



	47
16	17
45	76
15	44
14	46



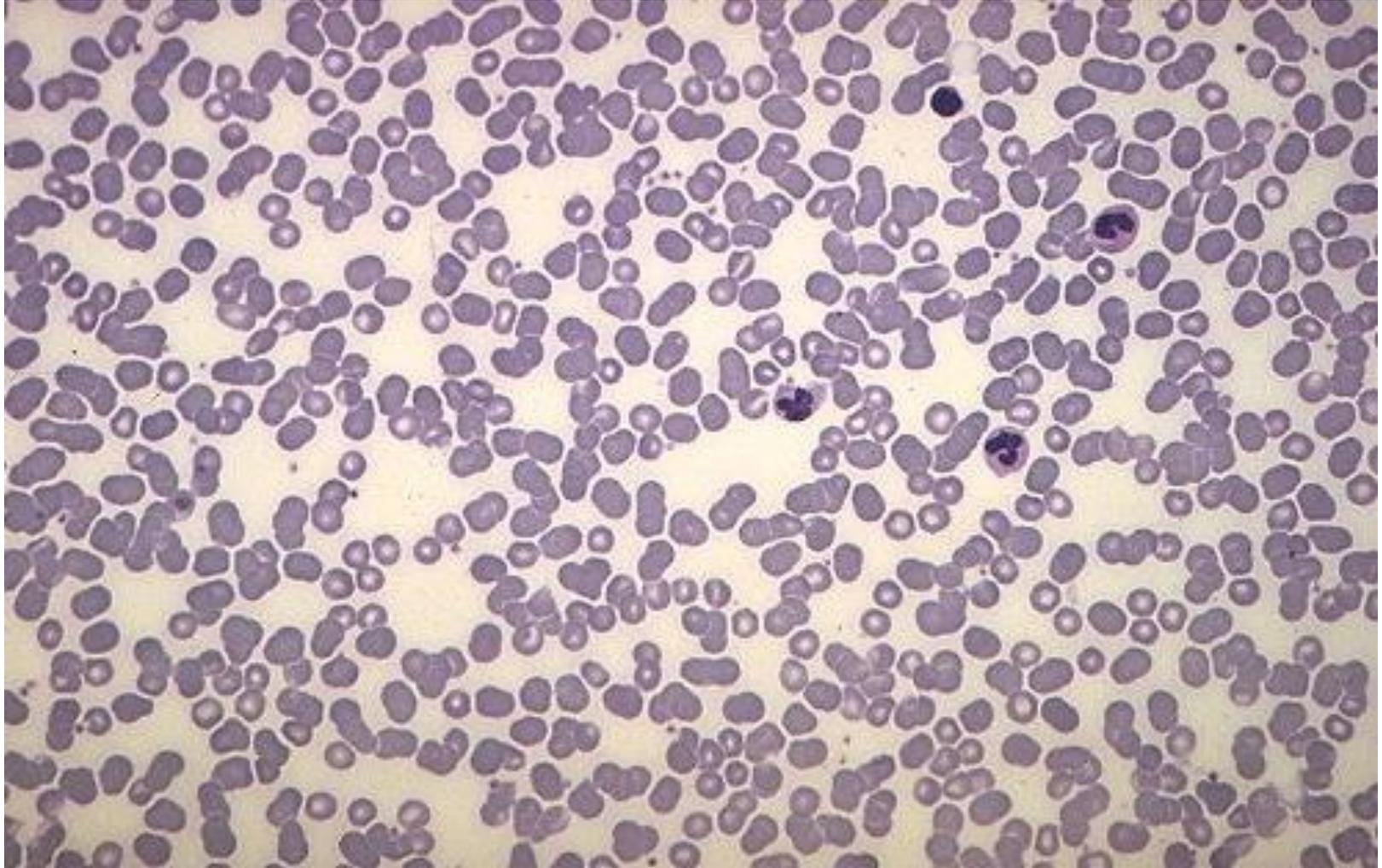
46	77
16	47
45	17
15	76
14	44

## 8.4. Algoritmo Silva e Buonfioli

- Foram feitos testes com imagens que possuíam clusters, fossem tais figuras produzidas para testes ou imagens reais.
- A figura abaixo apresenta uma imagem de microscopia de hemácias com tamanho de 497 x 320 pixels.



**CETM**  
Curso de Especialização em  
Tratamento de Minérios

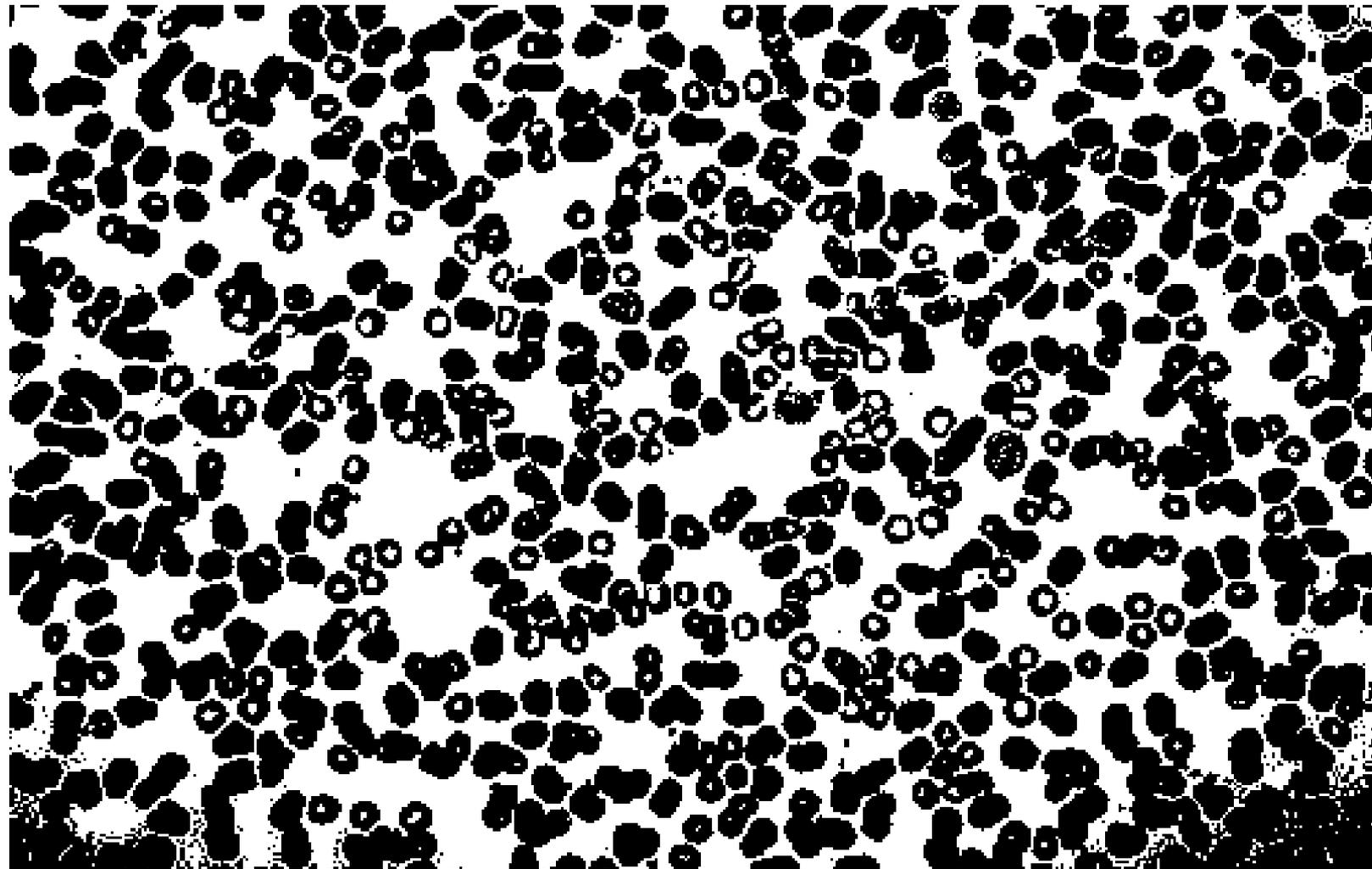


## 8.4. Algoritmo Silva e Buonfioli

- A figura a seguir apresenta a mesma figura anterior, porém a esta foi aplicado um filtro para remover as cores, que para o processo de quantificação de clusters são indiferentes.

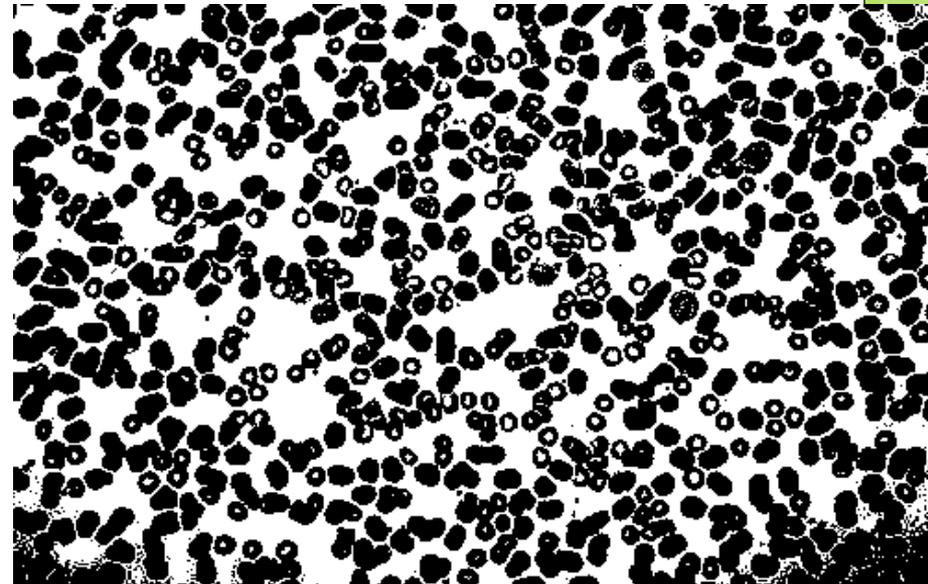
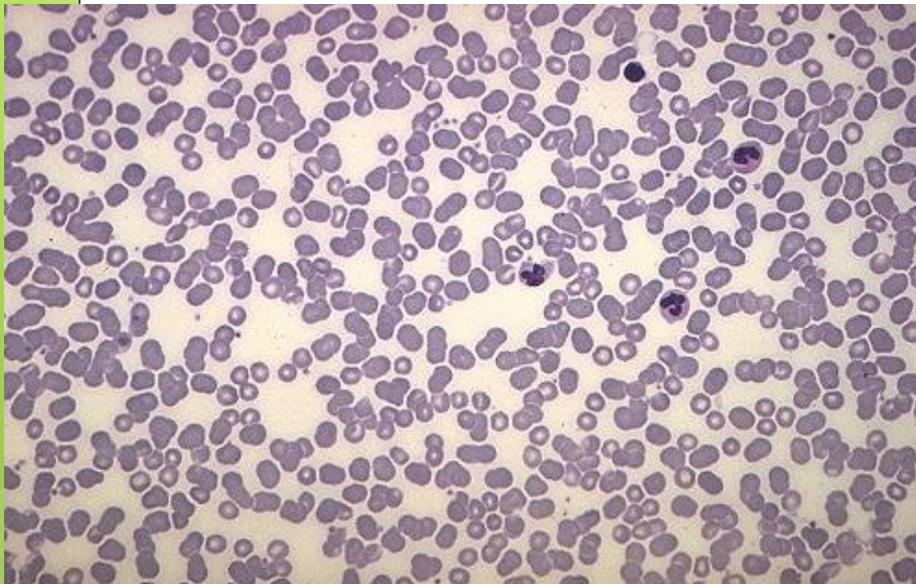


**CETM**  
Curso de Especialização em  
Tratamento de Minérios

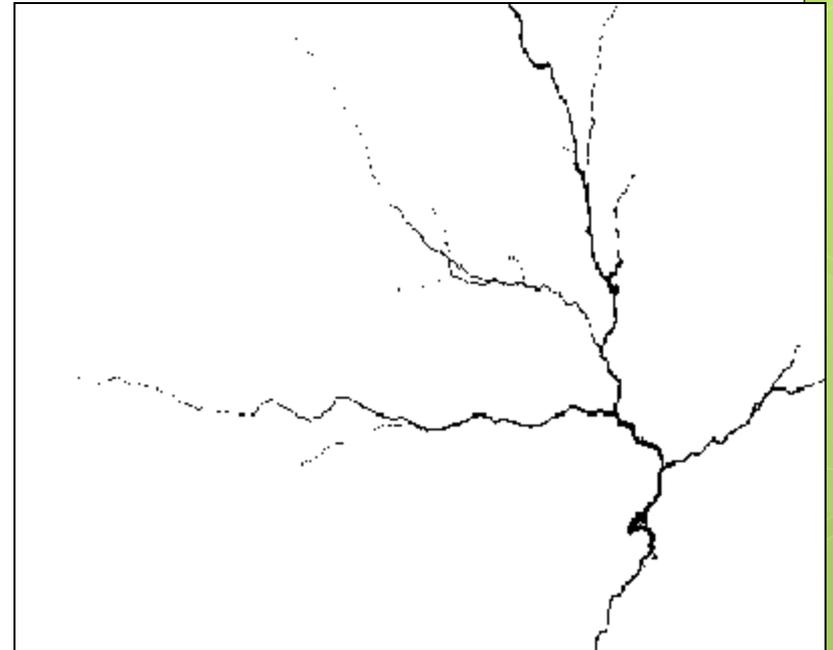


## 8.4. Algoritmo Silva e Buonfioli

- O resultado da contagem de clusters da figura foram:
  - 356 clusters encontrados;
  - Maior com 4.603 pixels;
  - Menor com 11 pixels;
  - Tamanho médio = 231,222 pixels;
  - Tempo total gasto = 1,391 segundos.
    - *Notebook Acer 5630 Core2Duo T5500 (1.67 GHz).*



- Tempo gasto: 00:00:01,391
- Clusters encontrados: 356
- Maior cluster: 4.603
- Menor cluster: 11
- Média dos clusters: 231,222

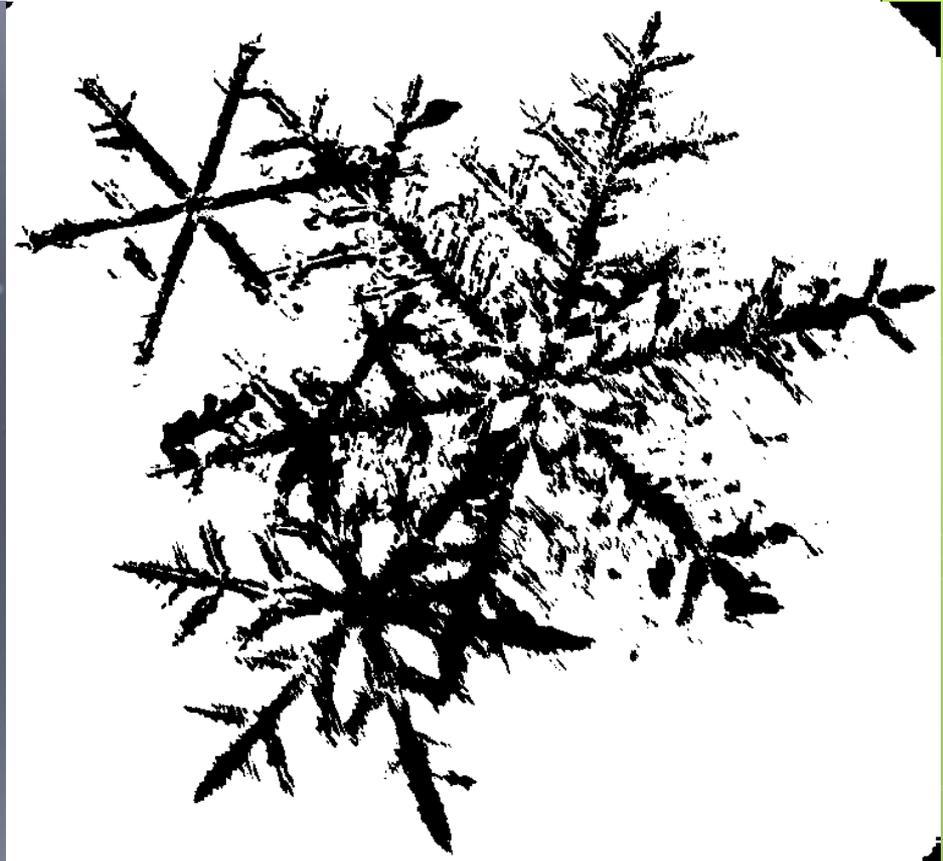


- Tempo gasto: 00:00:00.359
- Clusters encontrados: 455
- Maior cluster: 11
- Menor cluster: 1
- Média dos clusters: 1,708



**CETM**

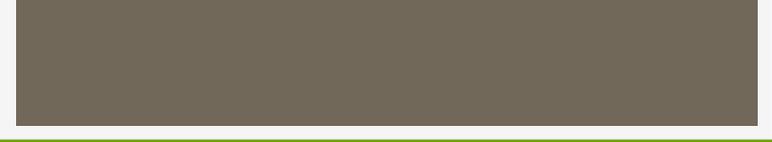
Curso de Especialização em  
Tratamento de Minérios



- Tempo gasto: 00:00:17.125
- Clusters encontrados: 691
- Maior cluster: 74398
- Menor cluster: 1
- Média dos clusters: 170,376

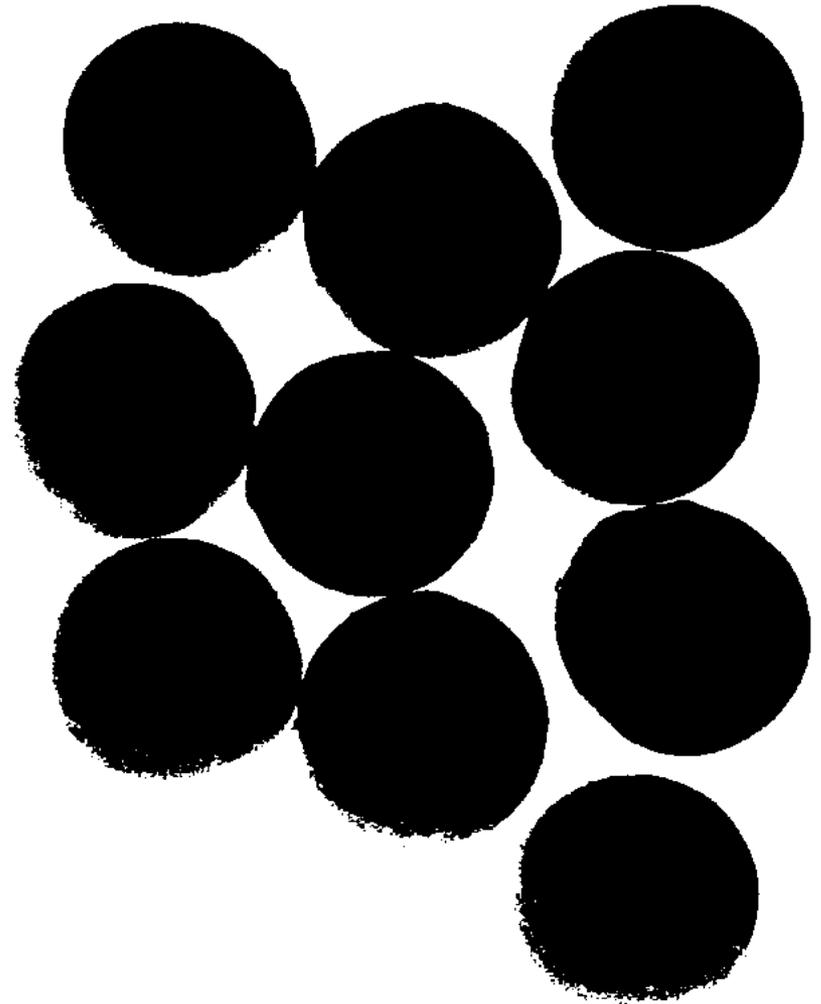
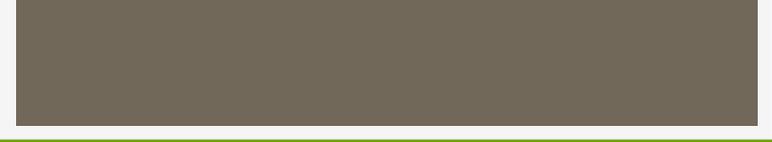


**CETM**  
Curso de Especialização em  
Tratamento de Minérios





**CETM**  
Curso de Especialização em  
Tratamento de Minérios





- Tempo gasto: 00:02:39.187
- Clusters encontrados: 2
- Maior cluster: 249057
- Menor cluster: 24541
- Média dos clusters: 136.799,000

