

MODELAGEM DE UM GERADOR AUTOMÁTICO DE MALHA DELAUNAY UTILIZANDO REDE DE PETRI COLORIDA

**Thomaz M. de ALMEIDA (1); Paulo C. CORTEZ (1); Auzuir R. de ALEXANDRIA (2);
Tarique da S. CAVALCANTE (1); Giovanni C. BARROSO (1);**

(1) Universidade Federal do Ceará – UFC, Caixa postal 6007, 60755-640 - Campus do Pici – Fortaleza/CE
{thomaz@lesc.ufc.br, cortez@lesc.ufc.br, tariquesc@yahoo.com.br, gcb@fisica.ufc.br}

(2) Instituto Federal de Educação Ciência e Tecnologia do Ceará - IFCE, Av. Treze de Maio, 2081 - Benfica - Fortaleza/CE, auzuir@ifce.com

RESUMO

A simulação e modelagem de sistemas são técnicas constantemente melhoradas através da Computação. Uma das melhorias proporcionadas pelo avanço da computação é o avanço das técnicas de modelagem e a triangulação é uma delas. A geração de malhas triangulares tem aplicações em temas como a modelagem de personagens de jogos para a Computação Gráfica, demarcação de espaço na Topografia/Cartografia, geração de malhas para resolução de problemas eletromagnéticos, segmentação de imagens e outros. Das técnicas mais avançadas para a geração de malha não estruturada os algoritmos Delaunay são os mais utilizados. Os algoritmos Delaunay, em especial, se baseiam em regras a fim de obter uma malha de refinamento positivo e adequado ao propósito estabelecido. Este trabalho propõe a modelagem da estrutura de dados de um gerador automático de malhas do tipo Delaunay utilizando redes de Petri colorida através da ferramenta CPN Tools. O modelo é testado no próprio *software* CPN Tools através de um método exaustivo e aleatório de disparos de transições. Ao final o modelo responde sem haver bloqueios ou marcações mortas ou não-vivas podendo ser sempre reinicializado.

Palavras-chave: triangulação de Delaunay, rede de petri colorida, cpn tools.

1 INTRODUÇÃO

A computação traz cada vez mais, novos métodos para resolução, simulação e modelagem de sistemas. Uma das melhorias proporcionadas pelo avanço da computação é o avanço das técnicas de modelagem e a triangulação é uma delas. A criação de malhas triangulares é uma artifício que pode ser utilizado em diversas áreas a fim de modelagem e simulação. Suas aplicações vão desde a modelagem de personagens de jogos para a Computação Gráfica, demarcação de espaço na Topografia/Cartografia até a geração de malhas para resolução de problemas de cálculo de campos eletromagnéticos e segmentação de imagens médicas.

As técnicas mais avançadas para a geração de malha não estruturada da atualidade são os algoritmos Delaunay e avanço de fronteira. Os algoritmos Delaunay, em especial, se baseiam em regras a fim de obter uma malha de refinamento positivo e adequado ao propósito estabelecido.

O objetivo deste trabalho consiste em modelar a estrutura de dados de um algoritmo gerador de malhas triangulares do tipo Delaunay utilizando redes de Petri colorida de forma a obter um protótipo adaptável a qualquer linguagem de programação orientada a objetos.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Rede de Petri

As redes de Petri, introduzidas em 1962, são uma ferramenta para o estudo de sistemas que permitem uma representação matemática e análise aplicável em sistemas de processamento de informação com características concorrentes, assíncronas, distribuídas, paralelas e estocásticas (MURATA, 1989).

Uma das principais vantagens da utilização de redes de Petri está no seu formalismo matemático e representação gráfica. Uma rede de Petri é constituída basicamente de dois tipos de componentes: um ativo denominado transição, correspondente a uma ação realizada dentro do sistema, e outro passivo denominado

lugar, relacionado a uma variável de estado do sistema (MURATA, 1989). Os arcos que ligam lugares a transições correspondem à relação entre condições verdadeiras que, em um dado momento, possibilitam a execução das ações enquanto que, os arcos que ligam transições a lugares representam a relação entre ações e condições que se tornam verdadeiras com a execução das ações.

Dessa maneira, Murata (1989) define uma rede de Petri como uma quintupla, $RP = (P, T, A, WM_0)$, em que:

- $P = P_1, P_2, \dots, P_m$ é um conjunto finito de lugares;
- $T = T_1, T_2, \dots, T_n$ é um conjunto finito de transições;
- $A \subseteq (P \times T) \cup (T \times P)$ é um conjunto finito de arcos;
- $W: F \rightarrow \{1, 2, 3, \dots\}$ é uma função peso;
- $M_0: P \rightarrow N^*$ é a marcação inicial, em que N denota os números naturais.

A simulação de uma rede de Petri ocorre de acordo com as suas regras de disparo, ou seja, uma transição está habilitada para disparar se houver um número de fichas no lugar de entrada igual ou superior ao peso do arco que as conectam. Após o disparo da transição, os lugares de entrada perdem fichas na mesma quantidade do peso do arco, sendo introduzidas nos lugares de saída com quantidade equivalente ao peso do arco.

2.2 Triangulação de Delaunay

As primeiras simulações computacionais eram realizadas por métodos numéricos (utilizando diferenças finitas ou método dos elementos finitos) na qual as malhas eram construídas de forma manual. Foi apenas nos anos 60, com a utilização dos computadores, que se tornou possível discretizar domínios mais complexos (BATISTA, 2005). A partir dos anos 80, com os computadores mais popularizados, surgiram várias técnicas de geração de malhas. Dentre essas técnicas a malha de Delaunay se destacou por possuir em seu critério básico a construção de elementos de forma que não exista nenhum vértice de poliedro que esteja no interior da esfera circunscrita de um outro poliedro (OWEN, 1998). Isto é, dado um conjunto de pontos a gerar uma determinada malha, existe um elevado número de maneiras de uni-los formando triângulos, porém o matemático Boris Delaunay provou que existe apenas uma única solução para unir os pontos de forma a maximizar o menor ângulo do triângulo tornando-os assim os mais equiláteros possíveis (Figura 1).

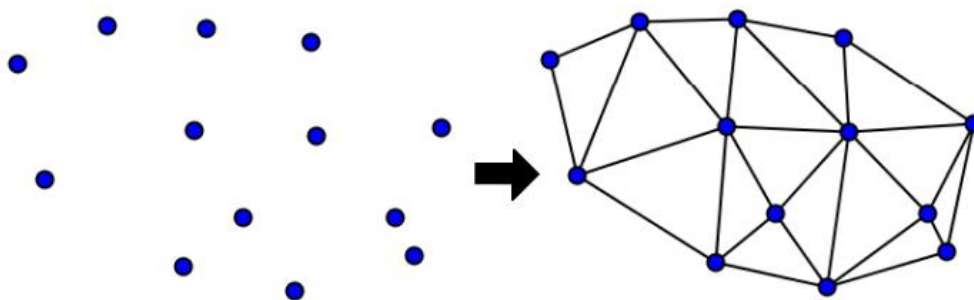


Figura 1 – Pontos formadores da malha Delaunay

A triangulação de Delaunay tem como dual o diagrama de Voronoi (Figura 2). Diagrama esse que divide o plano 2D em regiões contendo os pontos P tal que cada ponto P está contido em apenas uma região. Assim como a triangulação de Delaunay, os diagramas de Voronoi também tem suas aplicações no qual se destaca o típico exemplo dos correios, onde a área de atuação de cada agência precisa ser determinada. Considerando que as agências são idênticas, o diagrama de Voronoi das mesmas fornece regiões de abrangência desejadas. Porém, se as agências tiverem um número diferente de funcionários é de se esperar que a agência com mais funcionários possua uma região de abrangência maior (PIRES, 2008).

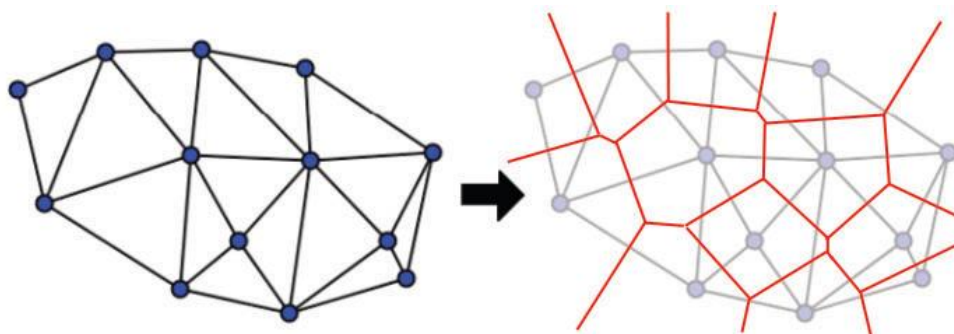


Figura 2 – Pontos formadores do diagrama de Voronoi

3 METODOLOGIA

3.1 Estrutura de Dados

A estrutura de dados abordada neste artigo é trabalhada em termos de Programação Orientada a Objetos, sendo, desta forma, dividida em classes que simulam os componentes da triangulação de Delaunay. Estes componentes são detalhados a seguir.

- **Nó:** é composto por duas coordenadas (x,y para malhar triangulares) e é responsável por simular os vértices dos triângulos da malha.
- **Segmento:** é composto por dois Nós e representa as arestas dos triângulos da malha.
- **Triângulo:** encarregado de fazer a união dos Segmentos. Logo, é composto por três segmentos.

As redes de Petri ditas coloridas reduzem o tamanho do modelo permitindo que marcas individualizadas (cores) representem diferentes processos ou recursos. Desta maneira, cada componente da estrutura de dados é representado por uma cor e cada cor é formada por diferentes tipos de dados padrões, como por exemplo “string”, “inteiro” e “booleano”:

- **Cor Nó:** formada por duas cores do tipo INT (inteiro).
- **Cor Segmento:** formada por duas cores do tipo NÓ.
- **Cor Triângulo:** formado por três cores do tipo SEGMENTO.

Além dessas três cores principais, outras cores auxiliares foram criadas, como por exemplo, a cor “Malha”. Essa cor é encarregada de atualizar a malha a cada novo triângulo criado ou apagado.

A criação de triângulos, bem como sua deleção e outras funções, é modelada em diferentes abas (ou *subpages*), como mostra a Figura 3.

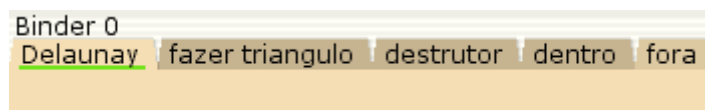


Figura 3 – Páginas do modelo.

3.2 Adição de Nós

Seguindo o critério de Delaunay, quando um novo ponto for adicionado este terá sua posição verificada a saber se o mesmo encontra-se dentro de alguma circunferência de algum triângulo da atual malha. Para isto, seguem os passos de verificação:

- São verificados todos os triângulos cujas circunferências que os circunscrevem envolvem o ponto a ser inserido, pois o novo ponto não pode estar dentro de nenhuma destas.

Seja uma circunferência que passa pelos pontos (x_1, y_1) , (x_2, y_2) , (x_3, y_3) . Utilizando a equação geral da reta definida por:

$$x^2 + y^2 + Ax + Bx + C = 0 \quad [\text{Eq. 01}]$$

Aplicando essa equação para os três nós, obtém-se:

$$x_1^2 + y_1^2 + Ax_1 + By_1 + C = 0 \quad [\text{Eq. 02}]$$

$$x_2^2 + y_2^2 + Ax_2 + By_2 + C = 0 \quad [\text{Eq. 03}]$$

$$x_3^2 + y_3^2 + Ax_3 + By_3 + C = 0 \quad [\text{Eq. 04}]$$

Reorganizando [Eq. 02], [Eq. 03] e [Eq. 04] e aplicando a regra de Crammer obtém-se A, B e C,

$$A = \frac{\begin{vmatrix} -(x_1)^2-(y_1)^2 & y_1 & 1 \\ -(x_2)^2-(y_2)^2 & y_2 & 1 \\ -(x_3)^2-(y_3)^2 & y_3 & 1 \end{vmatrix}}{\begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix}} \quad B = \frac{\begin{vmatrix} x_1 & -(x_1)^2-(y_1)^2 & 1 \\ x_2 & -(x_2)^2-(y_2)^2 & 1 \\ x_3 & -(x_3)^2-(y_3)^2 & 1 \end{vmatrix}}{\begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix}} \quad C = \frac{\begin{vmatrix} x_1 & y_1 & -(x_1)^2-(y_1)^2 \\ x_2 & y_2 & -(x_2)^2-(y_2)^2 \\ x_3 & y_3 & -(x_3)^2-(y_3)^2 \end{vmatrix}}{\begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix}} \quad [\text{Eq.05}]$$

Assim é possível chegar às coordenadas do centro da circunferência (C_x, C_y) que circunscreve o triângulo formados pelos pontos (x_1, y_1) , (x_2, y_2) , (x_3, y_3) e seu devido raio, definidos respectivamente por

$$C_x = -\frac{A}{2}, \quad [\text{Eq. 06}]$$

$$C_y = -\frac{B}{2} \quad [\text{Eq. 07}]$$

$$Raio = \sqrt{C_x^2 + C_y^2 - C} \quad [\text{Eq. 08}]$$

Com essas informações é possível mensurar a distância do centro ao novo ponto. Se essa distância for menor ou igual ao raio, o novo ponto está dentro da circunferência, se não, está fora.

- ii. Caso o ponto esteja inserido na circunferência de algum triângulo os segmentos do mesmo devem ser marcados para serem apagados, pois estes triângulos não mais existirão.

Depois de apagados os segmentos, são criados novos triângulos com os vértices dos antigos e o novo ponto. Na Figura 4a é mostrada uma malha inicial formada por quatro pontos, dois triângulos e duas circunferências. Na Figura 4b um novo nó é inserido dentro de apenas um triângulo e a circunferência que o circunscreve. Na figura 4c são mostradas as novas ligações feitas entre o novo nó e os vértices do triângulo cujo ponto está dentro. Por fim, na Figura 4d aparece a malha final.

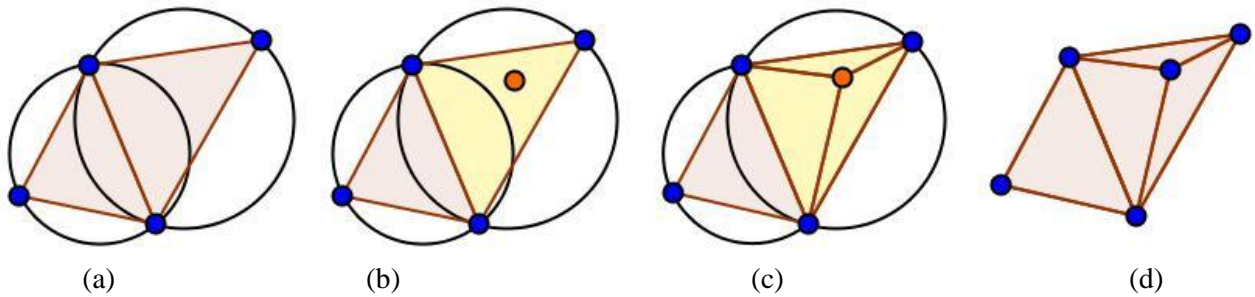


Figura 4 – Sequência de adição de um novo nó. (a) Malha inicial. (b) Novo nó inserido. (c) Novas ligações entre o novo nó e os vértices do triângulo. (d) Malha resultante.

- iii. Se o ponto não estiver inserido em nenhuma circunferência, ele é conectado a qualquer triângulo desde que não haja interseção entre os vértices desse triângulo e o novo nó a ser adicionado. Logo, é necessário testar os segmentos feitos com os vértices de cada triângulo em relação ao novo nó e marcar como inválido os triângulos cujo vértice faz interseção ao novo nó. Para testar as interseções é utilizado o método de equação paramétrica das retas:

Dado um segmento de reta AB e outro CD, são encontrados os parâmetros α (segmento AB) e β (segmento CD). Se α e β retornados pelas equações [Eq. 09] e [Eq. 10] estiverem entre 0 e 1, então a interseção é verdadeira.

$$\alpha = \frac{(D_x - C_x)(C_y - A_y) - (D_y - C_y)(C_x - A_x)}{(D_x - C_x)(B_y - A_y) - (D_y - C_y)(B_x - A_x)} \quad [\text{Eq. 09}]$$

$$\beta = \frac{(B_x - A_x)(C_y - A_y) - (B_y - A_y)(C_x - A_x)}{(D_x - C_x)(B_y - A_y) - (D_y - C_y)(B_x - A_x)} \quad [\text{Eq. 10}]$$

Se o denominador da [Eq. 09] ou da [Eq. 10] for igual a zero ou α e β estiverem fora do intervalo de zero e um não há interseção.

Mesmo se o novo ponto estiver dentro da circunferência de algum triângulo, é necessária a verificação de interseção do novo nó com algum outro vértice de algum outro triângulo, pois é possível que um ponto esteja dentro de uma circunferência e ao mesmo tempo é passivo de conexão com algum outro triângulo. Pode-se exemplificar o que acontece pela figura 5. Na Figura 5a há uma malha inicial com quatro pontos, dois triângulos e duas circunferências. Na Figura 5b se vê que o novo ponto está dentro apenas da circunferência do triângulo da direita, porém, malha resultante que é ligação de todos os vértices possíveis e não apenas daqueles cujo triângulo gerou a circunferência na qual o novo ponto está inserido (Figura 5c).

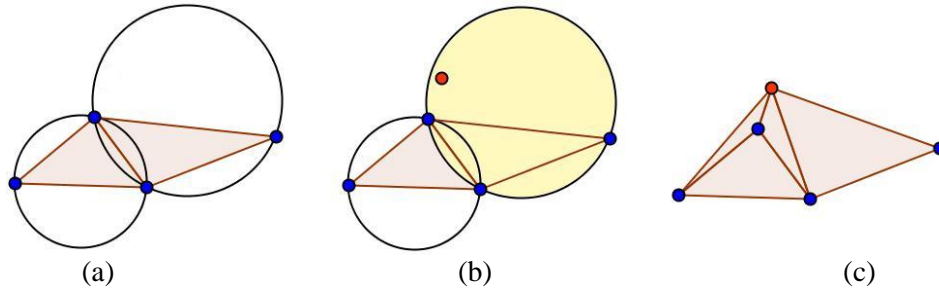


Figura 5 – Sequência de adição de um novo nó. (a) Malha inicial. (b) Novo nó. (c) Malha resultante com novas ligações.

3.3 Modelagem

Nesta seção são detalhadas as abas (ou *subpages*) e funções do modelo criado no CPN Tools.

3.3.1 Parte 1 – Malha Inicial

A primeira parte do modelo simula a abertura de uma imagem (a ser utilizada como plano de fundo) em um *software* em que o usuário fornece a primeira malha manualmente (composta por apenas um triângulo). Para isto, a imagem em branco é modelada por uma cor do tipo *String*: “teste.bmp”.

Depois de simulada a abertura da imagem três transições são habilitadas, uma após a outra, e a cada disparo destas transições, dois números inteiros são gerados aleatoriamente. Cada par de inteiros representa os primeiros três vértices.

Com os três vértices gerados aleatórios a aba “Fazer Triângulo” é habilitada.

3.3.2 Parte 2 – Criação da Malha

Esta função exemplifica bem semelhança do modelo com uma estrutura de dados orientada a objetos. A partir da Figura 6 percebe-se que a transição “CRIAR NOH” tem na sua entrada dois arcos com dois valores inteiros e na sua saída uma variável de cor PONTO (representando o Nó criado). Com dois Nós a transição “CRIAR SEGMENTO” é habilitada. Depois de criados os três segmentos, enfim, a transição “CRIAR TRIÂNGULO” é habilitada e quando disparada atualiza um contador e um lugar com a cor do tipo “MALHA”.

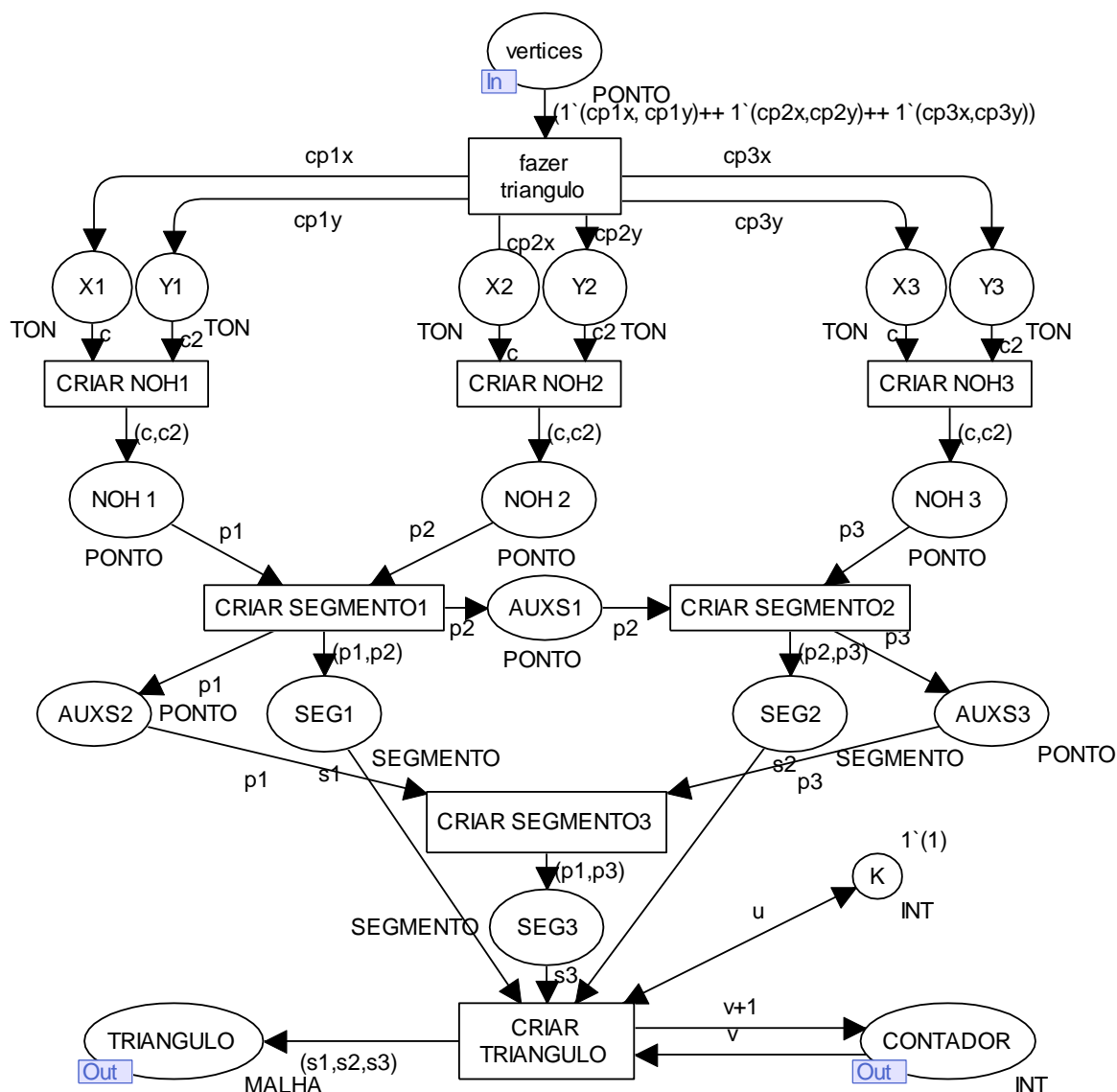


Figura 6 – Função criadora de triângulos.

3.3.3 Parte 3 – Adição do Novo Nó

A fim de evitar os cálculos apresentados na Seção 3.2, dado que este é apenas um modelo, os cálculos são abstraídos e simulados por uma cor do tipo “*Booleana*” em que retornará uma resposta se o novo ponto está dentro de alguma circunferência ou fora dela. Esta abstração não influencia no resultado final do modelo visto que qualquer linguagem de programação orientada a objetos está plenamente apta a fazer esses cálculos tidos como básicos.

3.3.4 Parte 4 – Destrutor

Sabendo que existe uma ficha para cada triângulo da malha, o destrutor da triangulação deve apagar todas as fichas uma por uma. Para isso é implementado um contador de fichas que habilita uma transição sorvedouro para apagar todas as fichas.

O destrutor pode ser acionado a qualquer momento da modelagem desde que não se esteja criando alguma nova malha naquele instante. Depois de acionado, o destrutor só libera uma ficha na saída quando todos os triângulos forem apagados. Depois de liberada, uma ficha do tipo *IMAGEM* segue para o início simulando uma função de *reset* e carregando uma nova imagem para um novo início. A função do destrutor é ilustrada na Figura 7.

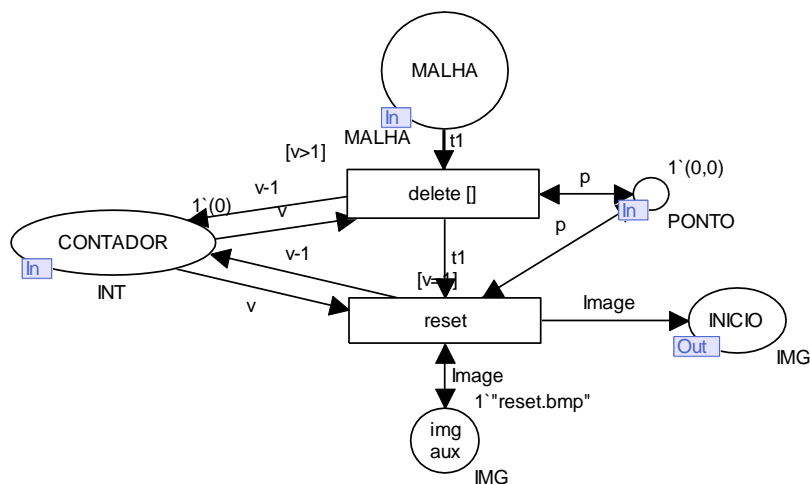


Figura 7 – Função destruidor.

4 RESULTADOS

A metodologia de testes foi implementada junto ao CPN Tools disparando todas as transições de forma manual e posteriormente de forma automática. O *software* possui um bloco de ferramentas chamado “*Simulation*” que é capaz de simular N disparos de forma aleatória, isto é, qualquer transição habilitada pode disparar. Foram simulados 500 disparos e se viu que o modelo sempre entra no destruidor em algum momento e retorna à marcação inicial. Porém, se através do acaso a transição do destruidor não for acionada o número de fichas, assim como o número de triângulos, no lugar “MALHA” (ver Apêndice A) tende ao infinito. O que é normal e possível.

5 CONCLUSÃO

Neste trabalho foi apresentada uma modelagem de uma estrutura de dados de um *software* através da ferramenta *CPN Tools*, onde é ilustrada na Figura 8 do Apêndice A. O trabalho alcança seu objetivo com sucesso apresentando idéias de modelos a serem implementados independente da linguagem adotada (neste caso a linguagem C++ já está sendo sendo utilizada para dar continuidade ao trabalho).

Vale ressaltar que o cálculo matemático foi abstraído da modelagem visto que a maioria das linguagens de programação está apta a fazer cálculos extremamente rápidos e de forma semelhante a outras linguagens. Logo, o trabalho se mostra completo no tocante à adaptação para outra linguagem de programação.

REFERÊNCIAS BIBLIOGRÁFICAS

- BATISTA V. H. F. **Geração de Malhas Não-Estruturadas Tetraédricas Utilizando um Método de Avanço de Fronteira**. 2005. 89 p. Dissertação (Mestrado), Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2005;
- MURATA, T. **Petri net: properties, analysis and applications**, Proceedings of the IEEE 77(4): 541-579. 1989.
- OWEN S. J. **A Survey of Unstructured Mesh Generation Technology**. Proceedings 7th International Meshing Roundtable, p. 239-267, 1998.
- PETERSON, J. L. **Petri Net Theory and the Modeling of Systems**, Prentice Hall PTR, Upper Saddle River, NJ, USA. 1981.
- PIRES, F. B. **Triangulações Regulares e Aplicações**. 57p. Dissertação (Mestrado em Ciências da Computação) - Universidade de São Paulo, São Paulo, 2008.

Apêndice A

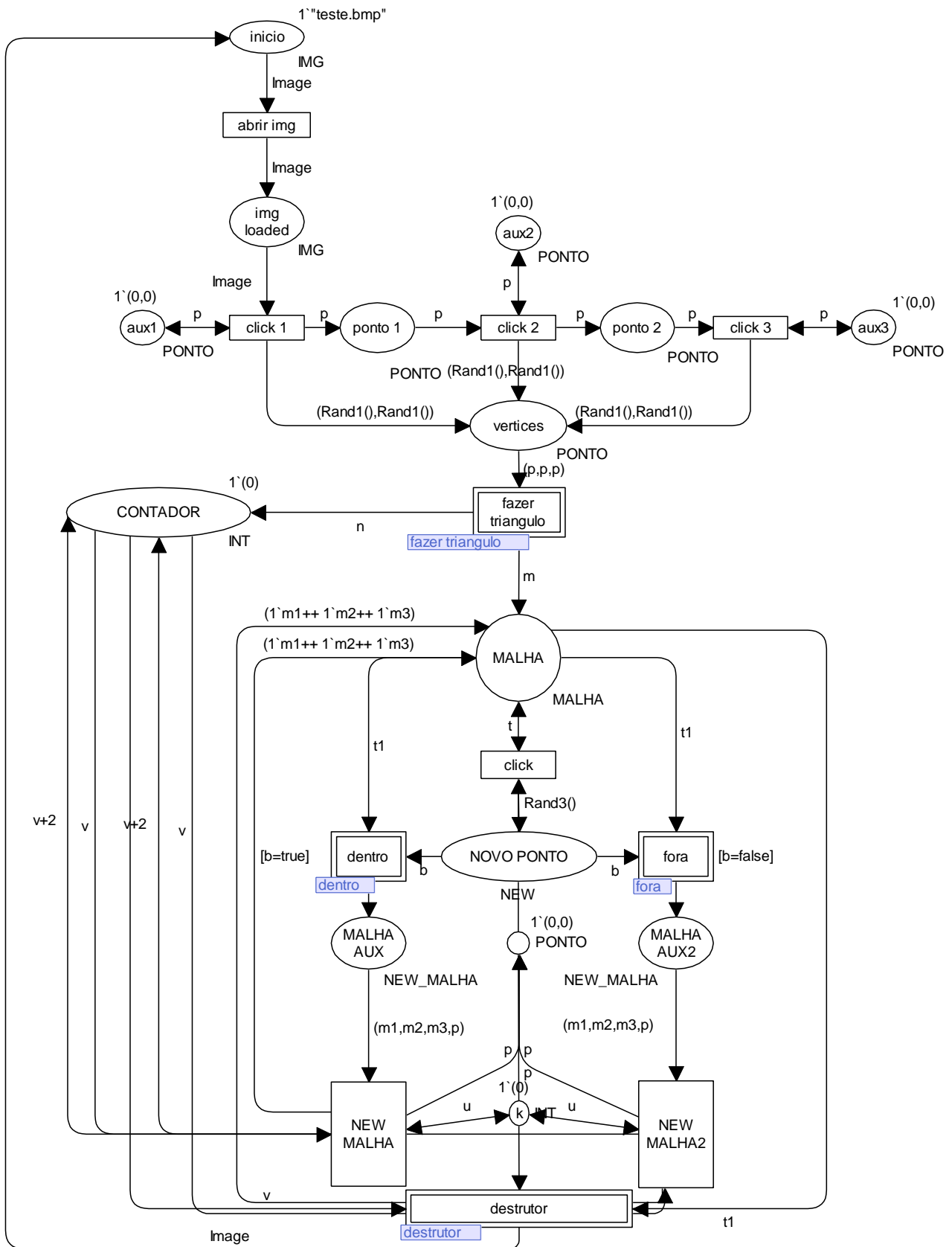


Figura 8 – Aba principal do modelo.