



Ururau

Free and open-source discrete event
simulation software

Coordenadores

- Prof. DSc João José de Assis Rangel
- Prof. DSc Ítalo de Oliveira Matias
- Prof. MSc Túlio Almeida Peixoto

Equipe Atual

- Eder Reis Tavares
- Eduardo Shimoda
- Fábio Freitas da Silva
- Geísa Pereira Marcilio
- Ítalo de Oliveira Matias
- João José de Assis Rangel
- Quézia Manuela Gonçalves Laurindo
- Túlio Almeida Peixoto
- Gustavo Rodrigues Fraga
- Cássio Rangel Paulista
- Tatiane Stellet Machado
- Yves Tavares Peres

Alunos Formados

- Anna Christine Azevedo Cordeiro
- Ariel Carvalho Nascimento
- Arthur Albuquerque Zopellaro Soares
- David Vasconcelos Corrêa da Silva
- Gabriel Lima de Oliveira
- Jhonathan Correa Camacho
- Leonardo das Dores Cardoso
- Marília Gonçalves Dutra da Silva

Autor: Yves Tavares Peres

Campos dos Goytacazes – 09 de novembro de 2016

Sumário

1- Baixando o Ururau	3
2- Interface do Ururau	5
Módulo <i>Create</i>	6
Módulo <i>Terminate</i>	6
Módulo <i>Function</i>	6
Módulo <i>Decision Maker</i>	8
Módulo <i>Batch</i>	10
Módulo <i>Separate</i>	11
Módulo <i>Record</i>	11
Módulo <i>Assign</i>	12
Módulo <i>Hold</i>	13
Módulo <i>Jump</i>	13

1- Baixando o Ururau

1.1. Acesse o site: bitbucket.org/ururau

1.2. Clique em URURAU como apresentado pela Figura 1:

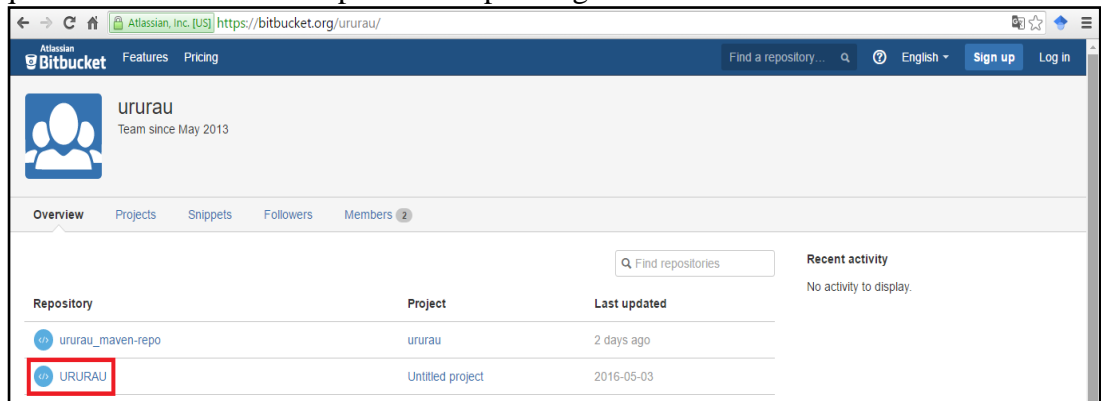


Figura 1: <https://bitbucket.org/ururau>

1.3. Clique em Downloads como apresentado pela Figura 2:

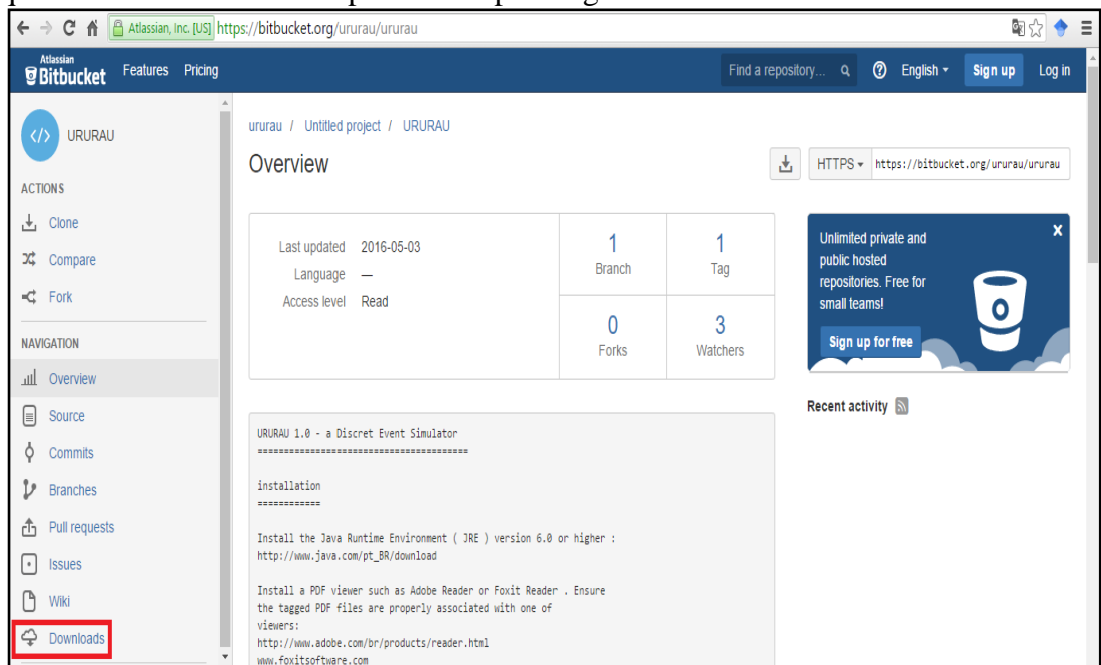


Figura 2: <https://bitbucket.org/ururau/ururau>

1.4. Clique em ururau-1.1.zip (Versão mais atualizada) como apresentado pela Figura 3:

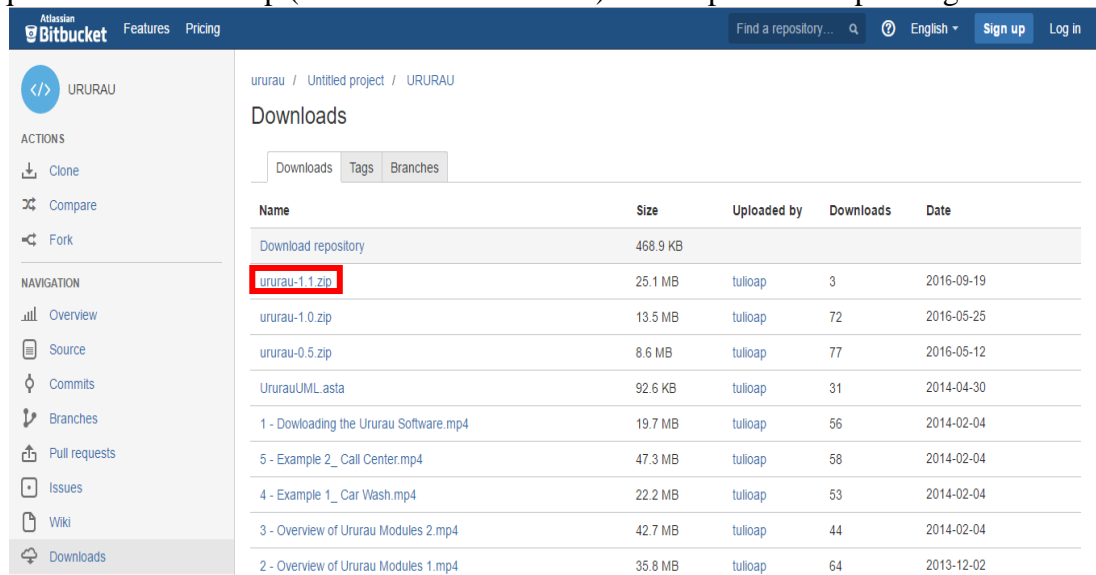


Figura 3: <https://bitbucket.org/ururau/ururau/downloads>

1.5. Após baixar o arquivo compactado, descompacte-o em um diretório e execute o arquivo ururau.jar.

Na Figura 4, o arquivo foi descompactado na área de trabalho:

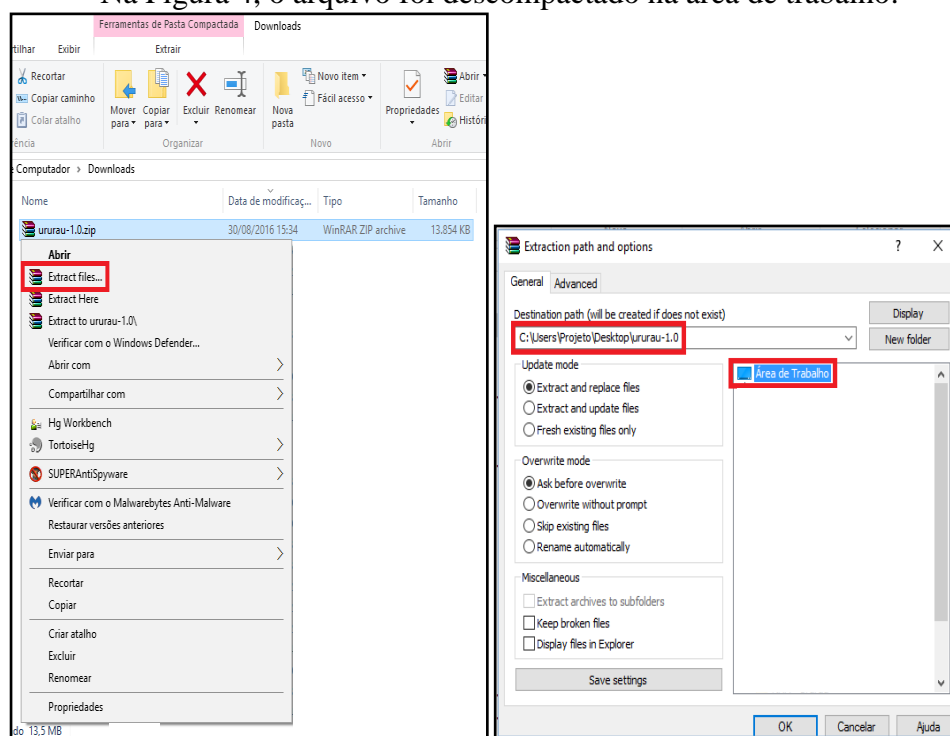


Figura 4: Descompactando o arquivo baixado.

2- Interface do Ururau

A Figura 5 demonstra a interface do Ururau.

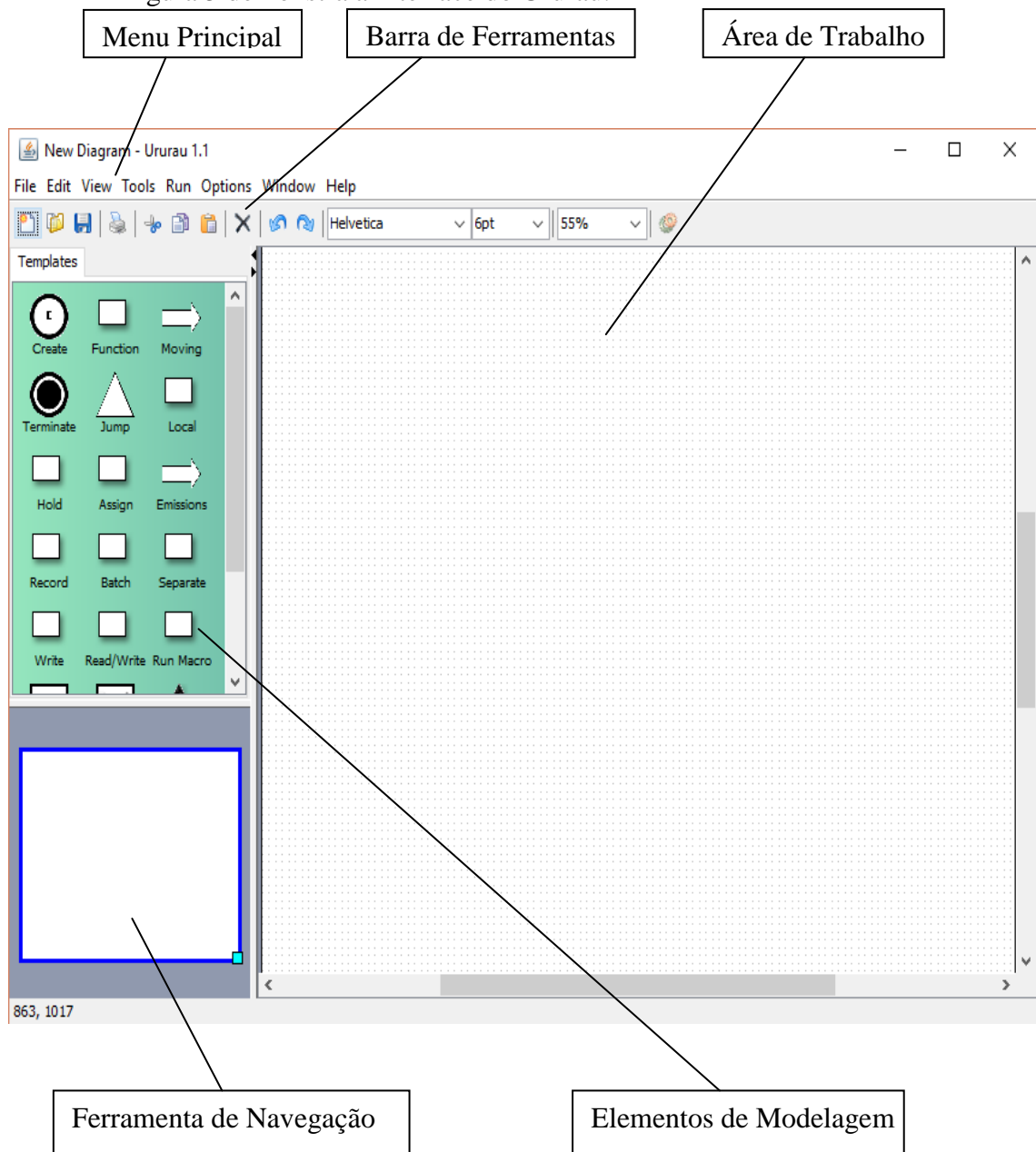


Figura 5: Demonstração da interface do Ururau.

Para iniciar a construção do modelo de simulação, é necessário clicar e arrastar os elementos de modelagem até a área de trabalho do Ururau e, em seguida, soltá-los. Para editar o módulo adicionado, é essencial clicar sobre ele com o botão direito do mouse e escolher a opção “editar”. Cada elemento possui a sua caixa de edição, que deve ser preenchida de acordo com os dados referente ao que se deseja simular.

A fim de ligar dois elementos, é preciso que o modelador passe o cursor do mouse em cima do primeiro elemento até que ele fique com a borda verde. Logo após, clicca-se e arrasta-se ele até o outro elemento; quando este ficar com a borda verde, deve-se soltá-lo.

Elementos principais:

2.1. Módulo *Create*: Esta classe vem antes de todos os comandos no processo de modelagem. É com ela que se criam as entidades. O módulo *Create* é apresentado na Figura 6:

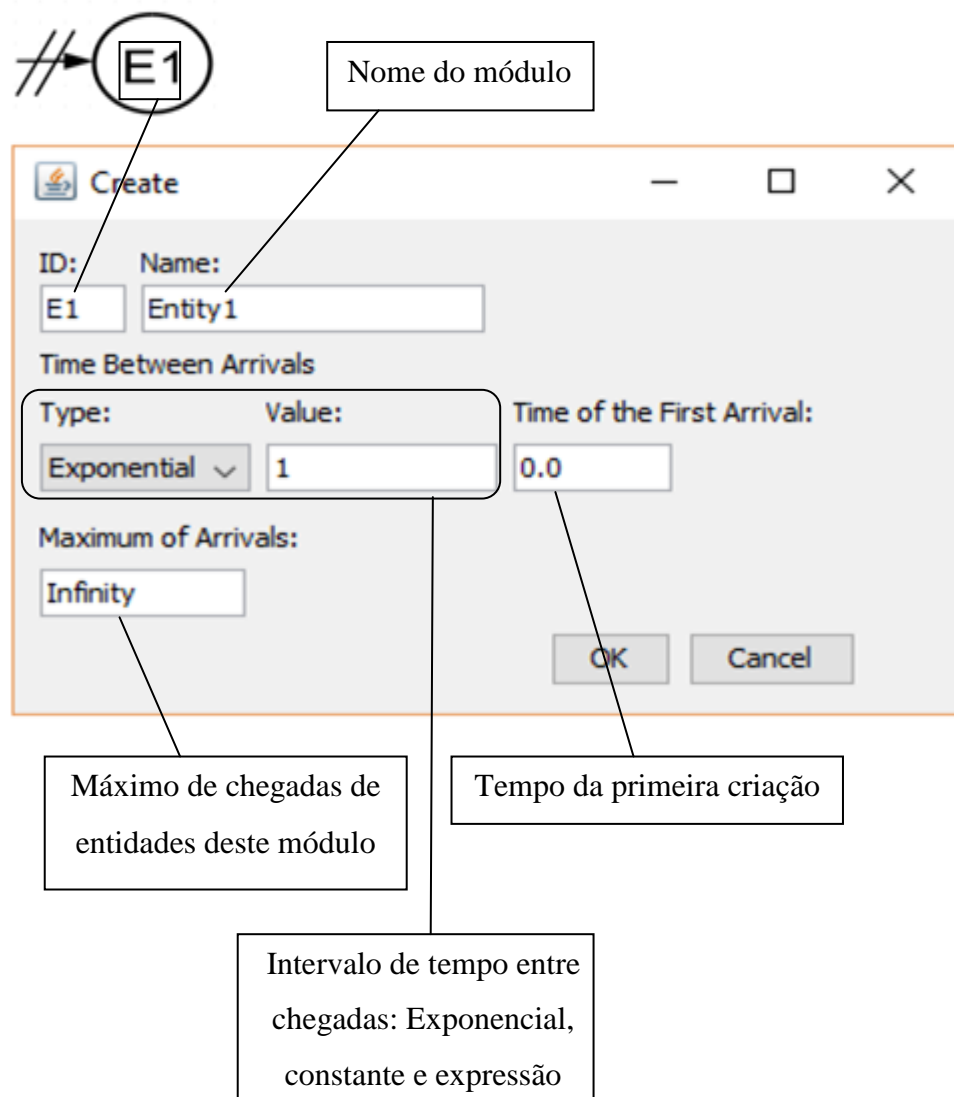


Figura 6: Demonstração do módulo *Create*.

2.2. Módulo *Terminate*: Finaliza a lista de comandos. É contabilizada a quantidade de entidades visíveis até esse ponto. O módulo *Terminate* é apresentado na Figura 7:



Figura 7: Demonstração do módulo *Terminate*.

2.3. Módulo *Function*: É o comando onde está contido o *seize*, *delay*, *release* nesta ordem, ou seja, primeiro segura a entidade, demora um tempo no processo e libera. Na Figura 8 o módulo funciona como um *delay*, pois não tem recurso.

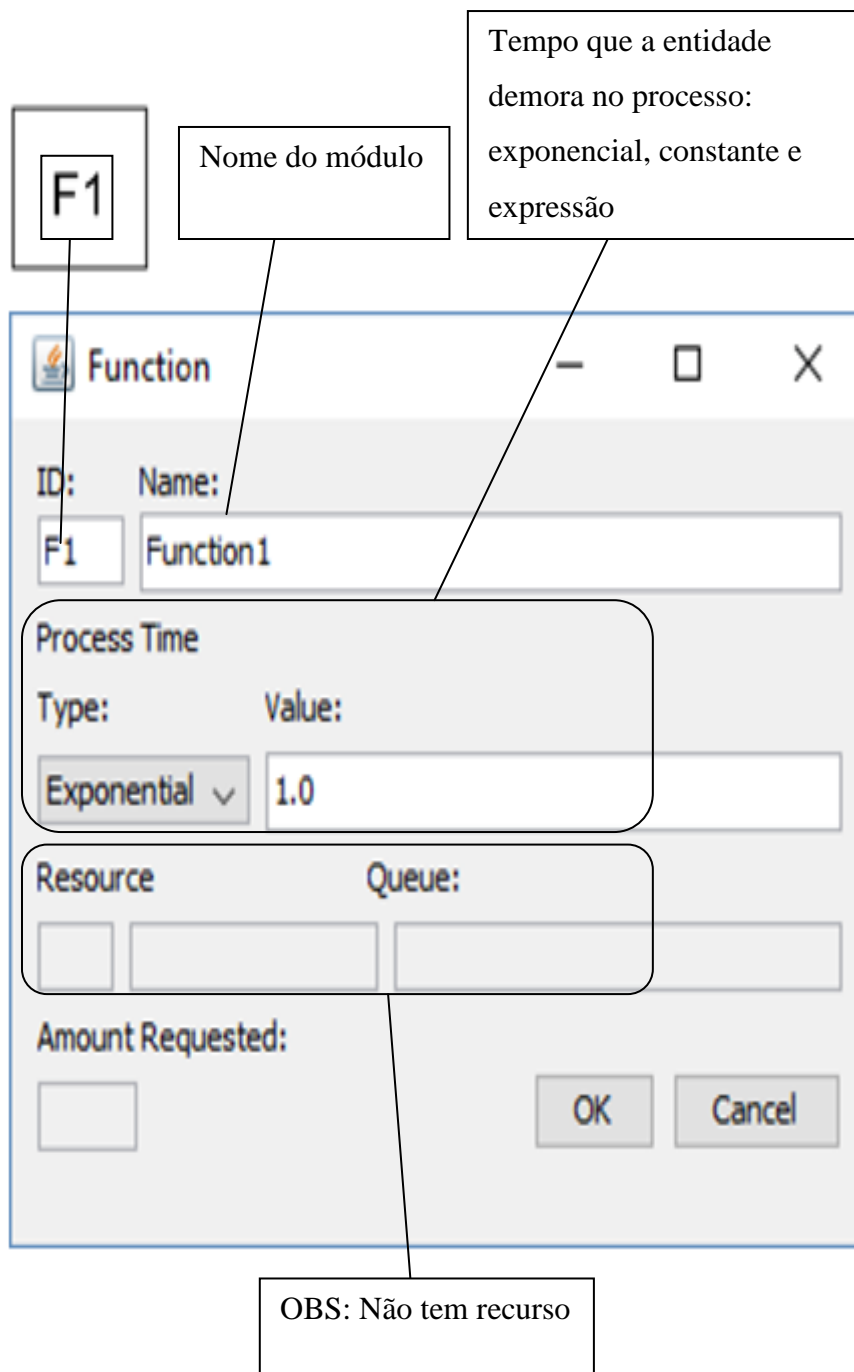


Figura 8: Demonstração do módulo *Function* sem recurso.

Note que, para adicionar o recurso, deve-se clicar no elemento “recurso” e arrastá-lo até a função. A Figura 9 apresenta o módulo *Function* com recurso.

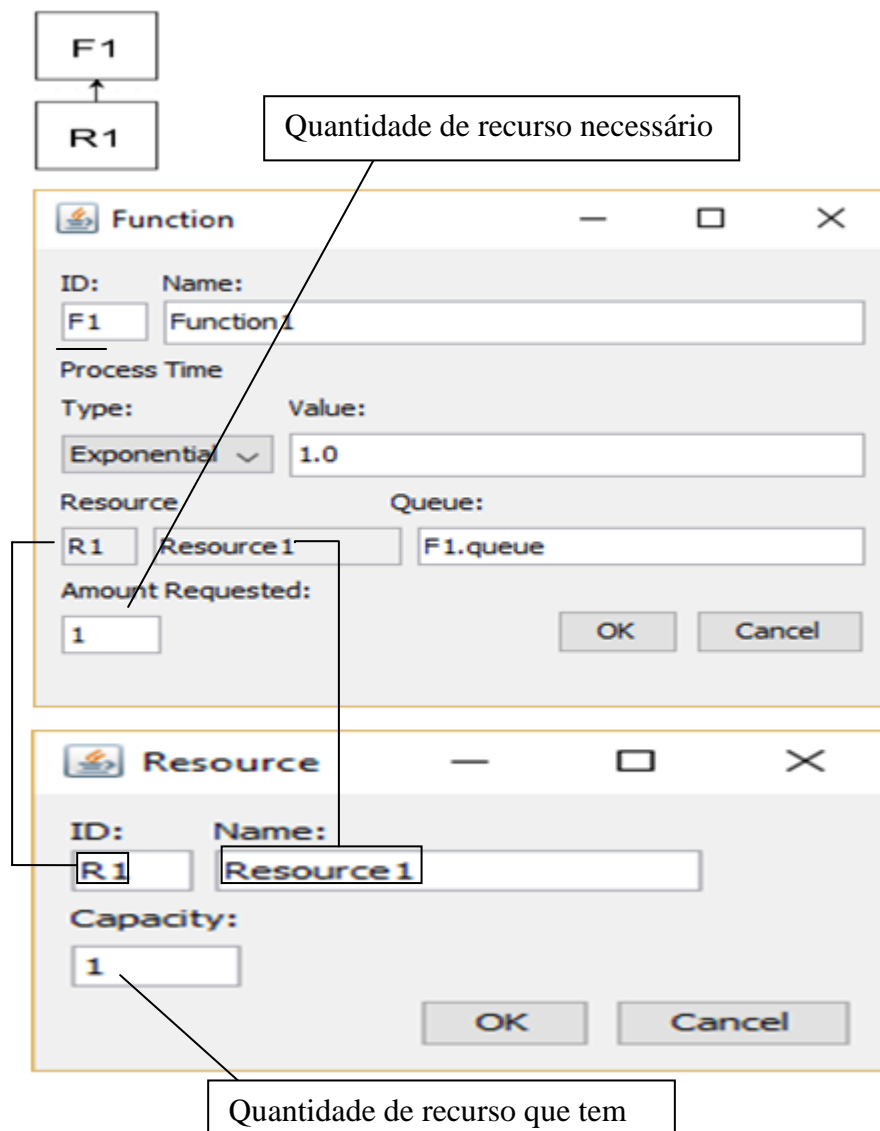


Figura 9: Demonstração do módulo *Function* com recurso.

Assim que for adicionado o recurso, é necessário informar o nome, quantos são requeridos e a capacidade deste.

2.4. Módulo *Decision Maker*: É o responsável por desviar a execução dos comandos de acordo com uma condição que foi submetida. Existem dois tipos: por chance e por condição. Caso seja por chance, deverá atribuir-se quantos % será verdadeiro na linha que antecede o fluxo verdadeiro e atribuir-se quantos % será falso na linha que antecede o fluxo falso. Se for por condição, deve-se atribuir uma expressão, se for verdadeiro, segue para o ramo verdade, se não, segue para o ramo falso. A ligação desse módulo com outro deve ser feita com o “*connector*”, encontrado na parte de elementos de modelagem, já que com esse é possível editar a linha. A Figura 10 mostra o exemplo do módulo *Decision Maker*, a Figura 11 demonstra um exemplo de como usar e na Figura 12 e 13 trata-se da edição das linhas horizontal e vertical, respectivamente.

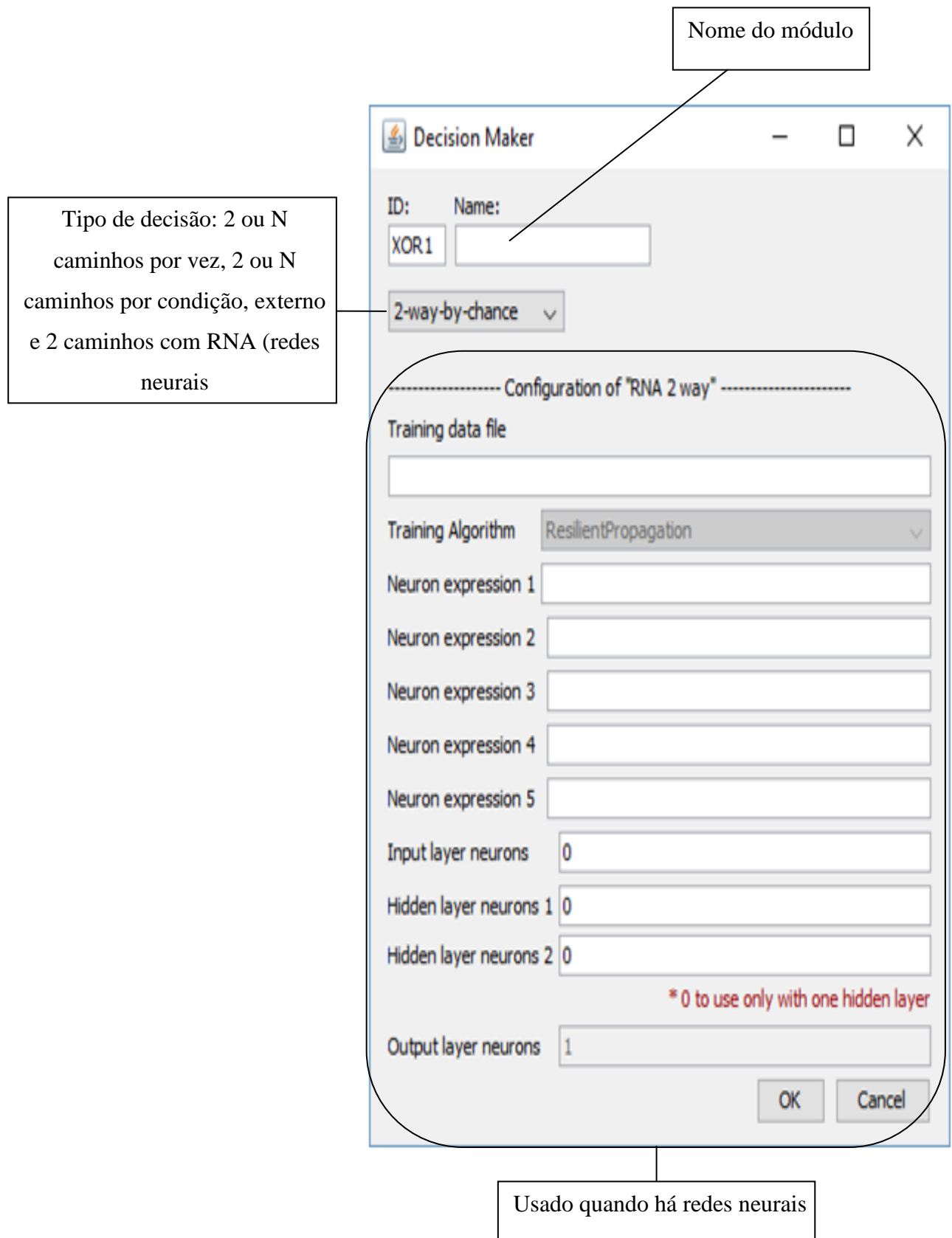


Figura 10: Demonstração do módulo *Decision Maker*.

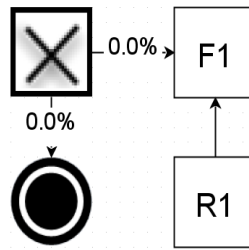


Figura 11: Exemplo do módulo *Decision Maker*.

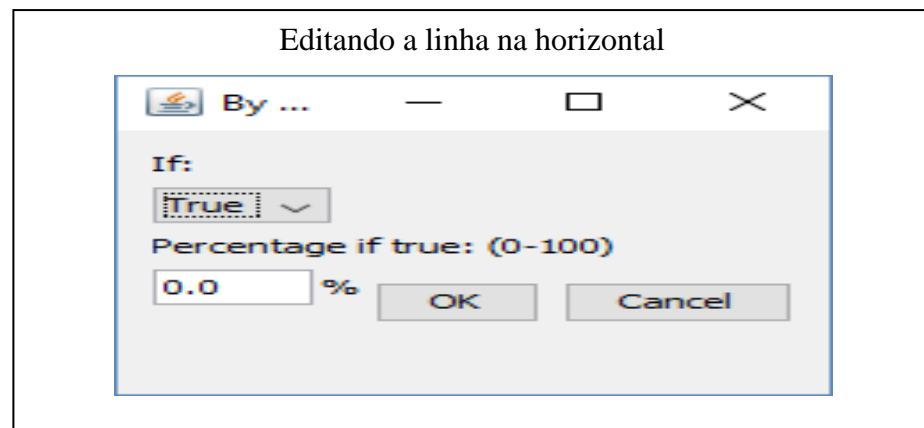


Figura 12: Edição da linha na horizontal.

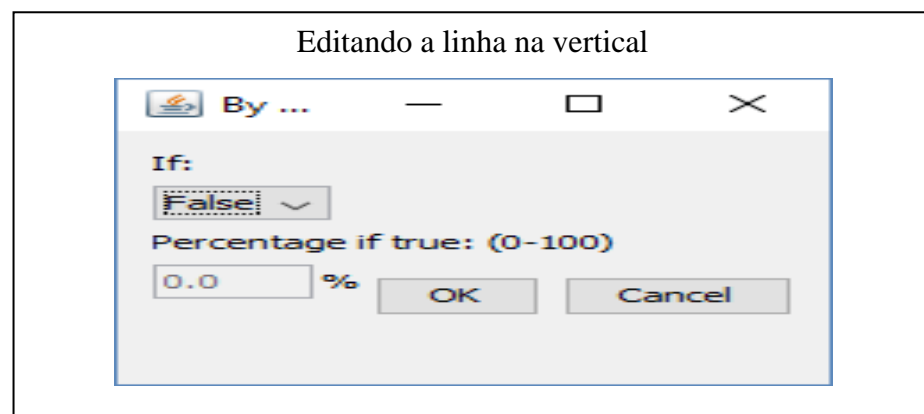


Figura 13: Edição da linha na vertical.

2.5. Módulo *Batch*: É o módulo utilizado para criar agrupamentos de entidades. As entidades que chegam neste módulo formam uma fila até que chegue à quantidade que haverá no lote, transformando-as em uma única entidade. A Figura 14 demonstra o módulo *Batch*.

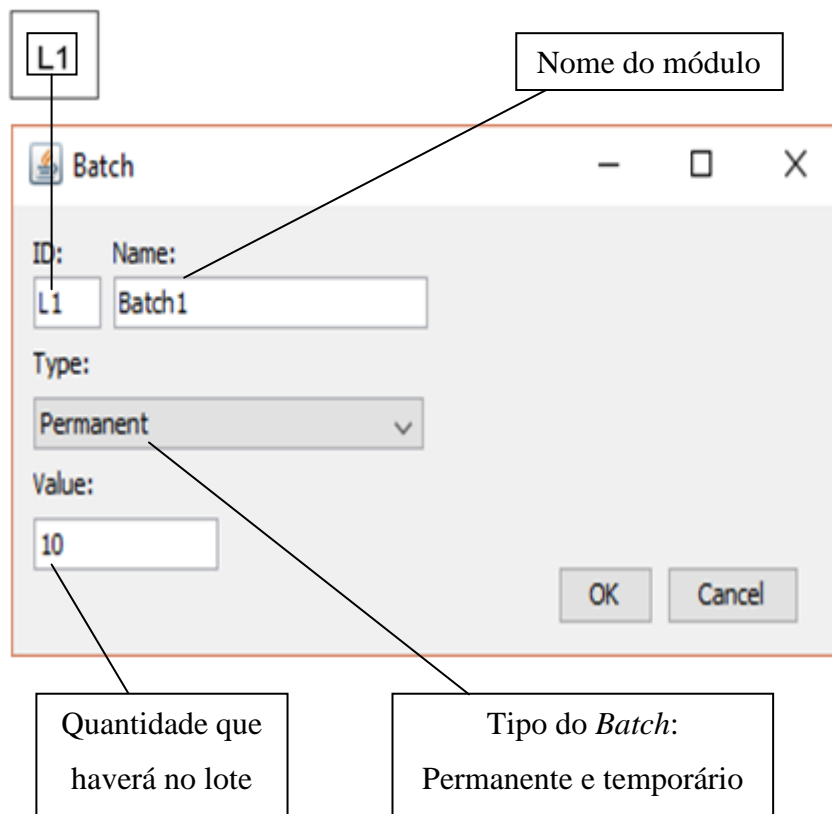


Figura 14: Demonstração do módulo *Batch*.

Este lote pode ser temporário ou permanente. Quando temporário, pode ser desfeito em seguida pelo *Separate*. Quando permanente, apenas a entidade-lote continuará no processo, ou seja, após o *batch*, não voltará a separar o lote.

2.6. Módulo *Separate*: Neste comando, é desfeito o agrupamento do lote temporário feito pelo comando *Batch*. A Figura 15 demonstra o módulo *Separate*.

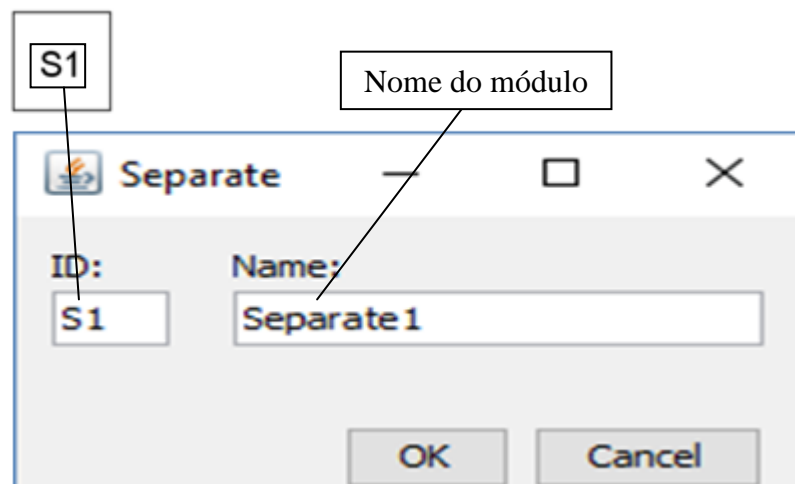


Figura 15: Demonstração do módulo *Separate*.

2.7. Módulo *Record*: Este comando tem a função de coletar informações do modelo, como: intervalo de tempo, contagem e tempo entre chegadas. A Figura 16 apresenta a demonstração do módulo *Record*.

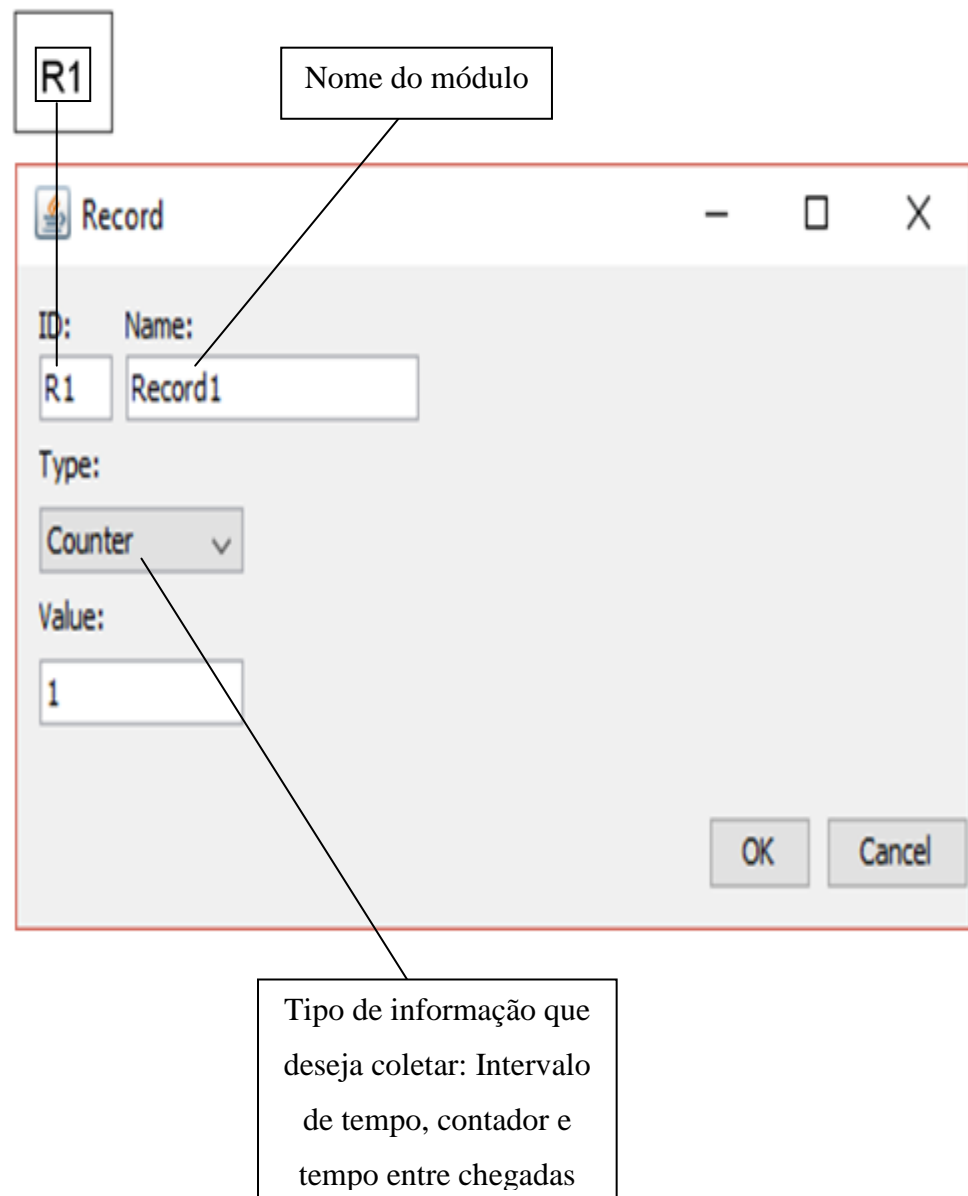


Figura 16: Demonstração do módulo *Record*.

2.8. Módulo *Assign*: Este módulo serve para adicionar um atributo, variável ou *tag* (booleana, inteira e real) no modelo. O valor do *Assign* é um número real, que foi gerado a partir da avaliação de uma expressão especificada no comando. A Figura 17 apresenta a demonstração do módulo *Assign*.

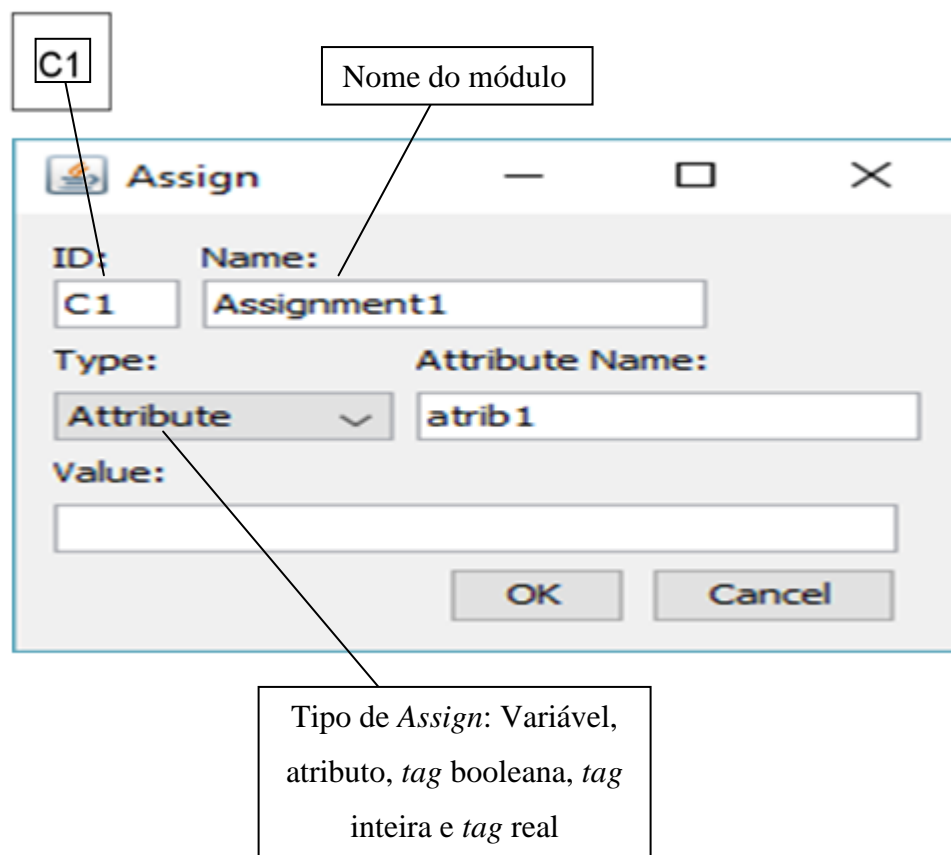


Figura 17: Demonstração do módulo *Assign*.

2.9. Módulo *Hold*: Este comando segura as entidades na fila até que a condição estabelecida seja satisfeita. A Figura 18 apresenta a demonstração do módulo *Hold*.

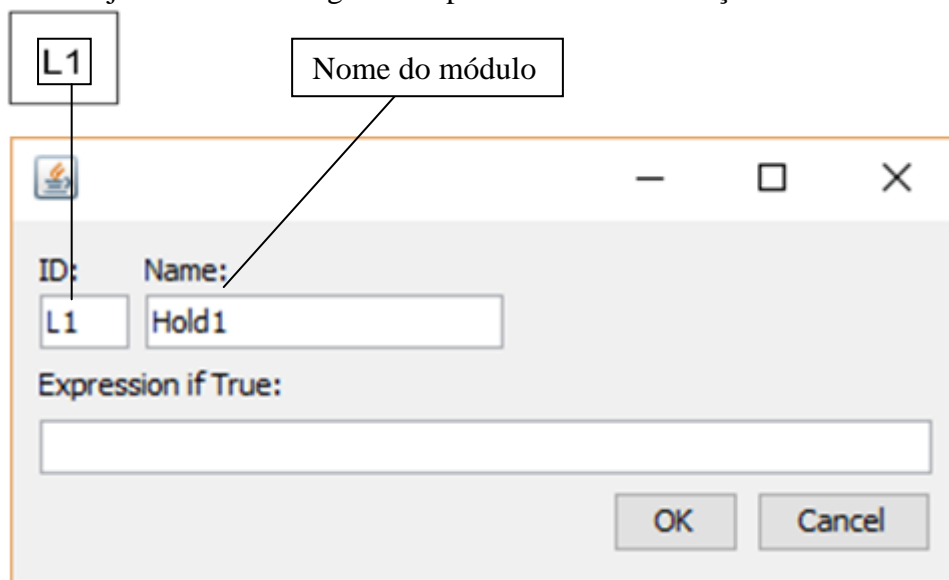


Figura 18: Demonstração do módulo *Hold*.

2.10. Módulo *Jump*: Desvia o processamento de comandos para outro comando. É passado um índice que simboliza a posição de onde está o comando. Deve-se colocar dois elementos deste comando, um sendo o original e o outro não. A Figura 19 apresenta a demonstração do módulo *Jump*.

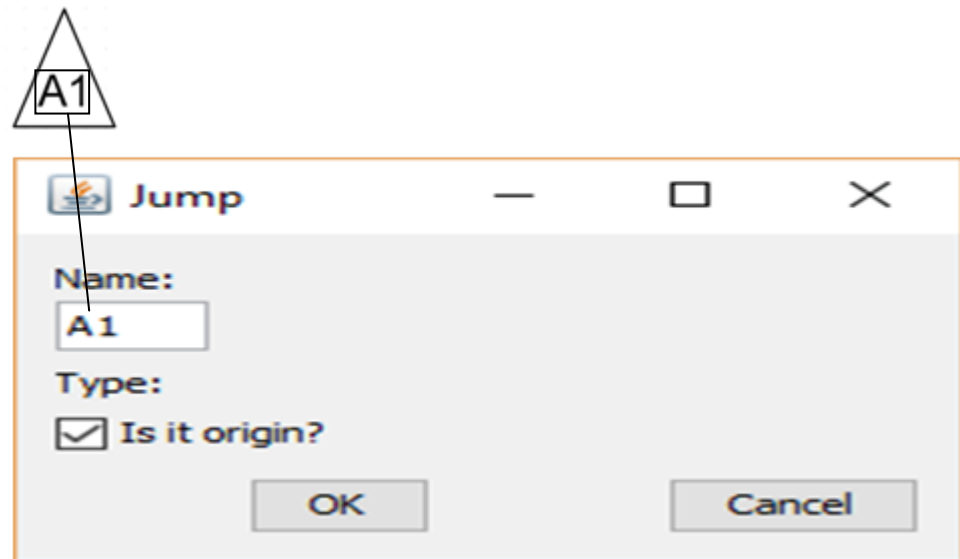


Figura 19: Demonstração do módulo *Jump*.

A Figura 20 exemplifica um processo bem simples para entender a utilização do *Jump*. O *Jump* que está na frente do *Create* é o original e aquele antes do *Function* não é original.

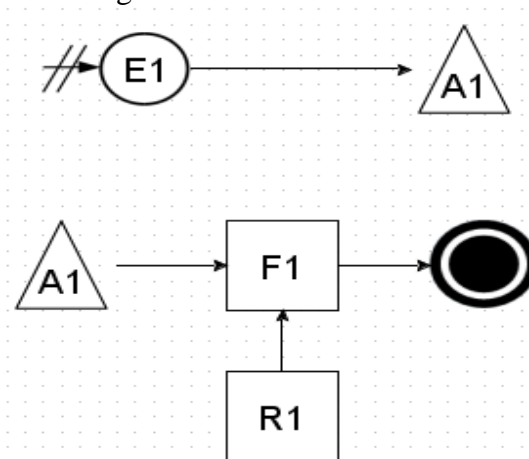


Figura 20: Exemplo do módulo *Jump*.