

O que é o ArgoUML?

O ArgoUML é uma aplicação open source que usa UML para modelar o desenho de softwares. Ele roda na maior parte das plataformas por ser implementado em Java. Ele dá suporte para todos os tipos de diagramas UML padrão e inclui suporte cognitivo.

A *Software Development Magazine* realiza uma premiação anual entre ferramentas populares de desenvolvimento de *software* em várias categorias. Em 2003 o ArgoUML foi um dos finalistas na categoria "Ferramentas de *Design* e Análises". Ele recebeu um prêmio de revelação, derrotando várias ferramentas comerciais.

Até a versão 0.20, o ArgoUML não tinha conformidade completa com o padrão UML e não oferecia suporte total a alguns tipos de diagramas, incluindo diagramas de seqüências. A partir daí até a versão mais recente, a 0.24, ele ganhou novas características como:

- Todos os 9 diagramas do UML 1.4 suportados;
- Uma plataforma independente: Java 5+;
- O metamodelo padrão UML 1.4;
- Suporte ao XMI;
- Exportação de diagramas nos formatos GIF, PNG, PS, EPS, PGML e SVG;
- Disponível em 10 idiomas, incluindo português;
- Um avançado editor de diagramas e ferramenta zoom;
- Suporte ao OCL;
- Engenharia reversa.

Dentre outras características.

No entanto, o ArgoUML tem alguns pontos fracos como:

- Incompatibilidade entre versões;
- Não há auto-chamadas em diagramas de seqüências, você precisa criar novamente os atores no diagrama;
- Ausência do “desfazer”, tornando a edição dispendiosa, uma vez que se cometa algum erro terá que deletar e criar novamente o objeto.

Ainda assim o ArgoUML pode ser utilizado para criar seus diagramas de uma forma fácil. Sendo uma ferramenta CASE (do inglês: Computer-Aided Software Engineering, uma classificação que abrange todas as ferramentas baseadas em computadores que auxiliam atividades de engenharia de software), trabalhar nele faz com que a pessoa crie seus diagramas seguindo os padrões do UML, não permitindo ir contra seus princípios.

Instalando o ArgoUML

O ArgoUML é escrito totalmente em Java e utiliza a Fundação de classes Java. Permitindo que ele rode virtualmente em qualquer plataforma com Java5 ou Java6.

Devido a isso, você precisará ter um sistema operacional que suporte Java, 10 MB de espaço livre em disco e o Java 2 JRE ou JDK versão 1.4 ou superior.

Para utilizar seu ArgoUML sem precisar conectar-se com a página oficial dele através do Java Web Start, neste curso vamos ensinar apenas como instalá-lo através da distribuição binária.

Primeiramente certifique-se de ter ao menos um Java 2 JRE ou JDK instalado, você pode fazer isso no terminal de comando digitando:

```
aptitude install sun-java5-jdk
```

que já será verificado se está instalado e se não estiver ele o instalará.

Após isso, baixe a distribuição binária do ArgoUML na seguinte página:

<http://argouml-downloads.tigris.org/argouml-0.24/>

e baixe o arquivo ArgoUML-0.24.tar.gz. Depois de baixar o arquivo, crie uma pasta onde você poderá guardar todos os arquivos do ArgoUML, coloque o arquivo .gz nessa pasta e extraia seu conteúdo com o comando:

```
tar -zxvf ArgoUML-0.24.tar.gz
```

E sem mais rodeios, já podemos utilizá-lo!

Para isso digite o comando no terminal:

```
java -jar argouml.jar
```

que ele carregará e abrirá a seguinte janela:

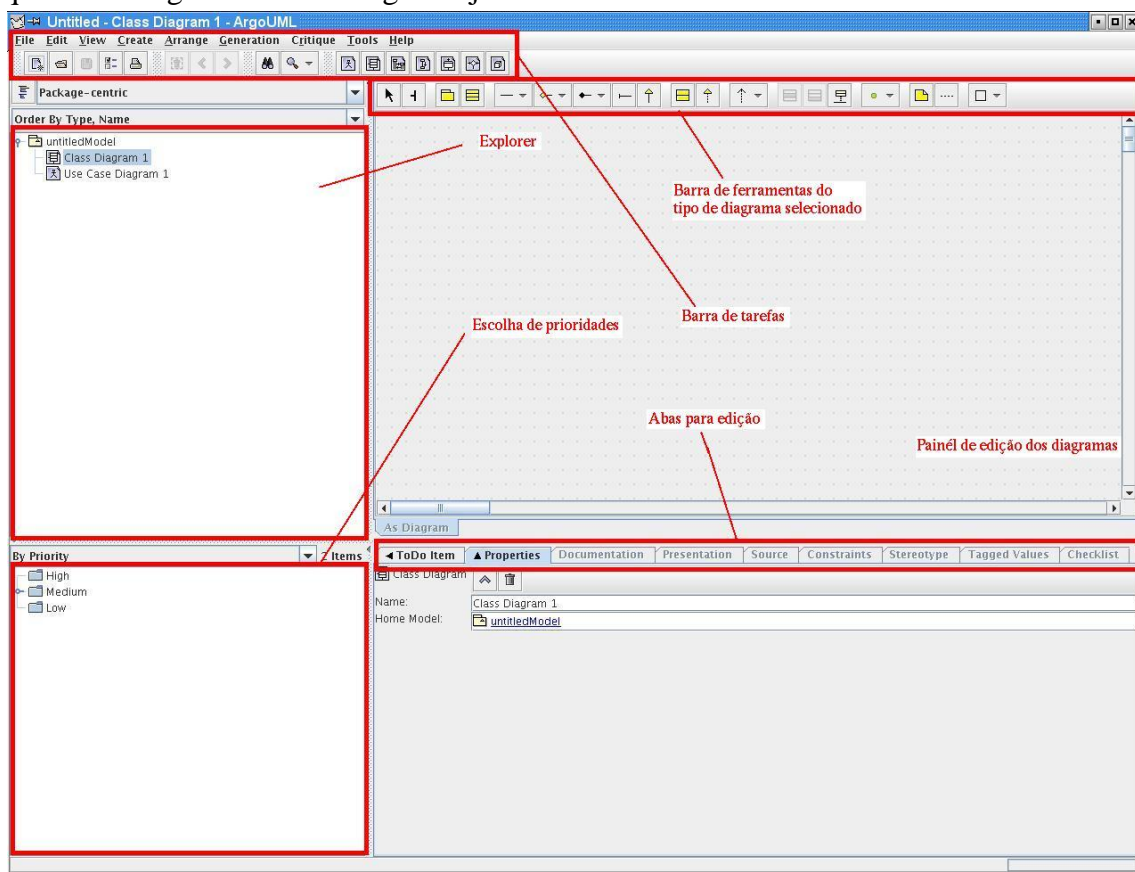
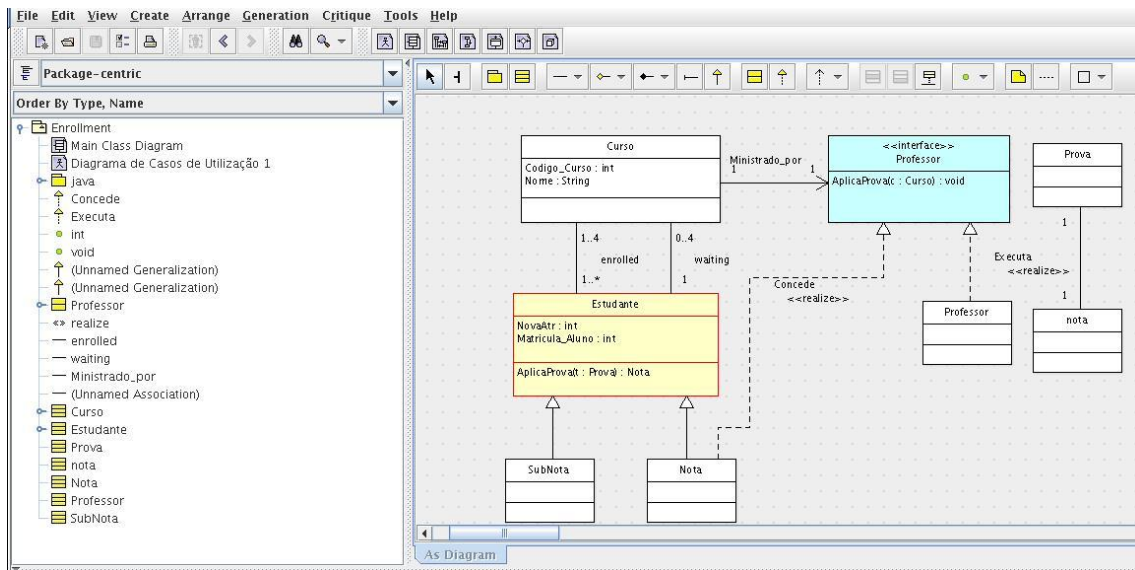


Diagrama de Classes

Agora vamos ver como se constrói um diagrama de classes utilizando a ferramenta CASE ArgoUML.

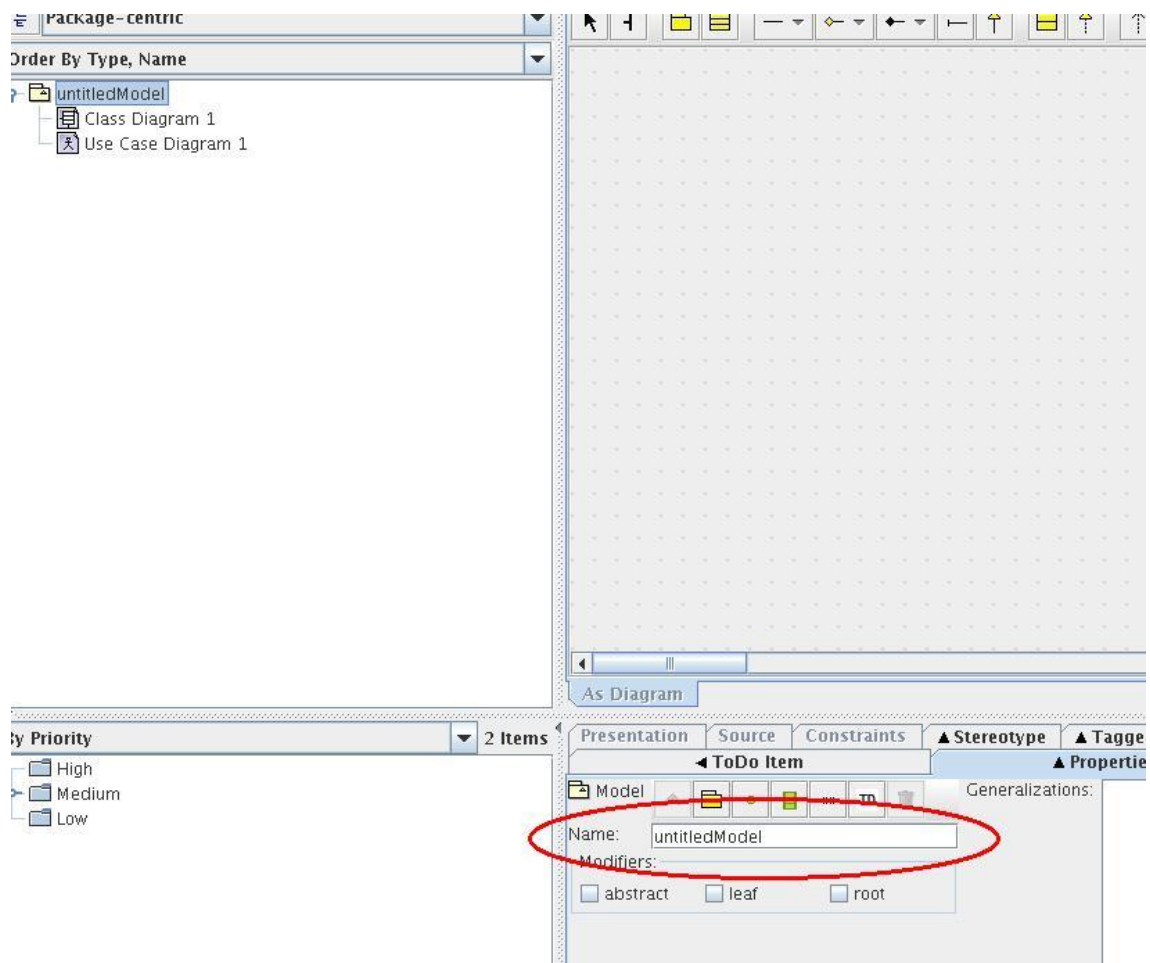
Chamaremos o projeto de “Matricula”. A **figura 1** mostra mais ou menos como ficará o resultado final:




Se você não estiver com um documento em branco, clique em File > New, para criar um novo projeto.

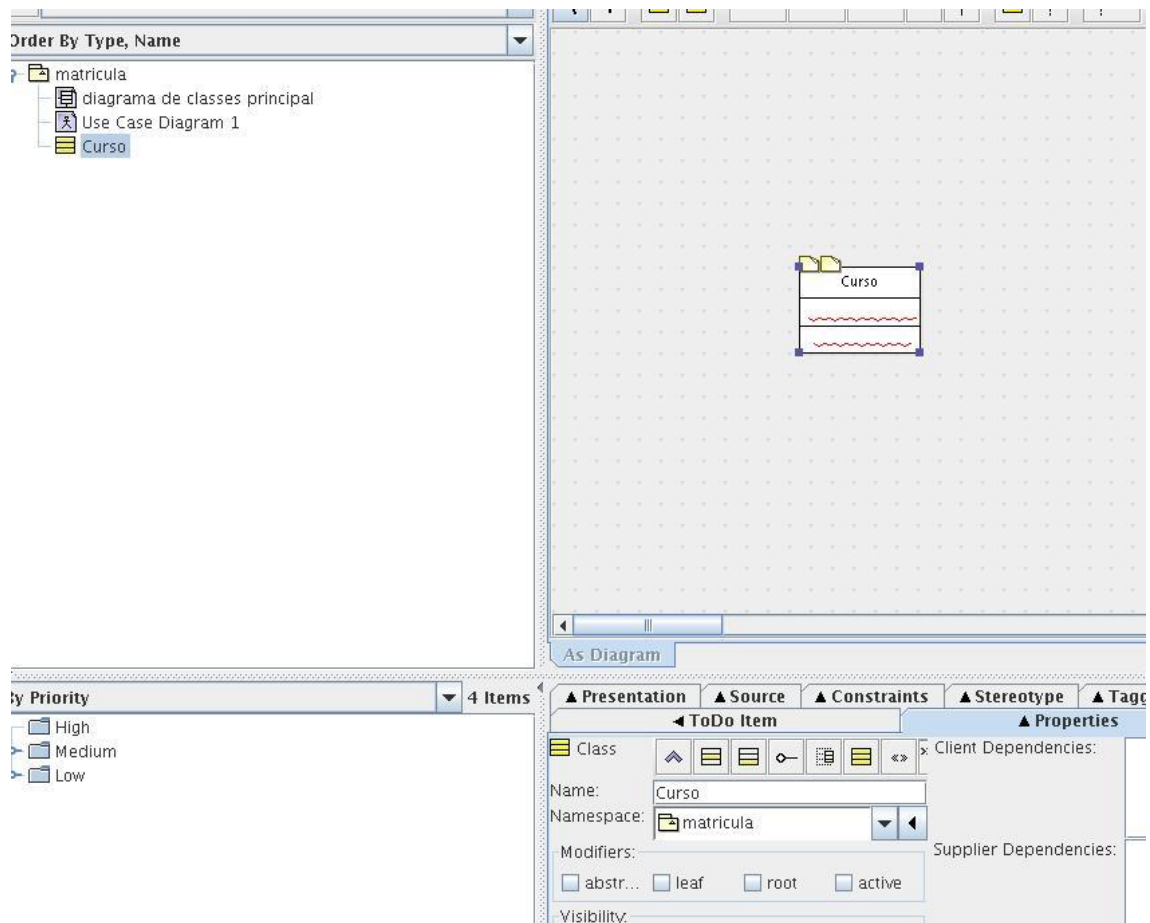
Agora que temos um projeto em branco, vamos dar um nome a ele (o nome Untitled vem como padrão). Para nomeá-lo, vá em File > Save Project As..., e salve com o nome “Matricula”. Ele salvará no formato .zargo, que pode ser aberto facilmente pelo ArgoUML com todo o seu projeto.


Para nomear seu modelo, ou seja, o pacote principal, clique no ícone do modelo no Explorer à esquerda que por padrão tem o nome “untitledModel”, em seguida clique na aba Properties na parte inferior direita. Renomeie para “matricula” no campo “Name:” conforme a **figura 2**:

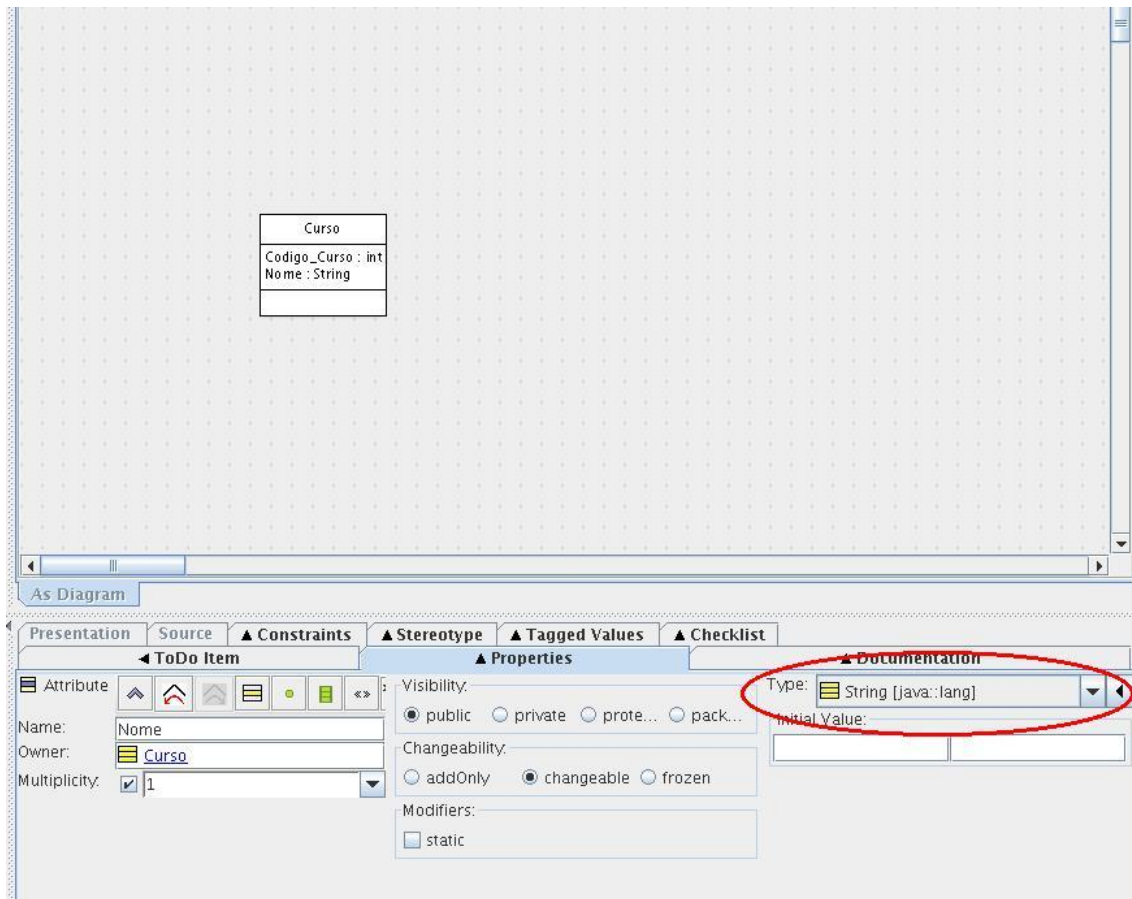


Utilize o mesmo procedimento para renomear o “Class Diagram 1” para “diagrama de classes principal”.

Para adicionar uma nova classe ao diagrama, clique no ícone New Class  na barra de ferramentas, e clique em qualquer parte do painel de edição (a parte cheia de pontos). Para nomear a nova classe, clique na classe e na aba Properties dê a ela o nome “Curso” no campo “Name:” assim como anteriormente. Observe que o Explorer à esquerda é atualizado automaticamente. Veja a **figura 3** como seu projeto deve ficar:

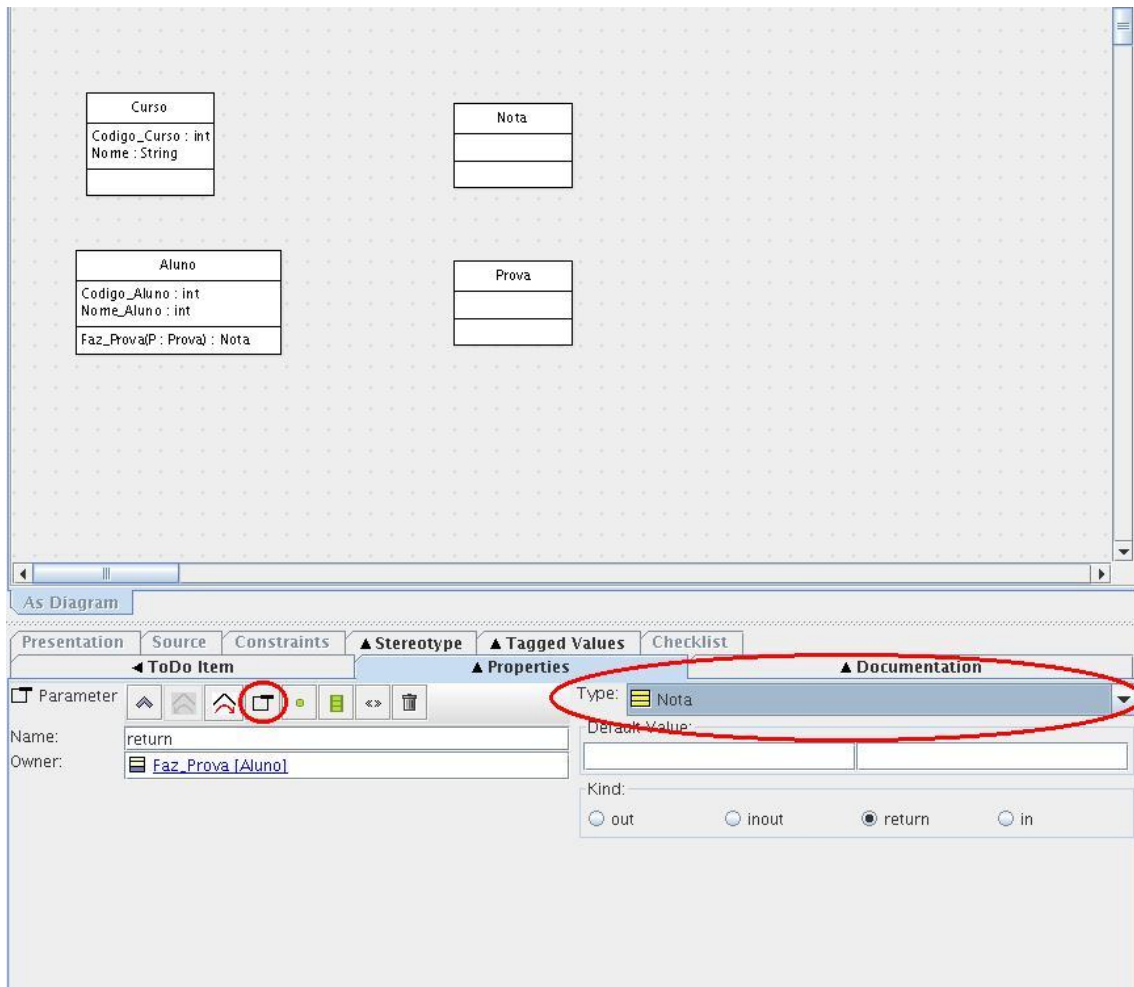


Para adicionar um novo atributo à classe Curso, clique no ícone “New Attribute” . Para renomear o atributo, você pode clicar no nome dele do diagrama (newAttr : int) e renomeá-lo no campo “Name:” e editá-lo na aba Properties. Nomeie-o como “Codigo_Curso”. Em seguida adicione um novo atributo na mesma classe com o nome “Nome”. Mude o tipo do segundo atributo criado para “String” na parte esquerda da aba Properties, em “Type”. Observe a **figura 4**:



Note que os novos atributos aparecem no Explorer à esquerda. Se precisar, você pode abrir a árvore clicando duas vezes nos ícones dos atributos.

Agora adicione três novas classes perto da classe Curso, uma chamada “Aluno”, uma chamada “Nota” e a outra chamada “Prova” assim como foi feito para a classe Curso. E adicione 2 atributos na classe Aluno chamando-os de “Codigo_Aluno” e “Nome_Aluno”(que deve ser String). Também na classe Aluno, adicione uma operação clicando no ícone “New Operation” ao lado do ícone de novo atributo, chamando-a de “Faz_Prova”. Os parâmetros devem ser adicionados clicando no ícone “New Parameter” logo acima do campo “Name:” da operação. Chame esse parâmetro de “P” e mude o tipo dele para “Prova”. Agora clique novamente no nome da operação Faz_Prova para ter acesso a todos os parâmetros da mesma, e clique 2 vezes em “return:void” para mudar o tipo de retorno da operação para “Nota”. Observe a **figura 5** para ver como deve estar o seu diagrama:



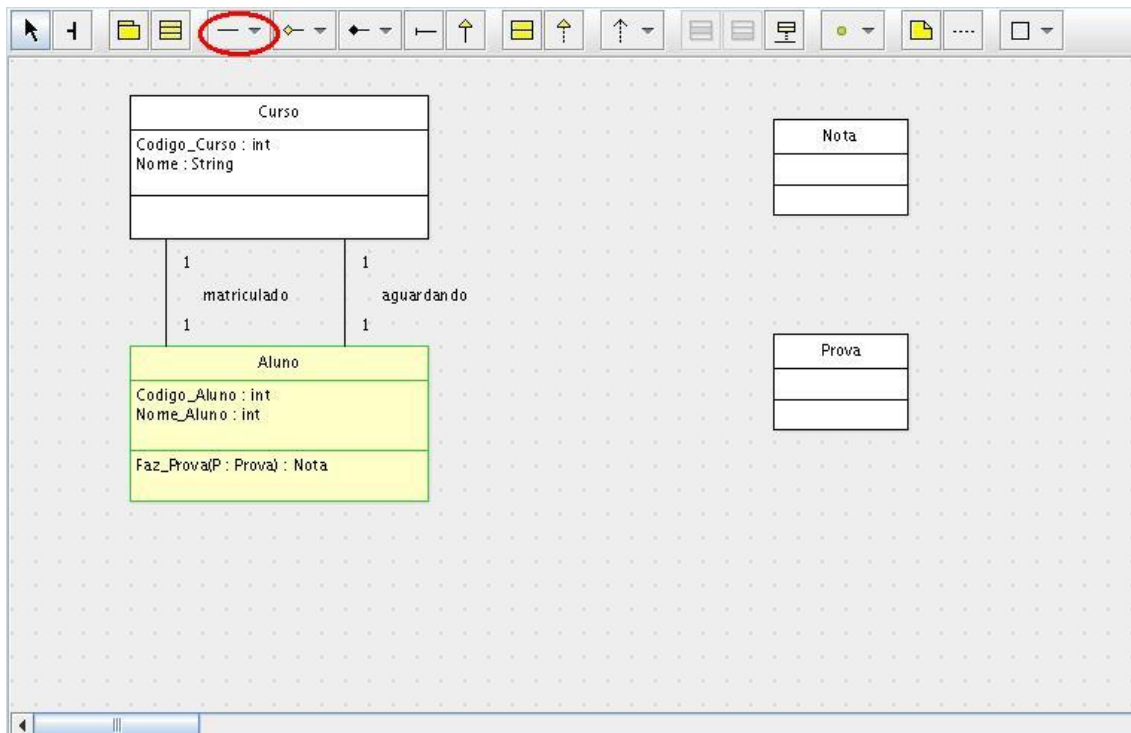
Agora vamos mudar a cor da classe Aluno, para isso clique na classe e em seguida na aba Presentation, nela você pode mudar a cor de fundo e da borda, escolha uma cor de seu gosto.

Vamos agora criar relacionamentos entre as classes.

Primeiro mova a classe Aluno para que fique abaixo da classe Curso e estique-as (basta clicar na classe e dimensioná-la como uma figura qualquer clicando em suas bordas).

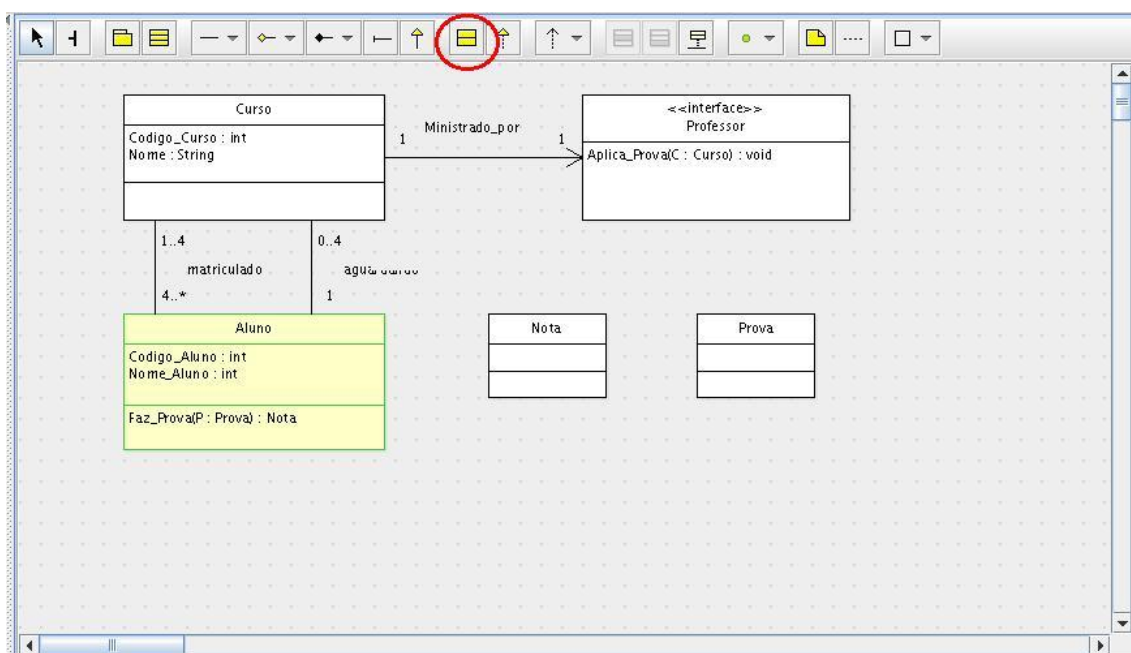
Para criar associações, clique no ícone New Association ao lado do ícone de nova classe, clique na classe Curso e arraste até a classe Aluno soltando o mouse, crie duas associações. Elas podem ser movidas e redimensionadas da mesma forma que as classes.

Você pode nomear as associações no campo "Name:" da aba Properties quando se clica na associação, chame uma de "matriculado" e a outra de "aguardando". Observe a **figura 6** para saber como o diagrama deve estar:

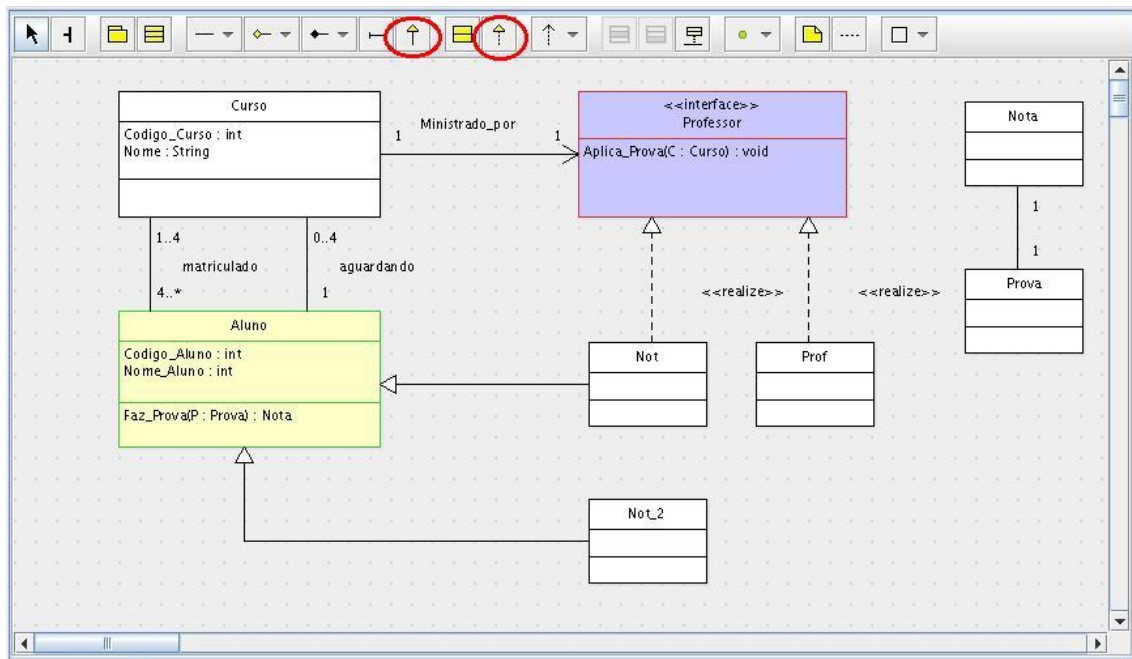


Clicando na associação com o botão direito, você pode definir a sua multiplicidade no menu Multiplicity. Para defini-las a seu gosto, selecione a associação “aguardando” por exemplo. Então, aba Properties, no campo “Connections” você pode escolher qualquer das extremidades clicando duas vezes. Daí você poderá digitar o valor que desejar para a multiplicidade ou também selecionar um, clicando na seta do campo Multiplicity.

Agora se você quiser que a classe Curso utilize uma interface, clique no ícone New Interface na barra de ferramentas e depois em qualquer parte do diagrama. Em seguida nomeie a interface como “Professor”, e crie uma operação chamada “Aplica_Prova” nessa interface. Ajuste os parâmetros de acordo com a **figura 7**. Então adicione uma associação entre a classe Curso e a nova interface e nomeie a associação como “Ministrado_por”. Repare que a associação será unidirecional seguindo os princípios do UML, já que estamos trabalhando com uma ferramenta CASE.



Para que tenhamos classes utilizando essa interface, vamos adicionar duas novas classes, “Not” e “Prof”. E vamos fazer uma realização dessas classes com a interface Professor. Em seguida também podemos fazer generalizações/heranças criando uma nova classe “Not_2” que herdará dados da classe Aluno. Observe a figura 8 para ver como pode ser seu diagrama:

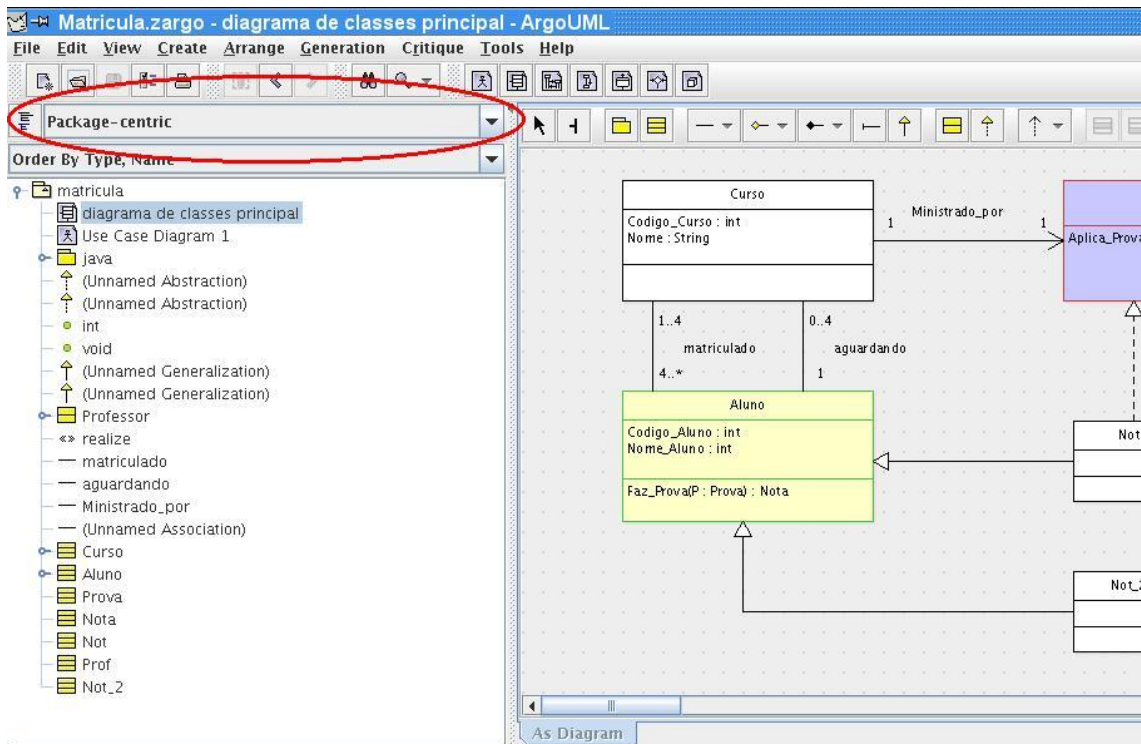


Agora você já está apto a criar seus diagramas, lembrando que as setas podem ser desenhadas a seu gosto clicando e arrastando-as para que assumam a forma que preferir. E também todo o diagrama pode ser movido apertando ctrl+a e arrastando-o.

Diagrama de Casos de Uso

Agora que já temos um Diagrama de Classes, vamos aprender a fazer um Diagrama de Casos de Uso e um Diagrama de Estados.

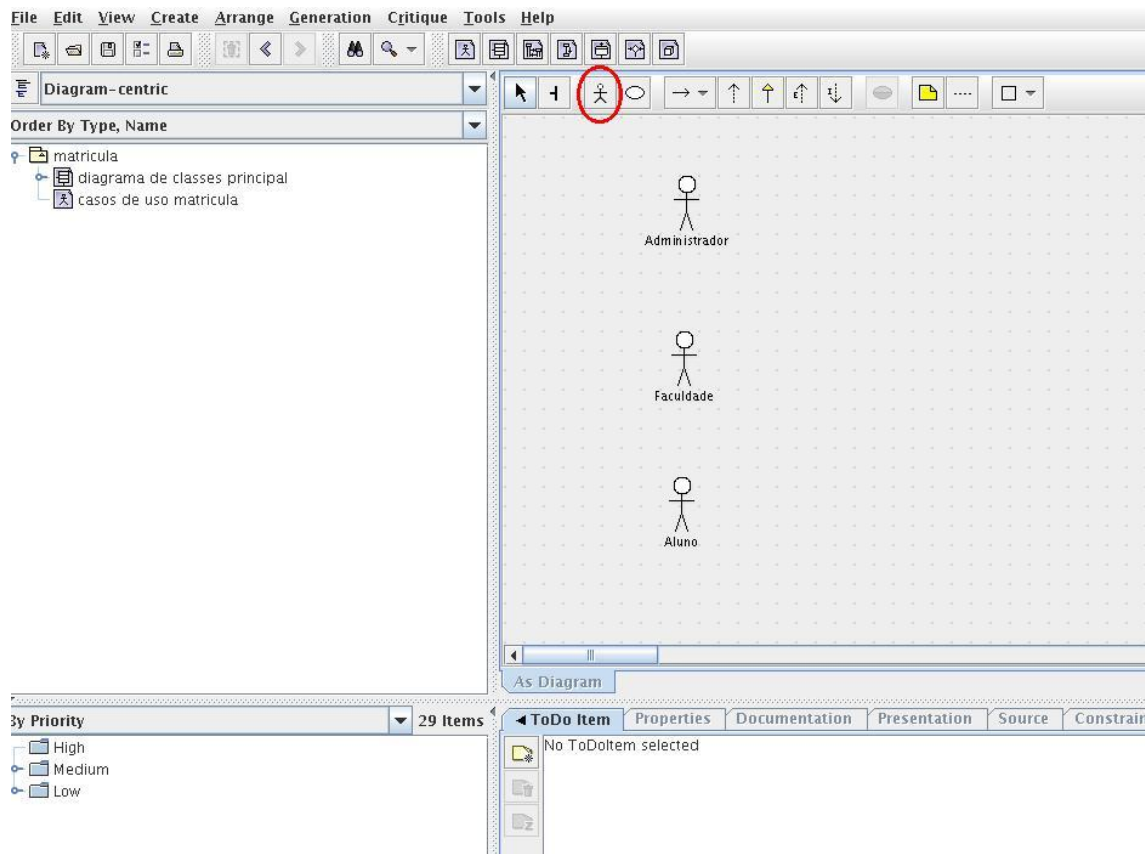
Se não estiver com seu projeto aberto, abra-o com o ArgoUML e mude o modo de exibição do Explorer da esquerda de “Package-centric” para “Diagram-centric” como na figura 9, isso serve para visualizar melhor a área em que você está trabalhando.



Agora clique duas vezes em “Use Case Diagram 1” no Explorer para exibir o painel de edição de diagramas de caso de uso com uma nova barra de ferramentas.

Vamos renomear o diagrama como “casos de uso matricula” como fizemos pra nomear tudo até agora.

Agora vamos adicionar como exemplo, 3 atores clicando duas vezes no ícone “New Actor” na barra de ferramentas. Clicar duas vezes faz com que você possa ir adicionando o objeto escolhido quantas vezes quiser sem ter que clicar novamente no ícone. Para encerrar, basta clicar novamente no ícone que escolheu inicialmente. Em seguida vamos nomear estes atores como “Administrador”, “Faculdade” e “Aluno”. Observe a figura 10 para saber como está indo o diagrama:



Agora vamos adicionar 2 novos casos de uso clicando duas vezes no ícone “New Use Case” e chamá-los de “Abrir um Curso” e “Fechar um Curso”. Uma ferramenta que auxilia na apresentação do diagrama é a “Broom”, que é uma espécie de régua. Clique no ícone dela na barra de ferramentas logo ao lado do ícone de novo ator, clique e arraste o mouse horizontalmente para um alinhamento vertical, ou verticalmente para um horizontal. Você pode esticar o tamanho da régua após escolher a forma de alinhamento, experimente, é bem fácil!

Em seguida vamos criar associações entre os atores e esses casos de uso clicando duas vezes no ícone “New Association”, selecionando um ator e arrastando para o caso de uso. Podemos nomear estas associações como fizemos no diagrama de classes. Observe na figura 11 como ficou o nosso diagrama de casos de uso:

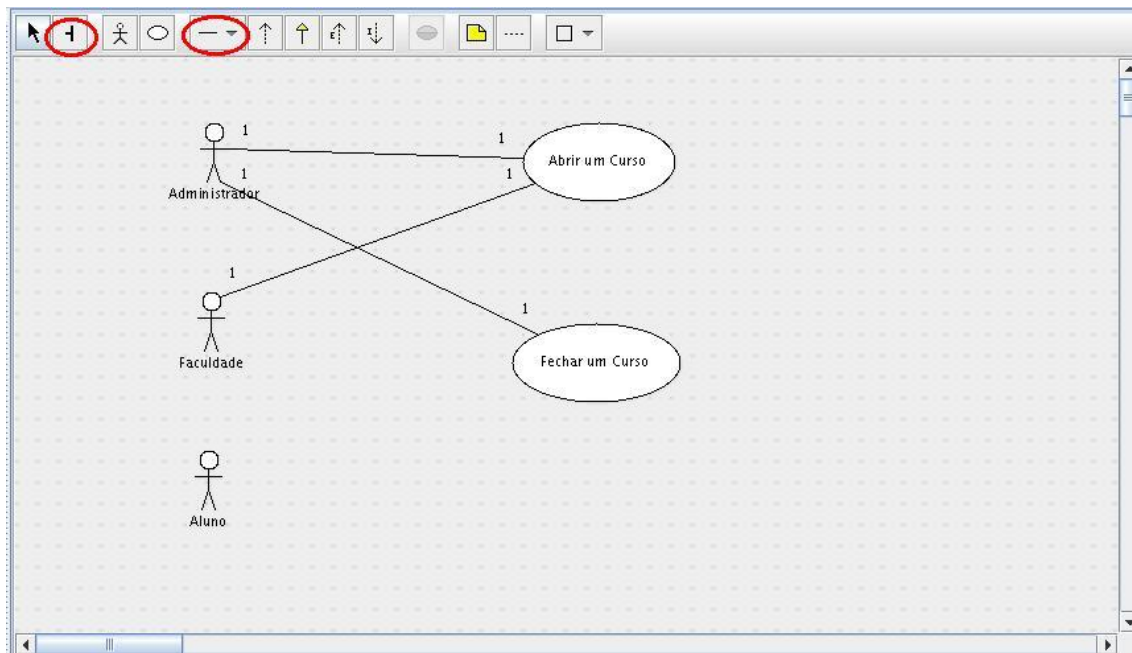
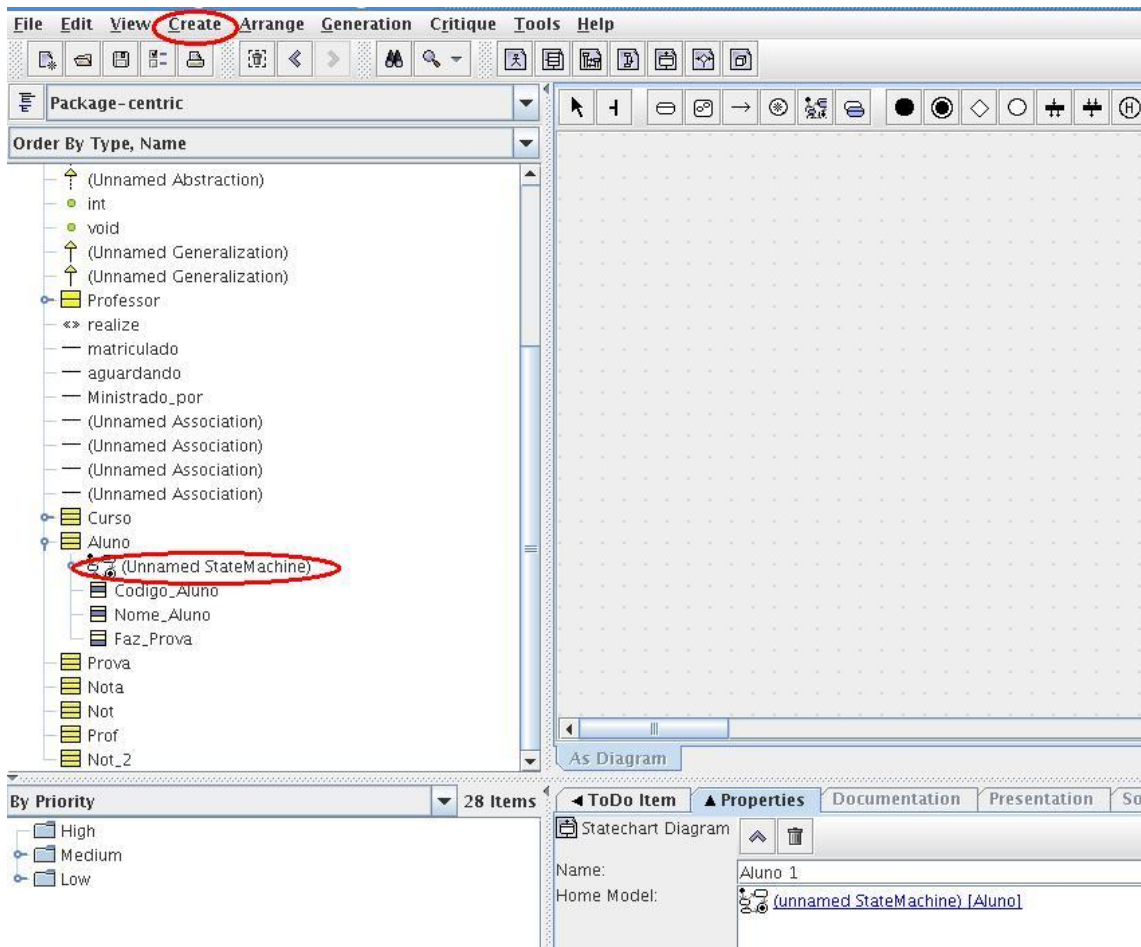


Diagrama de Estados

E aproveitando a facilidade que é fazer um diagrama de casos de uso, vamos fechar o curso vendo como construir um diagrama de estados.

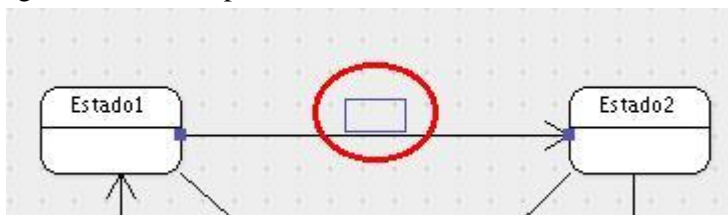
Cada diagrama de estado é relacionado a uma classe, portanto vamos clicar no diagrama de classes principal no Explorer e escolher uma classe para a criação do diagrama. Mas antes de escolher a classe, vamos mudar o modo de visualização do Explorer para que possamos enxergar o diagrama de estados. Vamos voltar de “Diagram-centric” para “Package-centric”.

Agora vamos escolher como exemplo a classe Aluno. Clique nela e na barra de tarefas do ArgoUML clique em Create > New_Startchart Diagram. Agora no Explorer, encontre a classe Aluno e abra a árvore dele para encontrar o diagrama “Unnamed Statemachine”. Observe a figura 12:

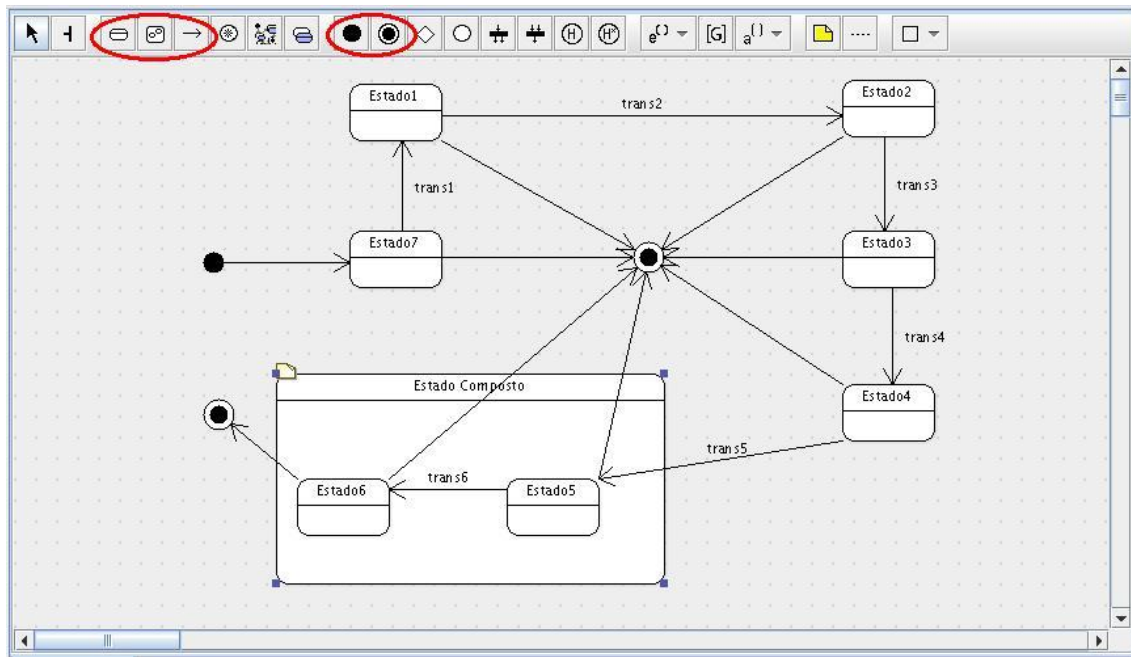


Clique nele e vamos chamá-lo de “Aluno Estados”. Após nomeado, vamos mudar o modo de exibição do Explorer para “State-centric”.

Agora vamos adicionar estados iniciais e finais clicando nos ícones “New Initial” e “New Final State”, respectivamente, na nova barra de ferramentas que apareceu. Crie como exemplo, um estado inicial e dois finais. Em seguida vamos criar 7 estados simples com o ícone “New Simple State”. Nomeie-os como quiser, depois vamos criar transições com o ícone “New Transition” podendo também nomeá-las clicando duas vezes no retângulo correspondente ao nome. Veja na figura 13 um exemplo:



Agora vamos criar um estado composto clicando no ícone “New Composite State” e vamos arrastar 2 estados simples para dentro dele. Observe na figura 14 um exemplo do diagrama de estados pronto:



Com essas instruções você já deve estar apto a criar seus diagramas sem problemas!

Lembrando que qualquer dúvida, basta colocar o mouse sobre o ícone para saber o nome dele.

Bom, agora que sabemos criar os diagramas, o que fazer com eles? Podemos transformar todos os diagramas ou apenas o diagrama selecionado em imagens para uma impressão simplificada indo em File > Export All Graphics ou Export Graphics, respectivamente.

Podemos também gerar o código do projeto indo em Generation > Generate All Classes marcando em qual linguagem deseja gerar o código e em seguida clicar em “Generate”.

No ArgoUML é possível também trabalhar de forma colaborativa. Pessoas conectadas com o mesmo projeto no ArgoUML podem construir diagramas juntas em tempo real com uma janela de chat. Mas como ensinar a fazer isso não era o objetivo deste curso, encerramos por aqui.