

UML

Unified Modeling Language

**Adelino Manuel de Oliveira Cabral
Lúcia Goretti Gonçalves de Araújo**

1. Introdução

O desenvolvimento de sistemas de software de grande porte são suportados por métodos de análise e projeto que modelam esse sistema de modo a fornecer para toda a equipe envolvida (cliente, analista, programador, etc.) uma compreensão única do projeto.

A UML (Unified Modeling Language) é o sucessor de um conjunto de métodos de análise e projeto orientados a objeto (OOA&D). A UML está, atualmente, em processo de padronização pela OMG (Object Management Group).

A UML é um modelo de linguagem, não um método. Um método pressupõe um modelo de linguagem e um processo. O modelo de linguagem é a notação que o método usa para descrever o projeto. O processo são os passos que devem ser seguidos para se construir o projeto.

O modelo de linguagem é uma parte muito importante do método. Corresponde ao ponto principal da comunicação. Se uma pessoa quer conversar sobre o projeto, como outra pessoa, é através do modelo de linguagem que elas se entendem. Nessa hora, o processo não é utilizado.

A UML define uma notação e um meta-modelo. A notação são todos os elementos de representação gráfica vistos no modelo (retângulo, setas, o texto, etc.), é a sintaxe do modelo de linguagem. A notação do diagrama de classe define a representação de itens e conceitos tais como: classe, associação e multiplicidade. Um meta-modelo é um diagrama de classe que define de maneira mais rigorosa a notação.

Neste trabalho, será apresentada a notação UML através de algumas definições e exemplos.

2. Comportamento do Sistema

O comportamento do sistema é capturado através de análise de casos de uso do sistema chamados “Use Case”.

Descrição informal do sistema automatizado de Matrícula num Curso

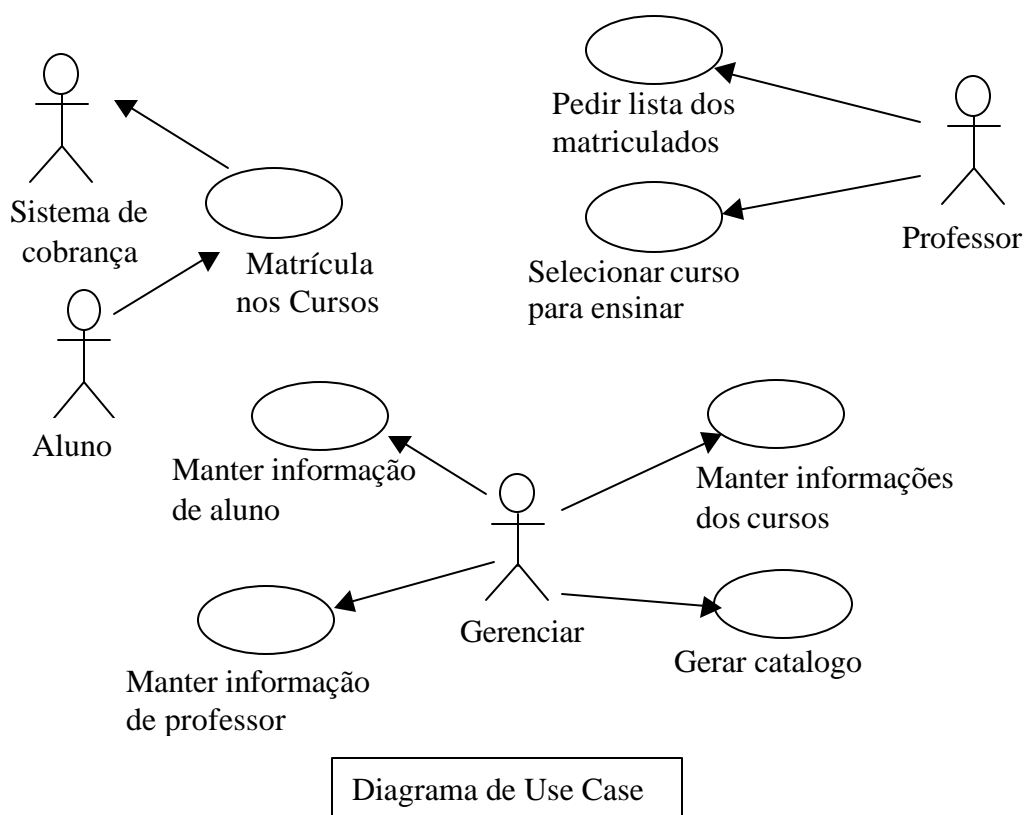
No início de cada semestre os alunos devem requisitar um catálogo de cursos contendo os cursos oferecidos no semestre. Este catálogo deve conter informações a respeito de cada curso tais como: professor, departamento e pré-requisitos. Desse modo, os alunos podem tomar suas decisões mais apropriadamente.

O novo sistema permitirá que os alunos selecionem quatro cursos oferecidos para o próximo semestre. Além disso, o aluno indicará dois cursos alternativos, caso o aluno não possa ser matriculado na primeira opção. Cada curso terá o máximo de 10 e o mínimo de 3 alunos. Um curso com número de alunos inferior a 3 será cancelado. Para cada matrícula feita por um aluno, o sistema envia informação ao sistema de cobrança para que cada aluno possa ser cobrado durante o semestre.

Os Professores devem acessar o sistema “on line”, indicando quais cursos irão lecionar. Eles também podem acessar o sistema para saber quais alunos estão matriculados em cada curso.

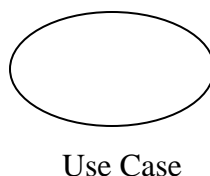
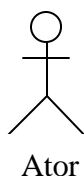
Em cada semestre, há um prazo para alteração de matrícula. Os alunos devem poder acessar o sistema durante esse período para adicionar ou cancelar cursos.

Diagrama de “Use Cases” para um sistema automatizado de Matrícula num Curso



Definição Diagrama de “Use Case”: É um diagrama usado para se identificar como o sistema se comporta em várias situações que podem ocorrer durante sua operação. Descreve o sistema, seu ambiente e a relação entre os dois. Os componentes deste diagrama são os atores e os “Use Case”.

A notação usada pelo Diagrama de “Use Case” é:



Ator: Representa qualquer entidade que interage com o sistema. Pode ser uma pessoa, outro sistema, etc. Algumas de suas características são descritas abaixo:

- ator não é parte do sistema. Representa os papéis que o usuário do sistema pode desempenhar.
- ator pode interagir ativamente com o sistema.
- ator pode ser um receptor passivo de informação.
- ator pode representar um ser humano, uma máquina ou outro sistema.

Use Case: Como foi exemplificado acima, é uma seqüência de ações que o sistema executa e produz um resultado de valor para o ator. Algumas de suas características são descritas abaixo:

- Um “Use Case” modela o diálogo entre atores e o sistema.
- Um “Use Case” é iniciado por um ator para invocar uma certa funcionalidade do sistema.
- Um “Use Case” é fluxo de eventos completo e consistente.
- O conjunto de todos os “Use Case” representa todas as situações possíveis de utilização do sistema.

Descrição textual resumida do “Use Case” “Matrícula nos Cursos”: Dos sete “Use Case” apresentados no Diagrama, será descrito apenas o de “*Matrícula nos Cursos*”. Este “Use Case” é iniciado pelo aluno. Fornece os meios para o aluno criar, anular, modificar e consultar o formulário de matrícula de um dado semestre.

Descrição do Fluxo principal de eventos associados a esse “Use Case”.

Este “Use Case” inicia-se quando o aluno fornece a chave de acesso. O sistema verifica se a chave de acesso do aluno é válida (E1) e apresenta ao aluno a opção de selecionar o semestre atual ou o próximo semestre (E2). O aluno seleciona o semestre desejado. O sistema pede ao aluno para selecionar a atividade desejada: Criar, Consultar, Modificar, Imprimir, Anular ou Sair do Sistema.

Se atividade selecionada é:

- Criar, o subfluxo A1 (Criar uma Matrícula Nova) é executado.
- Consultar, o subfluxo A2 (Consulta da Matrícula) é executado.
- Modificar, o subfluxo A3 (Modificação da Matrícula) é executado.
- Imprimir, o subfluxo A4 (Imprimir a Matrícula) é executado.
- Anular, o subfluxo A5 (Anular Matrícula) é executado.
- Sair, o “Use Case” é encerrado.

Descrição dos Subfluxos Alternativos associados a esse “Use Case”.

A1. Criar uma Matrícula Nova. O sistema apresenta numa tela um formulário de matrícula em branco. O aluno preenche-o com 4 cursos oferecidos, como primeira escolha e preenche 2 cursos oferecidos como segunda escolha (E3). A seguir, o aluno submete o formulário preenchido ao sistema. Para cada curso de primeira escolha, o sistema irá verificar se os pré-requisitos são satisfeitos (E4) e matricula o aluno no curso, se este estiver sendo oferecido, e se houver vaga (E5). O sistema imprime o formulário

de matrícula (E6) e envia a informação para ser processado pelo sistema de cobrança (E7). O sistema fica disponível.

A2.**Consulta da Matrícula**. O sistema recupera (E8) e apresenta numa tela as seguintes informações para todos os cursos nos quais o aluno está matriculado: nome do curso, número do curso, dias da semana, horário, localização e número de créditos. Quando o aluno indica que terminou a consulta, o sistema fica disponível.

A3.**Modificação da Matrícula**. O sistema verifica se a data limite para mudanças não expirou (E9). O sistema recupera (E8) e apresenta as seguintes informações para todos os cursos nos quais o aluno está matriculado: nome do curso, número do curso, dias da semana, horário, localização e número de créditos. O sistema oferece um menu com as seguintes opções: anule um curso oferecido, adicione um curso oferecido ou sair do sistema.

Se a atividade selecionada for:

- Anular um curso matriculado, o procedimento de anulação de um curso (A6), é executado.
- Adicionar um curso oferecido, o procedimento de adição de curso (A7) é executado.
- Sair do sistema, o sistema imprime formulário de matrícula (E6), envia a informação para ser processado pelo sistema de cobrança e fica disponível.

A4.**Imprimir a Matrícula**. O sistema imprime a matrícula do aluno (E6) e fica disponível.

A5.**Anular Matrícula**. O sistema recupera (E8) e apresenta as informações atuais da matrícula. O sistema pede ao usuário para confirmar a anulação da matrícula. Se efetuada, a matrícula é removida do sistema. Se a anulação não for confirmada, a operação é cancelada e o sistema fica disponível.

A6.**Anular um curso escolhido**. O aluno entra com o número do curso a ser anulado. O sistema pede ao usuário para confirmar a anulação do curso. Se efetuada, o curso é removido da matrícula do aluno. Se a anulação não for confirmada, a operação é cancelada e o sistema fica disponível.

A7.**Adicionar um curso**. O aluno entra com o número do curso a ser adicionado. O sistema verifica se os pré-requisitos são satisfeitos (E4) e adiciona o aluno ao curso, se o curso estiver sendo oferecido (E5) e o sistema fica disponível.

Descrição dos Subfluxos de Exceção.

E1.:**O aluno fornece chave de acesso inválida**. O aluno pode entrar com a chave de acesso novamente ou sair do sistema.

E2.**O aluno fornece um semestre inválido**. O aluno pode fornecer novamente um semestre ou sair do sistema.

- E3. **O aluno fornece número de curso inválido (formato).** O aluno pode fornecer outro número ou sair do sistema.
- E4. **O aluno não satisfaz todos os requisitos necessários.** O aluno é informado que não pode se matricular nesse curso e a razão para tal. Se possível, um curso alternativo é apresentado. O sistema segue adiante.
- E5. **O aluno é informado de que a matrícula para curso selecionado está encerrada.** Se possível, um curso alternativo é apresentado. O sistema segue adiante.
- E6. **A matrícula não pode ser imprimida.** A informação é armazenada e o aluno é informado de que o pedido de impressão deve ser repetido. O sistema segue a diante.
- E7. **O sistema armazenará todas as informações necessárias ao sistema de cobrança e a fornecerá assim que possível.** O fluxo segue adiante.
- E8. **O sistema não pode recuperar as informações de matrícula.** O aluno deve reiniciar o fluxo desde o início.
- E9. **O sistema informa ao aluno que a matrícula não pode ser alterada.** O aluno deve reiniciar o fluxo desde o início.

Documentação de um “Use Case”. Como apresentado acima, a documentação de um “Use Case” é composta de uma Descrição textual resumida e dos Fluxos de eventos (Fluxo principal, Subfluxos Alternativos e Subfluxos de Exceção).

Cenários.

Um cenário primário:

Rubens fornece sua chave de acesso. O sistema valida a chave e pede para Rubens escolher o semestre. Ele escolhe o semestre atual e pede para criar uma matrícula nova.

Rubens escolhe os cursos primários Inglês 101, Geologia 110, História Geral 200 e Álgebra 110. Também seleciona dois cursos alternativos: Teoria Musical 110 e Introdução à Programação Java 180.

O sistema constata que Rubens tem todos os pré-requisitos necessários e adiciona-o às listas de cada curso.

O sistema avisa que a matrícula foi realizada. Imprime o formulário de matrícula de Rubens. Envia informação de cobrança referente aos quatro cursos para ser processada no sistema de cobrança.

Cenários secundários:

Walker não seleciona os quatro cursos primários.

Um dos cursos primários selecionados não possui mais vagas.

Um curso primário ou secundário não está sendo oferecido.

Definição Cenário: É uma instância de um “Use Case”. O “Use Case” deve ser descrito através de vários cenários. Devem ser construídos tantos cenários quantos forem necessários para se entender completamente todo o sistema. Podem ser considerados como teste informais para validação dos requisitos do sistema.

Tipos de cenários:

Primários: São os cenários nos quais o fluxo segue normalmente. Não há quebra no fluxo por alguma espécie de erro.

Secundários: São os casos que compõem exceção. O fluxo normal de operação é interrompido.

Definição de Use case: É um modelo das funções a serem executadas pelo sistema e a interação com suas fronteiras. Sua principal aplicação é confirmar aos usuários e clientes as suas funcionalidades e comportamento.

3. Objetos e Classes de Objetos

Definição de Objeto. Representa um entidade que pode ser física, conceitual ou de software. É uma abstração de algo que possui fronteira definida e significado para a aplicação.

Componentes de um objeto:

- Estado
- Comportamento
- Identidade

Identidade: É o que identifica unicamente um objeto, mesmo que ele possua estados ou comportamento comuns a outros objetos.

Estado de um objeto: É cada condição na qual o objeto pode existir. É mutável ao longo do tempo. É implementado por um conjunto de atributos, os valores desses atributos e ligações que o objeto pode ter com outros objetos.

Comportamento de um objeto: Determina como um objeto age e reage a estímulos de outros objetos. É modelado através de um conjunto de mensagens que representam resultados visíveis das operações executadas internamente pelo objeto.

Definição de Classe É uma descrição de um grupo de objetos com atributos, comportamentos, relacionamentos com outros objetos e semântica comuns. Uma classe é uma abstração que enfatiza características relevantes dos objetos, suprimindo outras características. Portanto um objeto é sempre uma instância de uma classe.

Exemplo da Classe Curso.

Nome:

Curso

Estado:

Nome

Número do Curso

Localização

Dias do Curso

Número de créditos

Hora de Início

Hora de Término

Comportamento:

Adicionar um aluno

Cancelar um aluno

Obter alunos Matriculados

Determinar se a turma está completa

Exemplo da Classe Professor

Nome:

Professor

Estado:

Nome

Número do Empregado

Data de Admissão

Curso ministrado

Tipo de contratação

Salário

Comportamento:

Consultar Lista de Alunos

Indicar suas Disciplinas

A notação usada pela UML para representar uma Classe de Objetos é:

Nome
Atributo
Operações

A classe de objeto é representada por um retângulo, subdividido em três áreas. A primeira contém o nome da Classe. A segunda contém seus atributos. A terceira contém suas operações. A seguir tem-se os exemplos que esclarecem a representação descrita acima.

Curso	Professor
Número do Curso Nome Localização Dias do Curso Número de Créditos Hora de Início Hora de Término	Número do Empregado Nome Data de Admissão Curso ministrado Tipo de contratação Salário
Adicionar um Aluno() Cancelar um Aluno() Obter Alunos Matriculados() Determinar se a turma está completa()	Consultar Lista de Alunos() Indicar suas Disciplinas()

Estereótipos.

Estereótipo é um elemento de modelagem que rotula tipos de Classes de Objeto. Uma Classe de Objetos pode ter um ou mais tipos de estereótipos. Os estereótipos mais comuns são:

- Classe Fronteiriça
- Classe de Entidade
- Classe de Controle
- Classe de Exceção
- Metaclassse
- Classe Utilitária

<u><<Entidade>></u>	<u><<Entidade>></u>
Curso	Professor
Número do Curso Nome Localização Dias do Curso Número de Créditos Hora de Início Hora de Término	Número do Empregado Nome Data de Admissão Curso ministrado Tipo de contratação Salário
Adicionar um Aluno() Cancelar um Aluno() Obter Alunos Matriculados() Determinar se a turma está completa()	Consultar Lista de Alunos() Indicar suas Disciplinas()

A notação usada pela UML para Estereótipos, dentro da representação gráfica da Classe de Objeto, é coloca-lo entre << >> na área reservada para o nome da classe e acima deste.

Classe fronteiriça. É uma classe que modela a comunicação entre a vizinhança do sistema e suas operações internas. Exemplos: Interface tipo Janela, Protocolo de Comunicação, Interface de Impressão, Sensores, etc. No presente estudo de caso, sistema automatizado de Matrícula num Curso, as classes de objeto Formulário em Branco e Sistema de Cobrança são exemplos de estereótipos desta classe.

Classe de Entidade. É uma classe que modela objetos cuja informação e o comportamento associados são, de maneira geral, persistentes. No presente estudo de caso, sistema automatizado de Matrícula num Curso, as classes de objeto: Lista de Cursos, Curso, Catálogo e Matrícula, são exemplos de estereótipos desta classe.

Classe de Controle. É uma classe que modela o comportamento de controle específico para uma ou mais “Use Case”. Suas principais características são:

- Cria, ativa e anula objetos controlados.
- Controla a operação de objetos controlados
- Controla a concorrência de pedidos de objetos controlados.
- Em muitos casos corresponde a implementação de um objeto intangível.

No presente estudo de caso, sistema automatizado de Matrícula num Curso, a classe de objeto Gerente de Registro é um exemplo de estereótipo desta classe.

4. Interação entre objetos.

A UML se utiliza de dois diagramas para representar a interação entre os objetos: Diagrama de Seqüência e Diagrama de Colaboração.

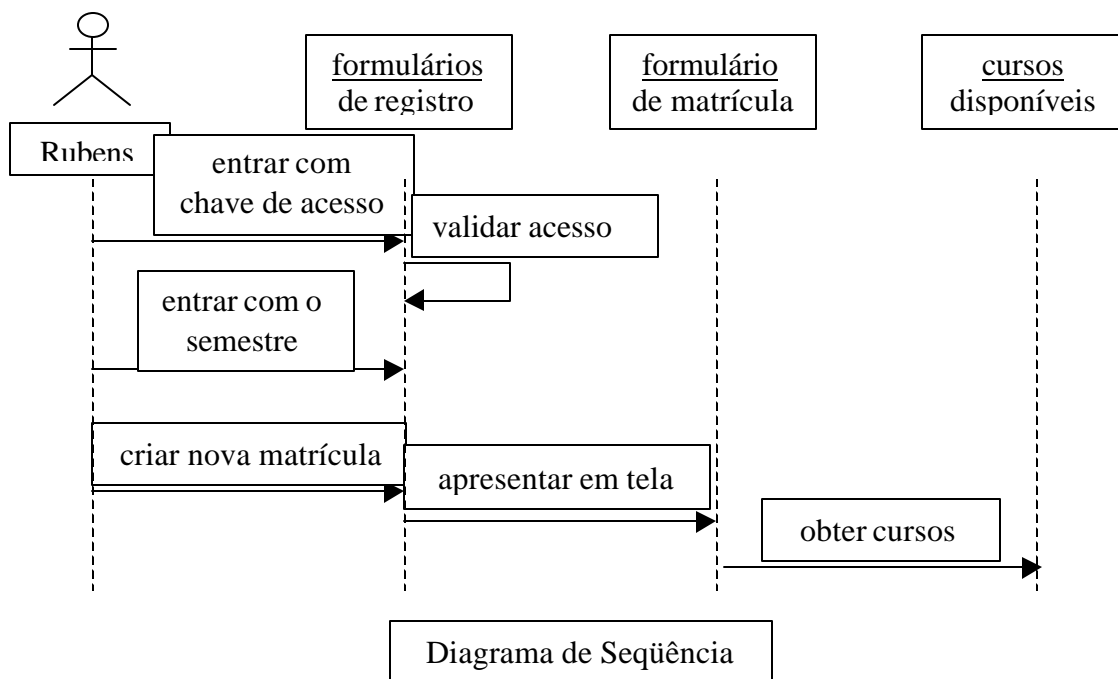
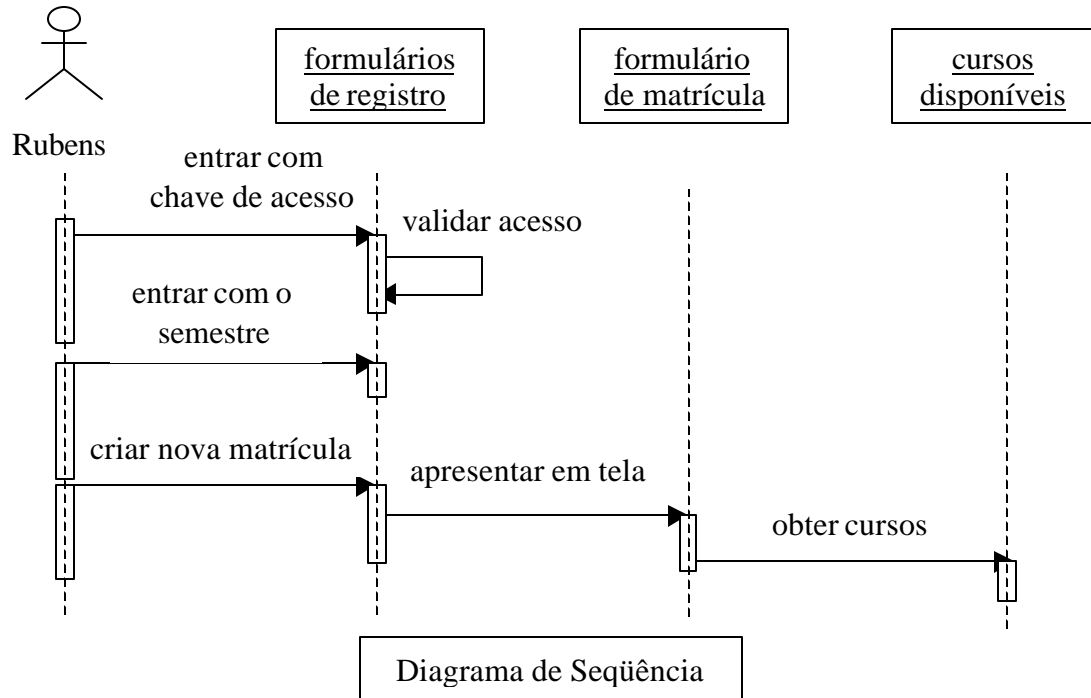


Diagrama de Seqüência mostra a interação entre os Objetos ao longo do tempo. Apresentando os objetos que participam da interação e a seqüência de mensagens trocadas.

A notação usada pela UML para representar o Diagrama de Seqüência, é a seguinte:

- Objetos são representados por retângulo com seus nomes sublinhados.
- As linhas de vida dos Objetos são representadas por linhas verticais tracejadas.
- As interações entre Objetos são indicadas por flechas horizontais que são direcionadas da linha vertical que representa o Objeto cliente para a linha que representa o Objeto fornecedor.

- As flechas horizontais são rotuladas com as mensagens.
- A ordem das mensagens no tempo é indicada pela posição vertical, com a primeira mensagem aparecendo no topo.
- A numeração é opcional e baseada na posição vertical.



Foco de Controle. Representa o tempo relativo que o fluxo de controle esta focalizado num dado Objeto. Ele representa o tempo que um Objeto dedica a uma dada mensagem.

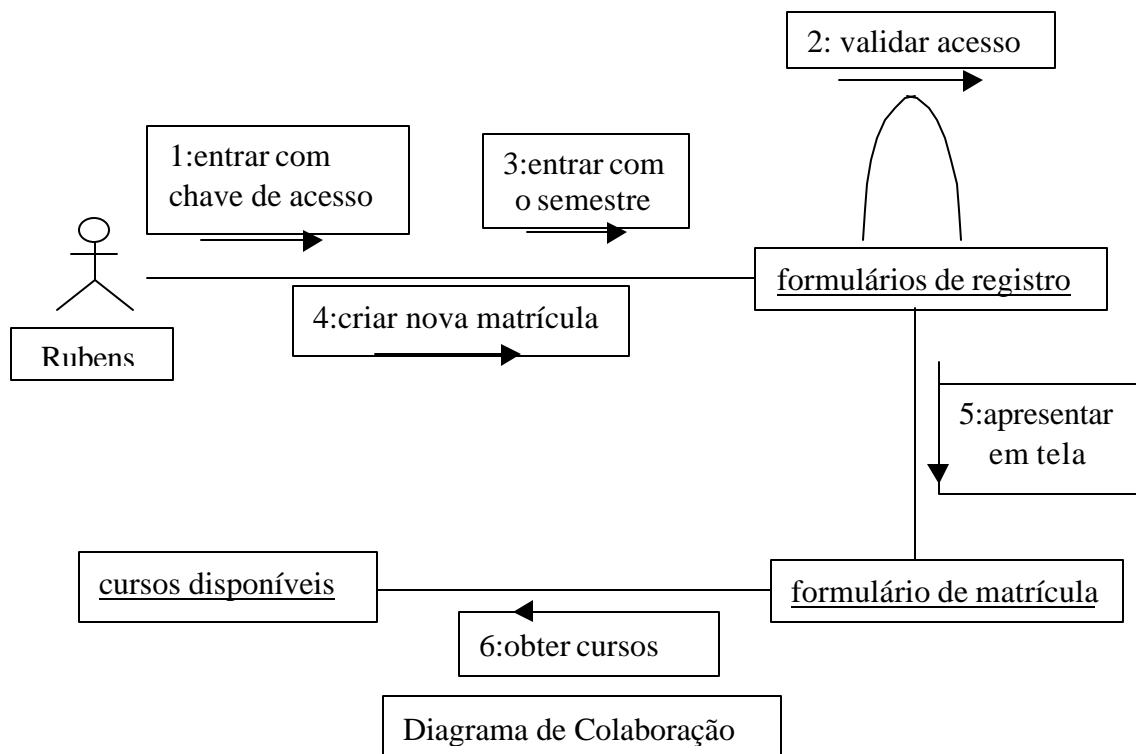


Diagrama de Colaboração. É um modo alternativo para representar a troca de mensagens entre um conjunto de Objetos. O Diagrama de Colaboração mostra a interação organizada entorno dos Objetos e suas ligações uns com os outros.

A notação usada pela UML para representar o Diagrama de Colaboração, é a seguinte:

- Objetos são representados por retângulo com seus nomes sublinhados.
- As interações entre Objetos são indicadas por uma linha conectando-os.
- As ligações indicam a existência de um caminho para comunicação entre os Objetos conectados.
- As ligações no Diagrama de Colaboração podem ser apresentada por:
 - flechas apontando do Objeto cliente para o Objeto fornecedor.
 - o nome da mensagem.
 - Numeração seqüencial, mostrando a ordem relativa de envio das mensagens.

5. Como Descobrir as Classes de Objetos

Análise do “Use Case” É o processo de examinar o conjunto de “Use Cases” para extrair os Objetos e Classes do sistema sob desenvolvimento. Os Objetos e Classes são podem ser obtidos do detalhamento dos Cenários.(instâncias de “Use Case”).

Cenário para criar matrícula-

Cleber entra com o **número de identificação do aluno 369 53 3449** e o **sistema** valida o **número**. O sistema pergunta a qual **semestre** refere-se a matrícula. Cleber indica que é para o **semestre atual** e escolhe a opção “**nova matrícula**”.

Da **lista de cursos disponíveis**, Cleber seleciona como **curso de primeira escolha**: **Engenharia de Software 101**, **Sistemas Computacionais Cliente Servidor 110**, **Tópicos em Análise Orientada a Objetos 200** e **Gerência de Mainframe 110**. Ele seleciona como **curso de segunda escolha**: **Introdução a Programação Java 200** e **Teoria da Musica 300**.

O sistema verifica que Cleber tem todos os **pré-requisitos necessários**, examinando os **registro do aluno** e adiciona-o à **lista de alunos do curso**.

O sistema indica que a **atividade** esta completa. O sistema imprime a **matrícula** e envia a **informação de cobrança**, referente aos **quatro cursos**, para processamento no **sistema de cobrança**.

Objetos pertencente à Classe de Entidades são identificados examinando-se os substantivos e frases substantivadas no cenário. No cenário acima estão destacados os substantivos candidatos à Objetos da Classe de Entidade. Os substantivos podem ser: Objetos, descrição do estado de um Objeto, entidade externa e/ou ator ou ainda nenhuma das anteriores.

Lista de substantivos e sua classificação.

Cleber.- ator

número de identificação do aluno 369 53 3449 – propriedade de aluno

sistema. – o que está sendo definido

número – propriedade do aluno

semestre – estado (que é selecionado quando aplicável)

semestre atual – mesmo que semestre

nova matrícula – Objeto candidato

lista de cursos disponíveis - Objeto candidato

cursos de primeira escolha - estado

Engenharia de Softwarwe 101 - Objeto candidato

Sistemas Computacionais Cliente Servidor 110 - Objeto candidato

Tópicos em Análise Orientada a Objetos 200 - Objeto candidato

Gerência de Mainframe 110 - Objeto candidato

cursos de Segunda escolha - estado

Introdução a Programação Java 200 - Objeto candidato

Teoria da Musica 300.- Objeto candidato

pré-requisitos necessários – cursos com outra identificação

registro do aluno - Objeto candidato

lista de alunos do curso. - Objeto candidato

atividade . expressão

matrícula – mesmo que nova matrícula

informação de cobrança - Objeto candidato

quatro cursos - informação necessária ao sistema de cobrança

sistema de cobrança. - ator

Lista de Objetos da Classe Entidade

nova matrícula – Lista de cursos para um dado semestre de um dado aluno

lista de cursos disponíveis – Lista de todos os cursos que estão sendo oferecidos no semestre

Engenharia de Softwarwe 101 – Um curso oferecido no semestre

Sistemas Computacionais Cliente Servidor110 - Um curso oferecido no semestre

Tópicos em Análise Orientada a Objetos 200 - Um curso oferecido no semestre

Gerência de Mainframe 110 - Um curso oferecido no semestre

Introdução a Programação Java 200 - Um curso oferecido no semestre

Teoria da Musica 300 - Um curso oferecido no semestre

registro do aluno – Lista dos cursos feitos pelo aluno nos semestres anteriores

lista de alunos do curso – Lista com os alunos matriculados num curso específico.

informação de cobrança – Informações necessárias para o ator sistema de cobrança.

Criando as Classes de Entidades - Baseando-se na similaridade de estrutura e de comportamento dos objetos.

Lista de Classes de Entidades presentes no Cenário “Criar Matrícula”.

Matrícula – Lista dos cursos para um dado semestre para um dado aluno.

Catálogo – Lista de todos os cursos oferecidos num semestre.

Curso – Curso oferecido para um semestre.

RegistroDoAluno – Lista dos cursos feitos anteriormente.

ListaDosAlunosNumCurso – Lista dos alunos matriculados num curso específico.

InformaçõesDeCobrança – Informações necessárias para o ator sistema de cobrança.

<<Entidade>> Matrícula

<<Entidade>> Curso

<<Entidade>> RegistroDoAluno

<<Entidade>> Catálogo

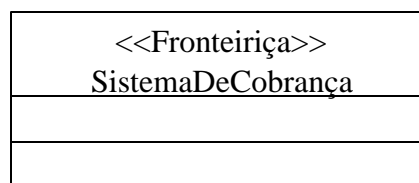
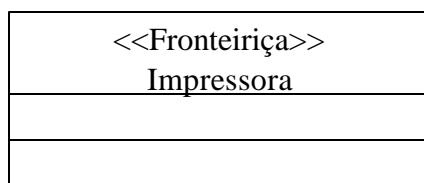
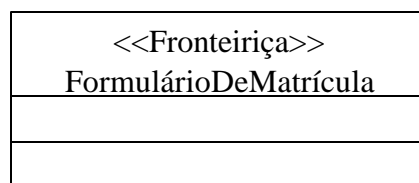
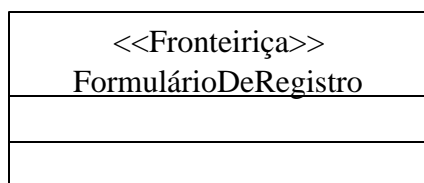
<<Entidade>> ListaDosAlunosNumCurso

<<Entidade>> InformaçõesDeCobrança

Objetos pertencente à Classe Fronteira são identificados examinando-se cada par ator/cenário e criando-se as classes fronteiriças óbvias. Classes Fronteiriças também são criadas para comunicação sistema/sistema e para descrever a escolha de protocolos de comunicação.

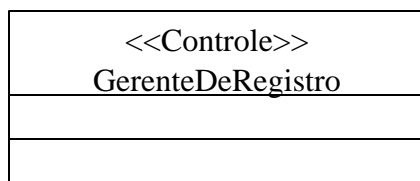
Exemplos de classe fronteira:

- Deve ser apresentado ao aluno, mais de uma opção do “Use Case” “Matrícula nos Cursos”. Para tanto, é criada a Classe “FormulárioDeRegistro” para permitir ao estudante selecionar a opção desejada.
- O aluno deve fornecer ao sistema a informação dos cursos escolhidos. Para tanto é criada a Classe “FormulárioDeMatrícula” para permitir que o aluno entre com a informação.
- O matrícula do aluno é impressa. Para tanto é criada a Classe “Impressora”.
- A informação de cobrança é enviada para o sistema de cobrança. Para tanto, é criada a Classe “SistemaDeCobrança”.



Objetos pertencente à Classe de Control tipicamente, contêm a informação de seqüenciamento. Cada “Use Case” deve ter uma Classe de Controle, responsável pelo fluxo de eventos. Para tanto, é criada a Classe “GerenteDeRegistro”. Esta classe, para cada curso selecionado na Classe Fronteira “FormulárioDeMatrícula” deve executar as seguintes atividades:

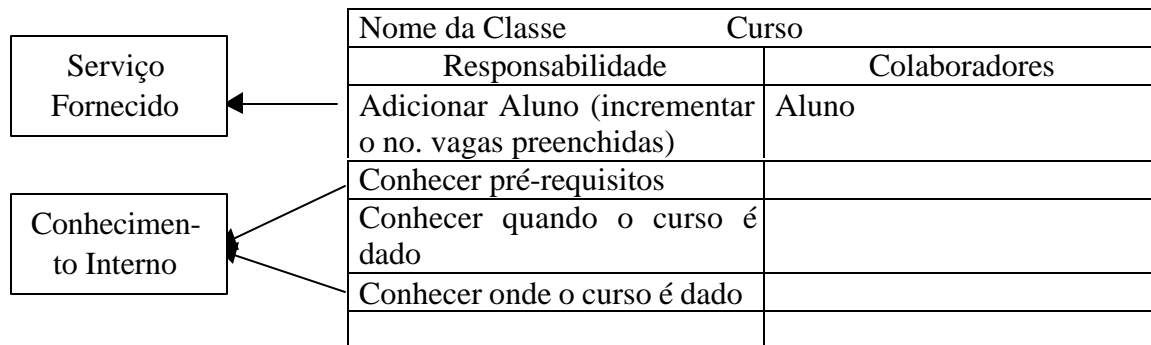
- Busca na Classe Curso seus pré-requisitos.
- Verifica, através da Classe RegistroDoAluno se todos os pré-requisitos do curso selecionado foram satisfeitos.
- Sabe o que fazer se um pré-requisito não foi satisfeito.
- Interroga se há vaga no curso.
- Se houver vaga, pede a classe ListaDeAlunosNoCurso para adicionar o aluno.
- Sabe o que fazer se um dos quatro cursos não está disponível.
- Cria os objetos: MatrículaDoAluno e InformaçãoDeCobrança.
- Verifica se o sistema de cobrança está habilitado a receber a informação de cobrança.



6. Cartão Class-Responsibility-Collaboration (CRC)

Novas Classe podem ser descoberta através do uso do cartão CRC. Um CRC é um cartão de 10cm x 15cm, que contém:

- Nome e descrição da Classe.
- As responsabilidades da Classe
 - Conhecimento interno da Classe
 - Serviços fornecidos pela Classe
- Os colaboradores com essas responsabilidades
 - Um colaborador é uma classe cujos serviços são necessários para execução de dada responsabilidade.



Uma sessão com o uso do cartão CRC compreende:

- Um grupo de pessoas é escolhido para representar um cenário.
- É criado um cartão para cada Classe de Objeto já identificado dentro desse cenário.
- A cada participante é associada uma Classe, de modo que cada pessoa torna-se aquela Classe.
- O cenário é encenado pelos participantes.
- Os cartões são preenchidos com as responsabilidades e os colaboradores.
- Novos cartões são criados para classes de Objetos descobertos na sessão.

Benefícios do uso do cartão CRC:

- À medida que os cenários são encenados, padrões de colaboração emergem.
- Os cartões que colaboram entre si podem ser arranjados fisicamente próximos
- Esse arranjo ajuda a identificar hierarquias de generalização/especialização ou agregação entre as classes.
- O uso do cartão CRC é mais efetivo para grupo iniciantes no uso de técnicas OO.

O uso do cartão CRC permite :

- Confirmar ou não as classes de objetos candidatas, além de permitir a descoberta de novas Classes.
- Determinar o relacionamento entre as Classes.
- Identificar atributos (conhecimento interno) e operações (serviços fornecidos).

À medida que Classes de Objetos são descobertas, elas são documentadas nos diagramas de interação (diagramas de seqüência e de colaboração) anteriormente confeccionados.

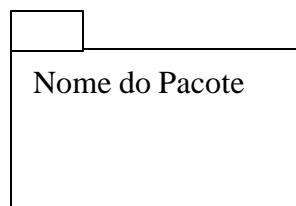
7. Pacote

As classes pertencentes ao Sistema de Matrícula podem ser agrupadas em três pacotes:

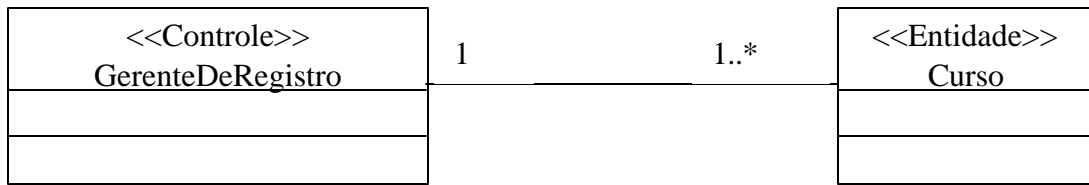
- ElementosDaUniversidade
 - Matrícula
 - Curso
 - RegistroDeAluno
 - Catálogo
 - ListaDosAlunosNumCurso
 - InformaçõesDeCobrança
- RegrasDeNegócio
 - GerenteDeRegistro
- Interfaces
 - FormulárioDeRegistro
 - FormulárioDeMatrícula
 - Impressora
 - SistemaDeCobrança

Definição de Pacote é uma generalização com o propósito de organizar as Classes de Objetos em grupos. Esta abordagem facilita a análise a medida que o número de Classes de Objetos cresce num do cenário.

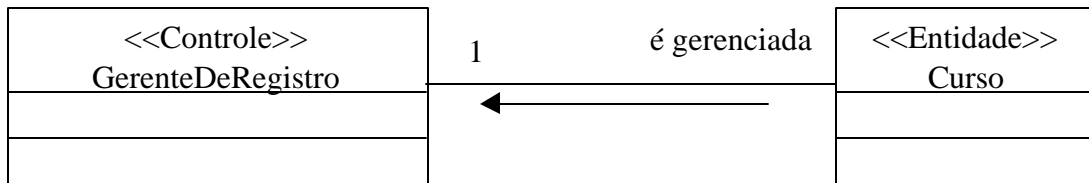
A notação usada pela UML para representar pacotes é:



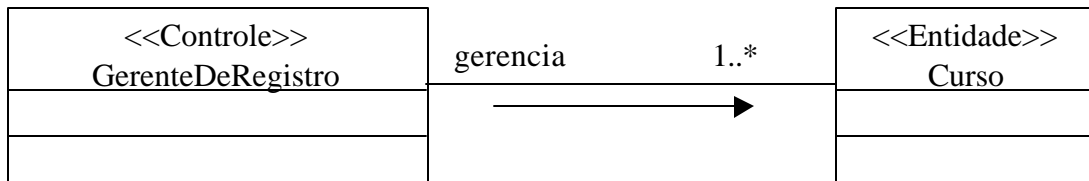
Exemplo de Indicador de Multiplicidade de uma associação



Exemplos de como interpretar (ler) a associação representada num Diagrama de Classes.

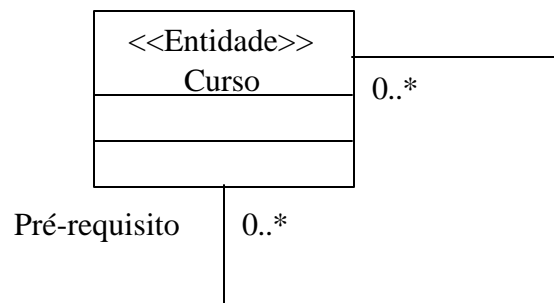


Um objeto curso é gerenciado por um único GerenteDeRegistro.



Um GerenteDeRegistro gerencia um ou muitos Cursos.

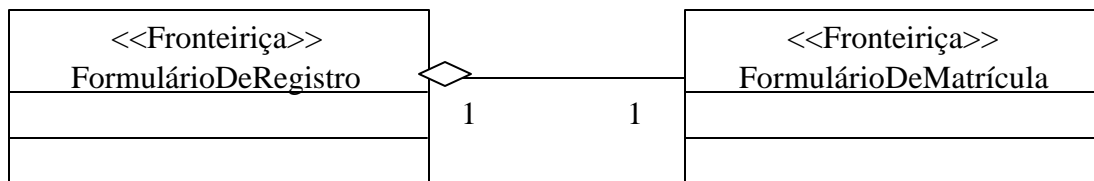
Associação Reflexiva – é quando os Objetos da própria Classe estão se relacionando.



Um curso está associado a nenhum ou muitos pré-requisitos. Ou ainda, um curso é pré-requisito de nenhum ou muitos cursos.

Agregação.

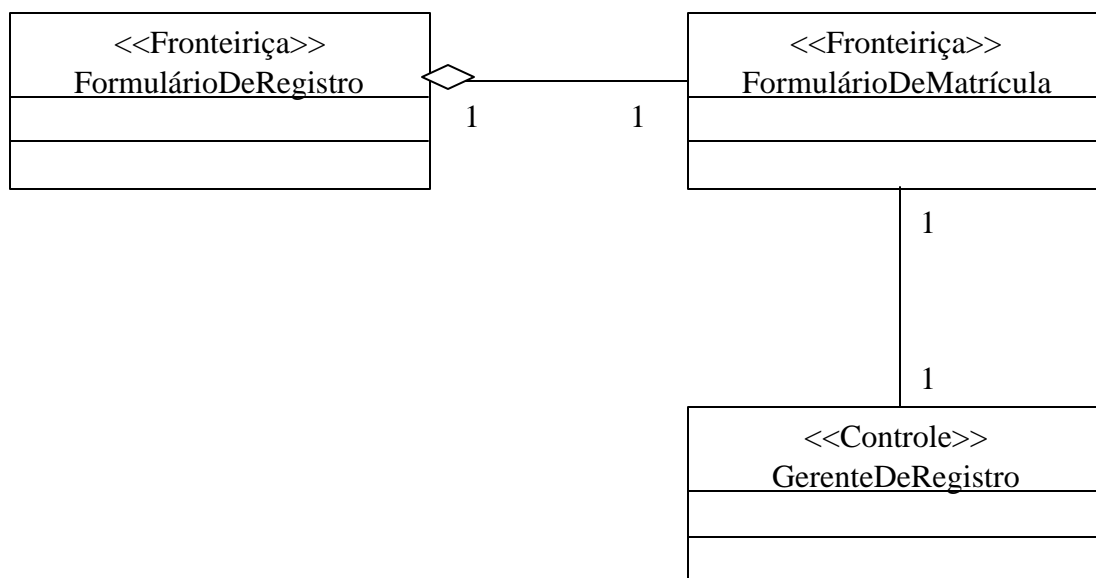
- É uma forma especializada de associação na qual um todo é relacionado com suas partes. Também conhecida como relação de conteúdo.
- É representada como uma linha de associação com um diamante junto à Classe agregadora.
- A multiplicidade é representada da mesma maneira que nas associações.



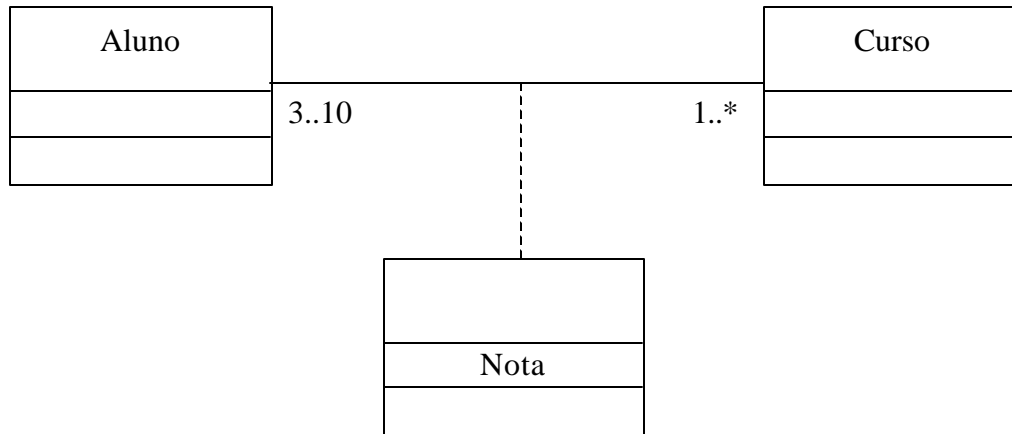
Um objeto da Classe FormulárioDeRegistro contém um único objeto FormulárioDeMatrícula. Um objeto FormulárioDeMatrícula está contido num único objeto FormulárioDeRegistro.

Agregação Reflexiva – é quando Objetos de uma Classe é composto de Objetos da própria Classe.

Associação ou Agregação



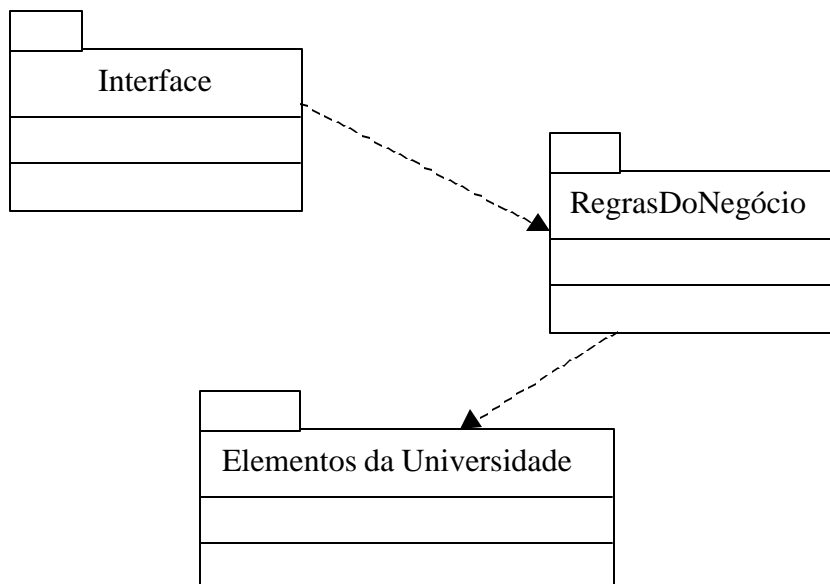
Classe de uma Associação de Classe – permite adicionar atributos, operações e outras características a uma dada associação.



A classe de uma Associação de Classe normalmente é gerada a partir de uma associação de muitos para muitos.

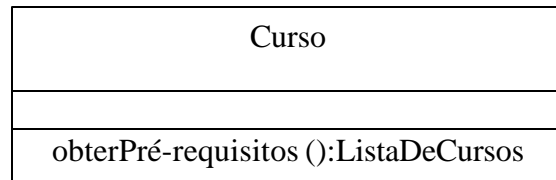
Relacionamento entre Pacotes

- Pacotes são relacionados uns com os outros usando um relacionamento de dependência.
- Se uma Classe de um pacote interage com uma Classe de outro pacote, a relação de dependência é adicionada a nível de pacote.
- Relacionamento entre pacotes são obtidos a partir dos diagramas de Classe e de Cenário.

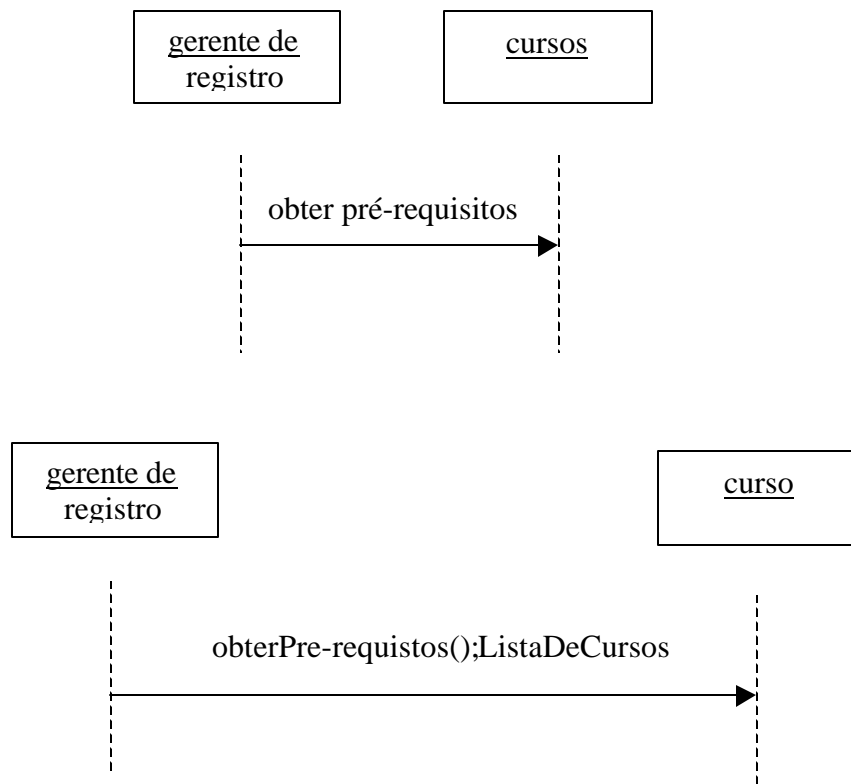


9. Operações e Atributos

Definição de Operações – São procedimentos que executam as responsabilidades de uma Classe de Objetos e portanto definem o comportamento dos objetos da Classe. Uma operação é um serviço que pode ser requisitado por um Objeto para realizar um comportamento. Operações devem ser nomeadas em função de suas saídas e não em função de seus passos internos. Exemplo:



A notação usada pela UML é apresentar Operações no terceiro compartimento da Caixa de representação de Classe de Objetos, conforme mostra a figura acima.



O Diagrama de Seqüenciamento obtido na análise de cenário (representado no da figura acima) deve ser refeito de modo que inclua nas mensagens enviadas, de uma Classe de Objeto para outra, as operações e os seus resultados (conforme a parte inferior da figura acima).

Definição de Atributos - São dados que caracterizam uma instância da Classe de Objetos. Atributos não tem comportamento. Atributos são sempre valorados. Cada valor de um atributo é particular para um dado objeto. Atributos são nomeados por substantivo simples ou por verbo substantivado. Cada atributo tem uma definição concisa e clara. Exemplo:

Curso
nome descrição localização horário númeroDeCréditos
obterPré-requisitos():ListaDeCursos

A notação usada pela UML é apresentar Atributos no segundo compartimento da Caixa de representação de Classe de Objetos, conforme mostra a figura acima.

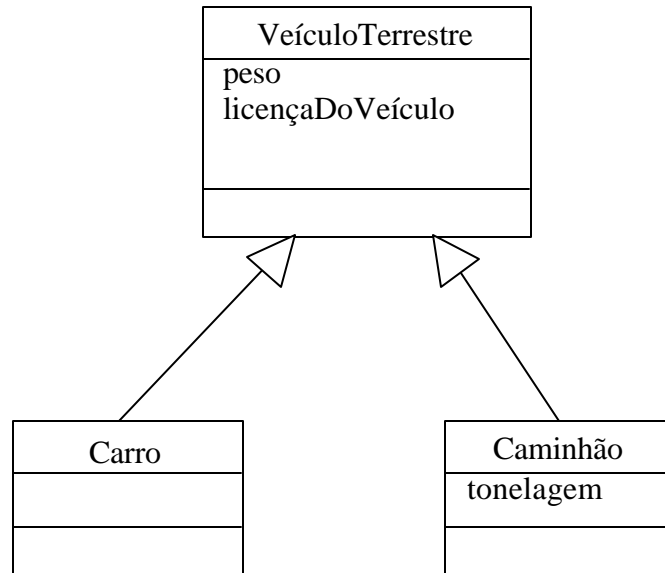
Cada atributo tem: tipo do dado e valor inicial. Exemplo, o tipo de dado para o atributo horário é: hh:mm:ss e o valor inicial é 00:00:00.

A determinação dos atributos de uma Classe de Objetos pode ser consequência de:

- Análise dos fluxos de evento nos “Use Case”.
- Definição de uma Classe de Objetos.
- Conhecimento do sistema.

10. Herança

Definição de Herança – É uma hierarquia de abstrações na qual uma subclasse herda a estrutura e/ou comportamento de uma ou mais superclasses.



A Classe de Objetos Caminhão tem três atributos: peso e licençaDoVeículo, herdadas da classe VeículoTerrestre, e tonelagem próprio da Classe. A Classe de Objetos Carro tem dois atributos: peso e licençaDeVeículo, herdadas da classe VeículoTerrestre, e nenhum atributo próprio. Caminhão é uma espécie de VeículoTerrestre. Carro é uma espécie de VeículoTerrestre.

Herança simples é quando uma subclasse herda estrutura e/ou comportamento de uma única superclasse.

Herança múltipla é quando uma subclasse herda estrutura e/ou comportamento de mais de uma superclasse.

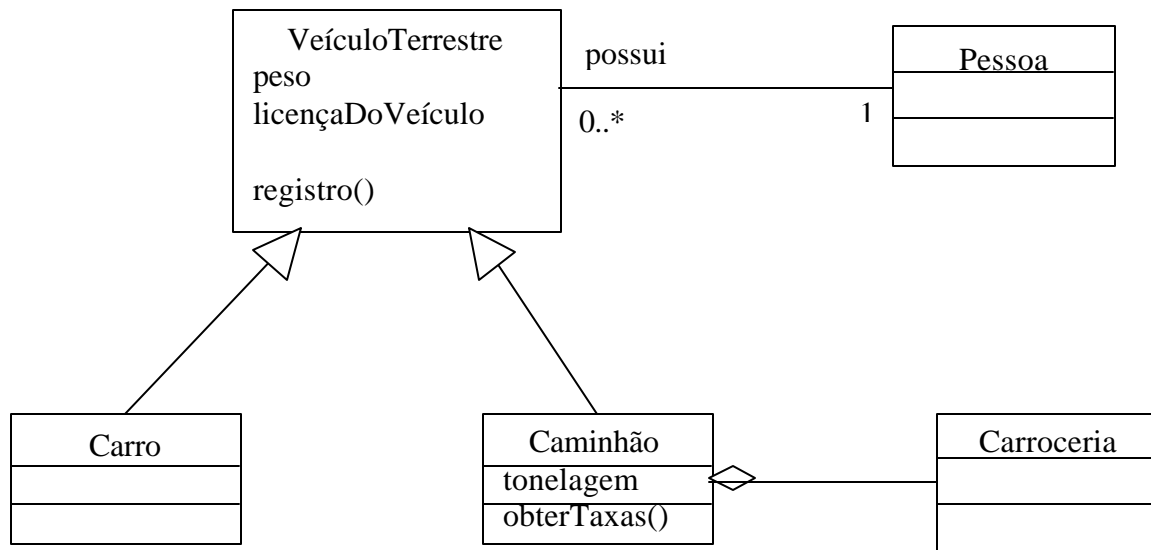
Herança é uma relação entre Classes de Objetos e não uma relação entre instância das Classes.

Uma subclasse herda:

- Atributos
- Operações
- Relacionamentos

Uma subclasse pode:

- Adicionar novos atributos, operações e relacionamentos.
- Redefinir operações herdadas.



Herança de Atributos – A subclasse de Objetos Caminhão herda os atributos: peso e licençaDeVeículo da superclasse VeículoTerrestre.

Herança de Operações – A subclasse de Objetos Caminhão herda a operação registro() da superclasse VeículoTerrestre.

Herança de Relacionamentos – As subclasses de Objetos Caminhão e Carro herdam o relacionamento “possui” existente entre a superclasse VeículoTerrestre e a Classe Pessoa.

Generalização – É a capacidade de se criar superclasses que encapsulam estrutura e/ou comportamento comuns a várias subclasses.

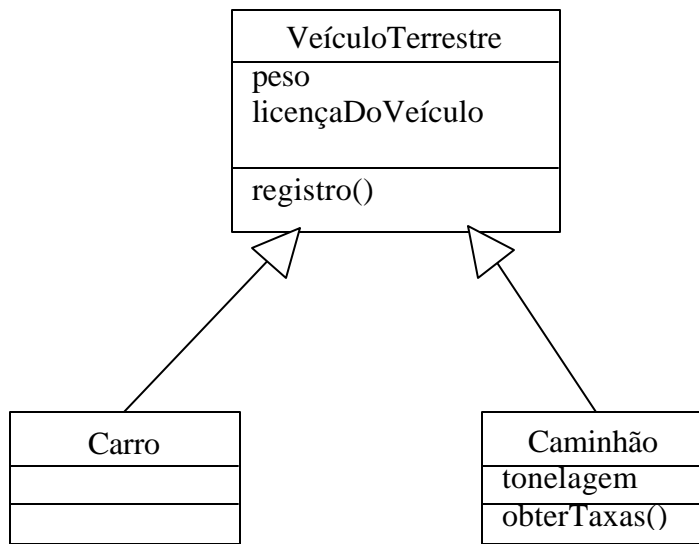
Os procedimentos para se obter a generalização são:

- Identificar similaridades de estrutura/comportamento entre várias Classes.
- Criar a superclasse para encapsular a estrutura/comportamento comum.
- As classes originais passam a ser subclasses da nova superclasse criada.

Especialização – É a capacidade de se criar subclasses que representam refinamentos no qual a estrutura e/ou comportamento da superclasse são adicionados ou modificados.

Os procedimentos de especialização são:

- Perceber que algumas instâncias apresentam estrutura e/ou comportamento especializado.
- Criar subclasses para agrupar instâncias de acordo com sua especialização.



A superclasse **VeículoTerrestre** é uma **generalização** das subclasses **Carro** e **Caminhão**. Por outro lado, as subclasses **Carro** e **Caminhão** são **especializações** da superclasse **VeículoTerrestre**.

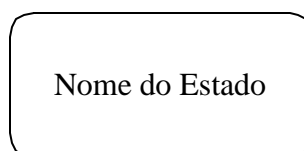
11. Comportamento

O comportamento de uma Classe de Objetos é representado através de um Diagrama de Transição de Estado, que descreve o ciclo de vida de uma dada classe, os eventos que causam a transição de um estado para outro e as ações resultantes da mudança de estado.

O espaço amostral dos estados de uma dada Classe corresponde a enumeração de todos os estados possíveis de um objeto.

O estado de um Objeto é uma das possíveis condições na qual o objeto pode existir. O estado compreende todas as propriedades do objetos (estáticas) associadas aos valores correntes (dinâmico) de cada uma dessas propriedades.

A notação UML para representar o estado de uma classe corresponde a um retângulo com a bordas abauladas.

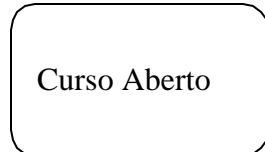


Estados e Atributos

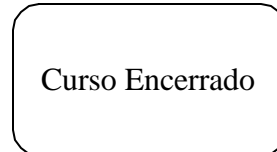
Estados podem ser distinguidos pelos valores assumidos por certos atributos.

Exemplo - O número máximo de estudantes por curso, no “Use Case” Matrícula do Aluno, é igual a 10.

NúmeroDeAlunos ≥ 0



NúmeroDeAlunos = 10

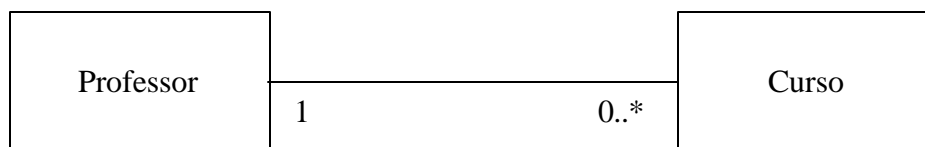


Estados e Ligações

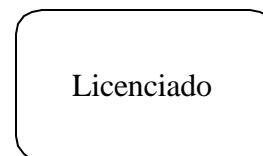
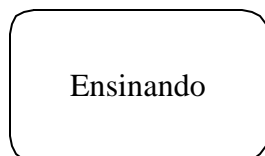
Estados também podem ser distinguidos pela existência de certas ligações.

Exemplo – A instância da Classe Professor pode ter dois estados:

- Ensinando: quando o Professor esta ministrando um Curso.
- Licenciado: quando não esta ministrando nenhum Curso.



Estados possíveis da Classe Professor.



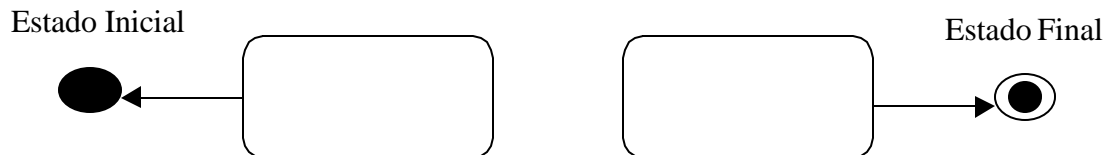
Estados Especiais

Estado Inicial é o estado atribuído a um objeto quando é criado. O estado Inicial tem as seguintes características:

- É mandatório
- Somente um estado Inicial é permitido.
- O estado Inicial é representado por um círculo preenchido.

Estado Final é o estado que indica o fim do ciclo de vida de um objeto. O estado Final tem as seguintes características:

- É opcional.
- Pode existir mais de um estado final.
- O estado Final é representado por um “olho de boi”.



Eventos

Um evento é uma ocorrência que acontece em algum ponto no tempo e que pode modificar o estado de um objeto, podendo gerar uma resposta. Exemplo:

- Adicionar um aluno a um curso.
- Criar um novo curso.

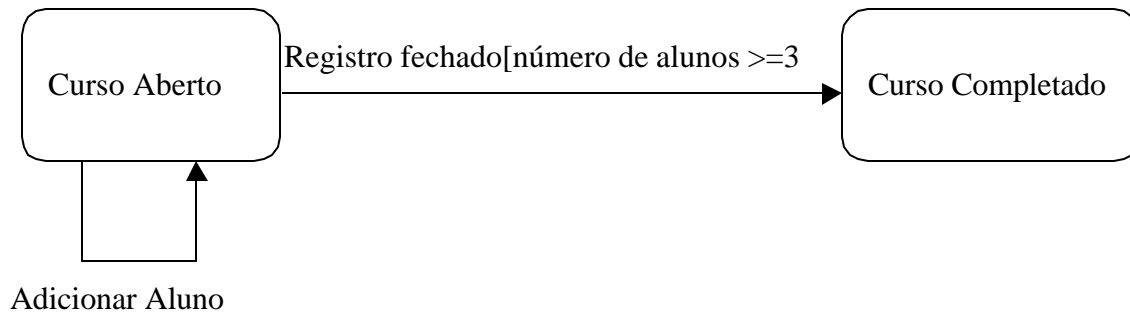
Transição

É a mudança do estado atual para o estado subsequente como resultado de algum estímulo. O estado subsequente pode ser igual ao estado original. Uma transição pode ocorrer em resposta a um evento. As transições rotuladas com o nome dos eventos.



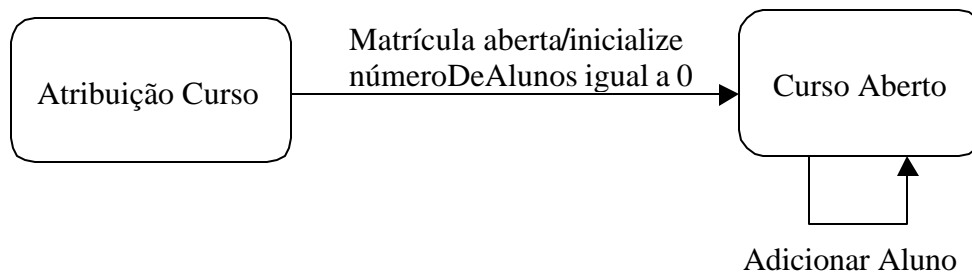
Condição de Guarda

A condição de guarda é uma expressão Booleana de valores de atributo que permitem que a transição ocorra somente se a condição assumida pela expressão é verdadeira.



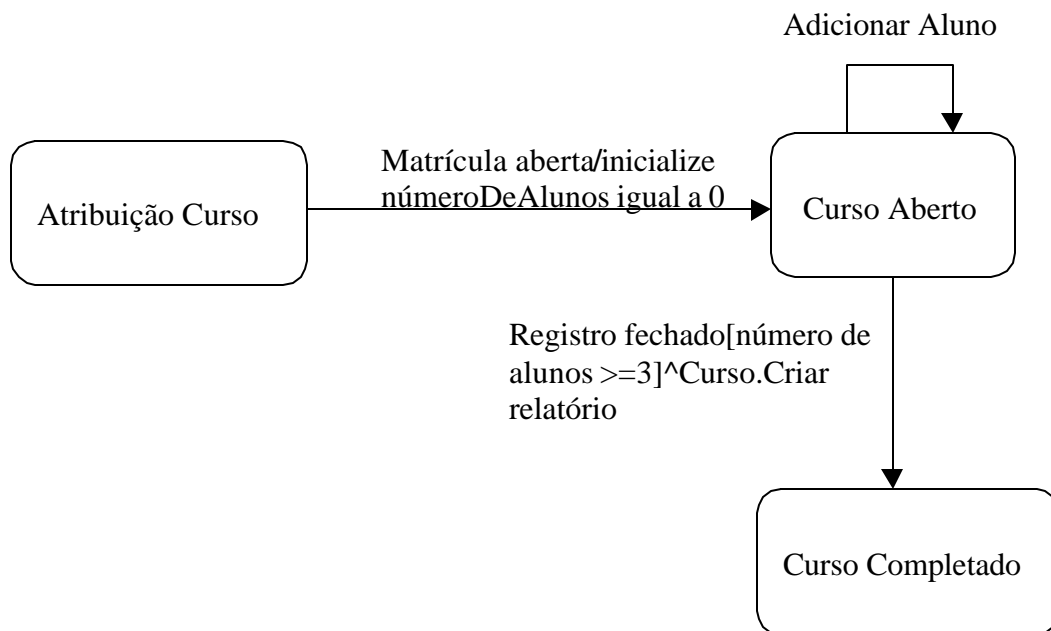
Ações

É uma operação que está associada a uma transição, ocorrendo instantaneamente e que não pode ser interrompida. Nome de uma ação é mostrado, na seta indicativa da transição, precedida por um barra inclinada (/).



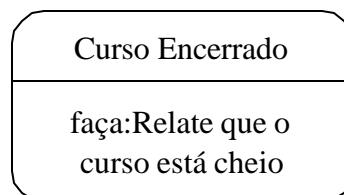
Envio de eventos a partir de outro evento

Um evento pode provocar o envio de outro evento. O nome do evento enviado é mostrado, na seta indicativa de transição, precedido por um circunflexo (^) seguido pelo nome da Classe para onde o evento será enviado, separados por um ponto. Exemplo: Evento1^Classe.Evento2, onde Evento1 é o evento que causou a transição e Evento2 é o evento gerado a partir da transição.



Atividade

É uma operação que está associada a um estado, leva um tempo para ser executada e que pode ser interrompida.

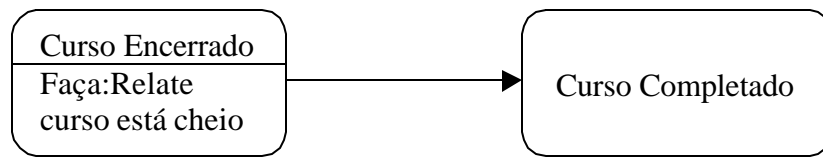


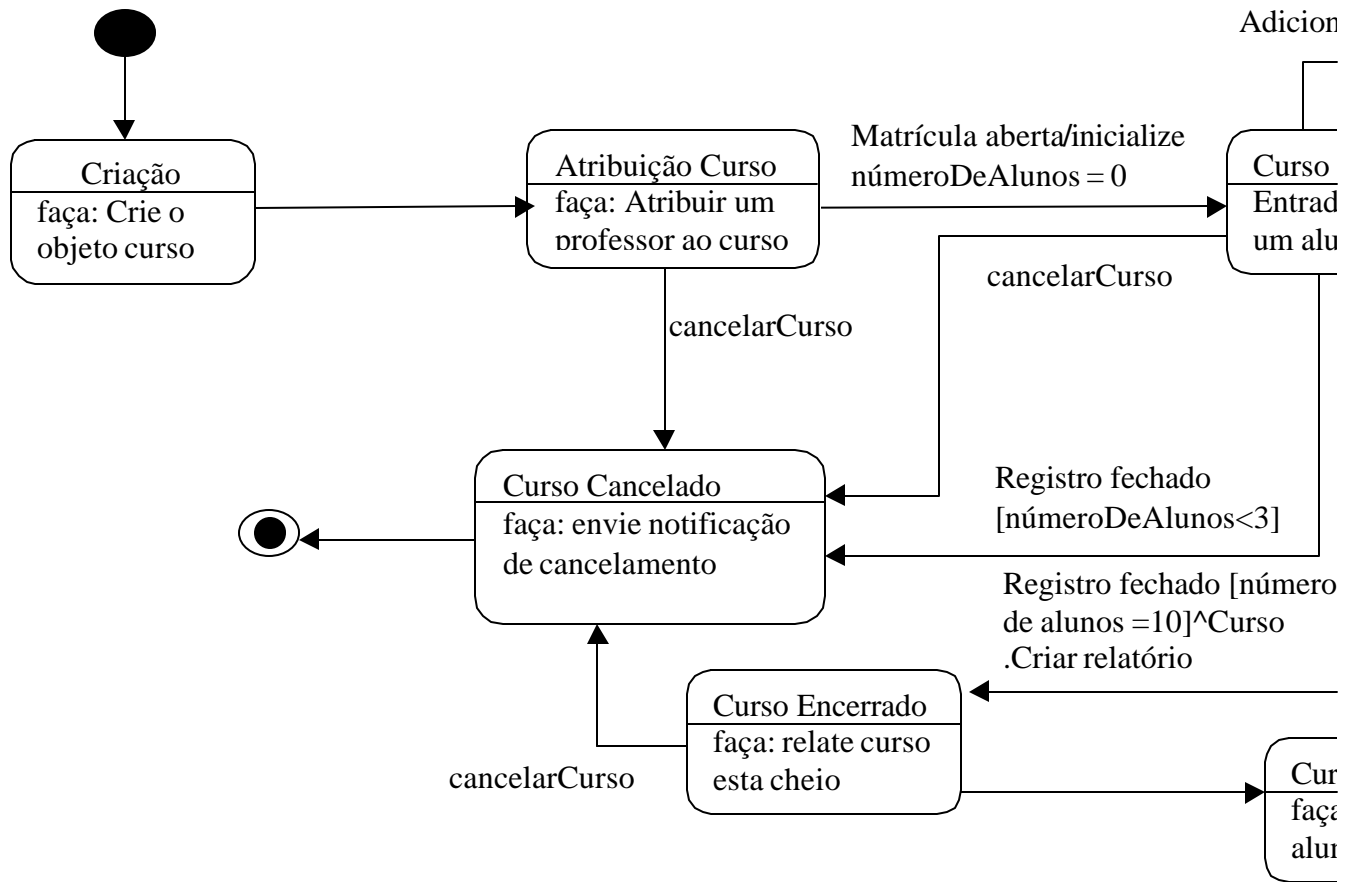
Envio de eventos a partir de atividade

Uma atividade também pode provocar o envio de um evento para um outro Objeto.

Transição Automática

Algumas vezes, o único propósito da existência de um estado é desenvolver uma atividade. Uma transição automática ocorre quando a atividade é completada. Se múltiplas transições automáticas existem, uma condição de guarda é necessária para cada transição e as condições de guarda devem ser mutuamente exclusivas.





D
i
c