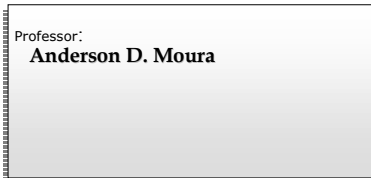


Faculdade Pilares - Bacharelado em Informática
Projeto e Desenvolvimento de Software

UML

Fevereiro, 2008



1

Faculdade Pilares - Bacharelado em Informática
Casos de Uso

Os Caso de Uso referem-se aos serviços, tarefas ou funções oferecidas pelo sistema, como emitir um relatório ou cadastrar a venda de algum produto. São utilizados para expressar e documentar os comportamentos pretendidos para as funções do sistema.



2

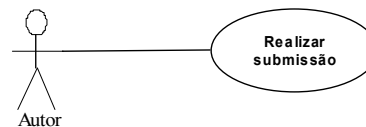
Faculdade Pilares - Bacharelado em Informática
Casos de Uso

Os Casos de Uso costumam ser documentados, demonstrando qual o comportamento pretendido para o Caso de Uso em questão e quais funções ele executará quando for solicitado. **Normalmente um Caso de Uso é documentado de maneira informal, mas nada impede que sejam inseridos detalhes de implementação em uma linguagem mais técnica.**

3

Faculdade Pilares - Bacharelado em Informática
Associações

As Associações representam os relacionamentos entre os atores que interagem com o sistema, entre os Atores e os Casos de Uso ou os relacionamentos entre os Casos de Uso e outros Casos de Uso, estes últimos recebem nomes especiais, como Inclusão, Extensão e Generalização. **Uma associação entre um Ator e um Caso de Uso demonstra que o Ator utiliza-se de alguma maneira, da função representada pelo Caso de Uso.**



4

Faculdade Pilares - Bacharelado em Informática
Associações

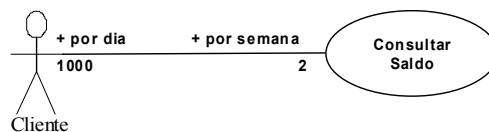
Um ator pode interagir com um caso de uso com os seguintes objetivos:

- Invocar um caso de uso: uma instância de ator sempre invoca uma instância de caso de uso;
- Solicitar dados armazenados no sistema que o caso de uso vai buscar e apresentar ao ator;
- Modificar os dados armazenados por meio de um diálogo com o sistema;
- Informar que algo ocorreu externamente e deve ser comunicado ao sistema.

5

Faculdade Pilares - Bacharelado em Informática
Associações

Um ator inicia um caso de uso, mas, depois de iniciado, o caso de uso pode comunicar-se com vários atores, portanto as associações servem para mostrar quais atores se comunicam com o caso de uso. A multiplicidade da associação mostra quantas instâncias de um ator podem se comunicar com um instância do caso de uso ao mesmo tempo



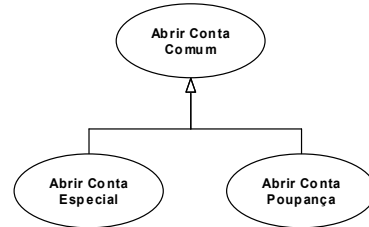
6

Especificações/Generalizações

Este relacionamento é uma forma de Associação entre casos de uso na qual existem dois ou mais casos de uso com características semelhantes, apresentando pequenas diferenças entre si. Quando tal situação ocorre, costuma-se definir um **Caso de Uso Geral**, que descreve as características compartilhadas por todos os casos de uso em questão, e então relaciona-lo com estes, cuja documentação conterá somente as características especificam de cada um. Assim não será necessário colocar a mesma documentação para todos os casos de uso envolvidos, porque toda a estrutura de um Caso de Uso generalizada é herdada pelos Casos de Uso Especializados, incluindo quaisquer possíveis Associações. **Esta associação é representada por uma reta com uma seta mais grossa, partindo dos Casos de Uso Especializados e atingindo o Caso de Uso Geral.**

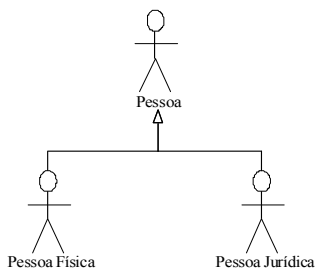
7

Especificações/Generalizações



8

Especificações/Generalizações



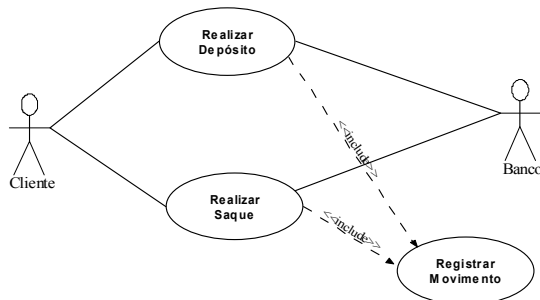
9

Inclusão

Esta situação é utilizada quando existe uma situação ou rotina comum a mais de um Caso de Uso. Quando isso ocorre, a documentação dessa rotina é colocada em um Caso de Uso específico para que outros Casos de Uso utilizem-se desse serviço, evitando-se descrever uma mesma seqüência de passos em vários Casos de Uso. Os relacionamentos de uma inclusão indicam uma obrigatoriedade, ou seja, quando um determinado Caso de Uso possui um relacionamento de inclusão com outro, a execução do primeiro obriga também a execução do segundo. **Um relacionamento de inclusão pode ser comparado à chamada de uma sub - rotina.**

10

Inclusão



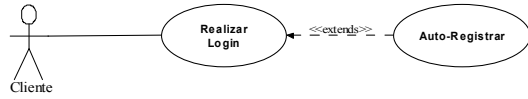
11

Extensão

Esta associação é utilizada pra descrever cenários opcionais de um Caso de Uso que somente ocorrerão se uma determinada condição for satisfeita. As associações de Extensão possuem uma representação muito semelhante às associações de Inclusão sendo também representada por uma reta tracejada, diferenciando-se por possuir um Estereótipo contendo o texto "<< extend >> ", e pela seta apontar para o Caso de Uso que estende.

12

Extensão



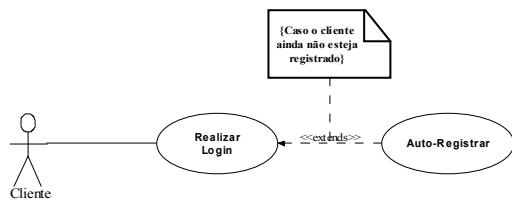
13

Restrições em Associações de Extensão

Restrições são compostas por um texto entre chaves e são utilizados para definir validações, consistências, condições etc., que deve ser aplicada à um determinado componente ou situação. Às vezes, nem sempre fica claro qual é a condição para que um Caso de uso estendido executado, assim, pode-se acrescentar uma Restrição à Associação de extensão por meio de uma nota explicativa, determinando a condição para que o Caso de Uso seja executado.

14

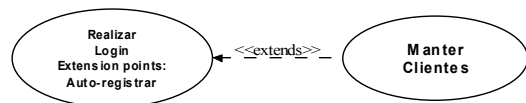
Restrições em Associações de Extensão



15

Pontos de Extensão

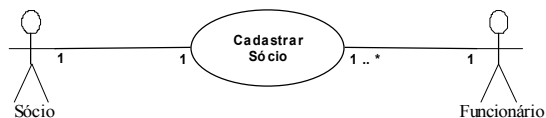
Podem ocorrer situações em que somente parte do comportamento de um Caso de Uso estendido seja utilizado pelo Caso de uso que estendeu. Nestes casos são utilizados Pontos de Extensão que especificam qual comportamento está sendo estendido.



16

Multiplicidade

A Multiplicidade em uma Associação entre um Ator e um Caso de Uso basicamente especifica o número de vezes que um Ator pode utilizar um determinado Caso de Uso.



17

Diagrama de Classes

O Diagrama de Classes é o diagrama mais utilizado da UML. Seu objetivo é permitir a visualização das classes utilizadas pelo sistema e como se relacionam. Esse diagrama apresenta uma **visão estática** de como as classes estão organizadas, preocupando-se em definir sua estrutura lógica.

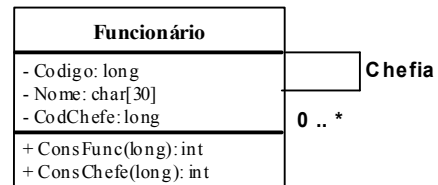
18

Relacionamento ou Associações

As classes costumam possuir relacionamento entre si, chamados Associações que as permite compartilharem informações e colaborarem para a execução dos processos executados pelo sistema. Uma associação descreve um vínculo que ocorre normalmente entre os objetos de uma ou mais classes. As associações podem também possuir títulos para determinar o tipo de vínculo estabelecido entre os objetos das classes, isto é útil quando é necessária alguma forma de esclarecimento, neste caso é recomendável determinar também a Navegabilidade da Associação para facilitar o sentido da leitura.

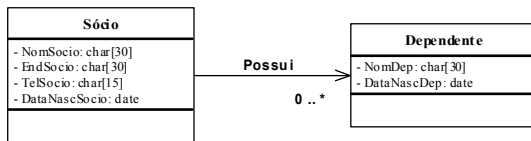
Associação Unária ou Reflexiva

Este tipo de associação ocorre quando existe um relacionamento de um objeto de uma classe com objetos da mesma classe.



Associação Binária

Associações binárias ocorre quando são identificados relacionamentos entre objetos de duas classes.



Associação Ternária ou N-ária

Associações Ternárias ou N-árias são associações que conectam objetos de mais de duas classes. São representadas por um losângulo para convergem todas as ligações da Associação.



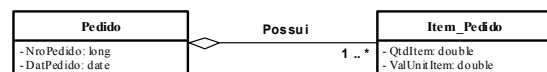
Agregação

Agregação é um tipo de Associação onde tenta-se demonstrar que as informações de um objeto (chamado objeto-todo) precisam ser complementadas pelas informações contidas em um ou mais objetos de outra Classe (chamados objeto-partes).

Este tipo de associação tenta demonstrar uma relação Todo/Parte entre os objetos associados. Objetos-parte não podem ser destruídos por um objeto diferente do objeto-todo. O símbolo de Agregação difere do de Associação por conter um losângulo na extremidade da classe que contém os objetos-todo.

A agregação simples é inteiramente conceitual e nada faz além de diferenciar o “todo” da “parte”. A agregação simples não modifica o significado da navegação pela associação entre o todo e suas partes, nem vincula o tempo de vida do todo e de suas partes.

Agregação



Composição

Esta Associação é uma variação da Agregação, onde é apresentado um vínculo mais forte entre os objetos-todo e os objetos-parte, procurando demonstrar que **os objetos-parte têm de estar associados a um único objeto-todo**. O símbolo de Composição diferencia-se graficamente do símbolo de Agregação por utilizar um losângulo preenchido. Da mesma forma que na Agregação o losângulo deve ficar ao lado do objeto-todo.

A composição é uma forma de agregação, com propriedade bem-definida e tempo de vida coincidente como parte do todo. As partes sem multiplicidade fixada podem ser criadas após a composição, mas, uma vez criadas, vivem e morrem com ela. Essas partes também podem ser removidas explicitamente antes da morte do objeto composto.

Composição



Agregação Composta

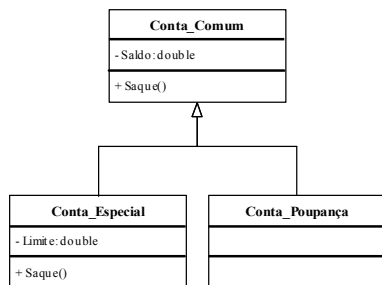
Em uma Agregação composta, um objeto poderá ser uma parte de somente uma composição em um determinado momento. Por exemplo, em um sistema de janelas, uma **Moldura** pertence exatamente a uma única **Janela**. Isso está em contraste com uma agregação simples, em que uma parte pode ser compartilhada por vários todos. Por exemplo, no modelo de uma casa, uma **Parede** pode ser parte de um ou mais objetos **Cômodos**.

Além disso, em uma agregação composta, o todo é responsável pela disposição de suas partes, significando que **o objeto composto deve gerenciar a criação e a destruição de suas partes**. Por exemplo, ao criar uma **Moldura** em um sistema de janelas, você deve anexa-la a uma **Janela** que a conterà. Da mesma maneira, quando você destrói a **Janela**, por sua vez, deve destruir a **Moldura**.

Especialização/Generalização

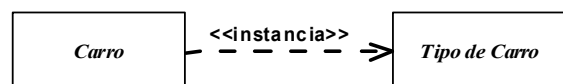
Esta Associação identifica **superclasses**, chamadas **gerais** e **classes-filha**, chamadas **especializadas**, demonstrando a ocorrência de heranças e possivelmente de métodos polimórficos nas classes especializadas.

Especialização/Generalização



Dependência

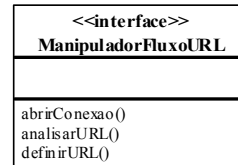
Este relacionamento, como o próprio nome diz identifica um certo grau de dependência de uma classe em relação à outra. O relacionamento de dependência é representado por uma reta tracejada contendo uma seta apontando a classe da qual a classe posicionada na outra extremidade do relacionamento é dependente.



Interface

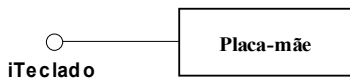
Uma interface é uma coleção de operações empregadas para especificar um serviço de uma classe ou de um componente. Ao contrário das classes e dos tipos, as interfaces não especificam qualquer implementação (portanto, poderão não incluir métodos, os quais fornecem a implementação de uma operação). Assim como as classes, uma interface poderá ter qualquer número de operações. Essas operações poderão receber adornos com propriedades de visibilidade, propriedades de concorrência, estereótipos, valores atribuídos e restrições.

Interface



Interface Fornecida

Uma Interface Fornecida descreva um serviço implementado por uma classe. O conjunto de interfaces implementadas por uma classe são suas Interfaces Fornecidas e representam o conjunto de serviços que a classe ao implementar uma interface suporta o conjunto de características possuídas por esta e obedece às suas restrições.



Interface Requerida

Descrevem os serviços que outras classes devem fornecer a uma determinada classe, que não precisa ter conhecimento de quais classes irão implementar estes serviços.



Interface Requerida

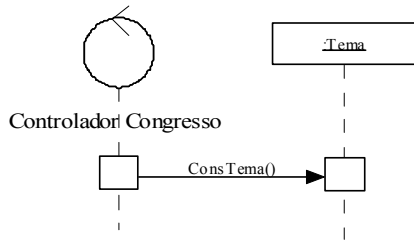
Muitas vezes é comum uma Interface Fornecida em uma classe ser uma Interface Requerida em outra, podendo facilmente ocorrer de ambas as interfaces surgirem juntas.



Diagrama de Seqüência

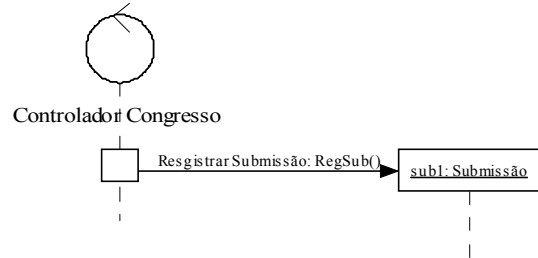
- Diagrama de Seqüência ilustra como os objetos interagem entre si
- Enfocando a seqüência em que as mensagens são enviadas e recebidas entre vários objetos
- Os DS também revelam as interações entre objetos durante um determinado intervalo de tempo na execução de um sistema
- Os DS são modelos que descrevem como grupos de objetos colaboram em algum comportamento

Mensagem – somente método



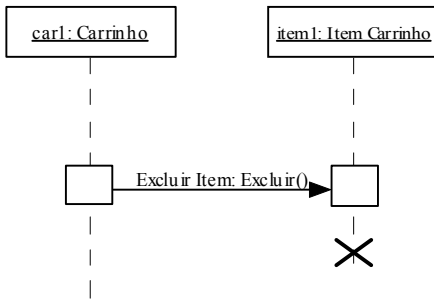
37

Mensagem – evento e método



38

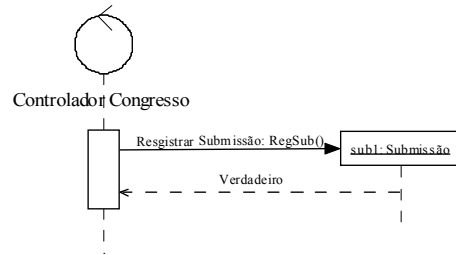
Mensagem – método destrutor



39

Mensagem de retorno

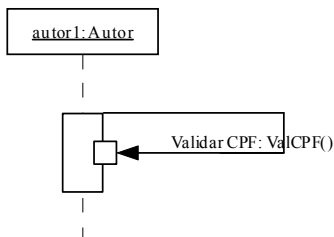
Este tipo de mensagem identifica a resposta a uma mensagem para o objeto que a chamou. Uma mensagem de retorno pode retornar informações específicas do método chamado ou apenas um valor indicado se o método for executado com sucesso ou não.



40

Auto-chamada ou auto-delegação

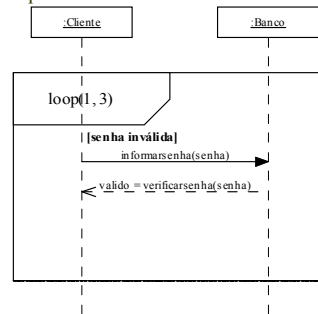
São mensagens que partem da linha de vida objeto e atinge a linha de vida do próprio objeto.



41

Condições ou Condições de guarda

Estabelecem uma regra ou condições para que uma mensagem possa ser disparada.



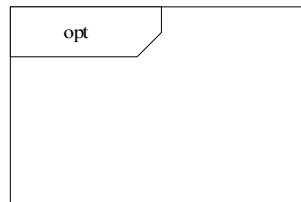
42

Controle Estruturado

Uma seqüência de mensagens é boa para mostrar uma única seqüência linear, mas freqüentemente precisamos mostrar condicionais e loops. Às vezes, é preciso mostrar a execução concorrente de várias seqüências. O tipo de controle de alto nível pode ser apresentado com operadores de controle estruturado nos diagramas de seqüência.

Execução Opcional

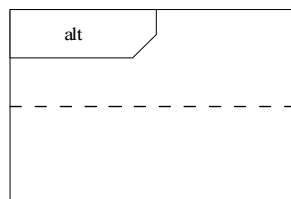
A tag é **opt**. O corpo do operador de controle é executado se uma condição de proteção for verdadeira na entrada do operador. A condição de proteção é uma expressão booleana que pode aparecer entre colchetes na parte superior de qualquer linha da vida no corpo, e pode fazer referência a atributos desse objeto.



Execução Condicional

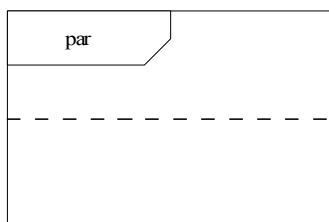
A tag é **alt**. O corpo do operador de controle é dividido em várias sub-regiões, por linha horizontais tracejadas. Cada sub-região representa um ramo de uma condicional. Cada sub-região tem uma condição de proteção. Se a condição de proteção de uma região for verdadeira, a sub-região será executada. Entretanto, no máximo uma sub-região pode ser executada; se mais de uma condição de proteção for verdadeira, a escolha de sub-região não será determinística, podendo variar de execução para execução. Se nenhuma condição de proteção for verdadeira, o controle continua passando pelo operador de controle. Uma sub-região pode ter uma condição de proteção especial **[else]**; essa sub-região será executada, se nenhuma das outras condições de proteção for verdadeira.

Execução Condicional



Execução Paralela

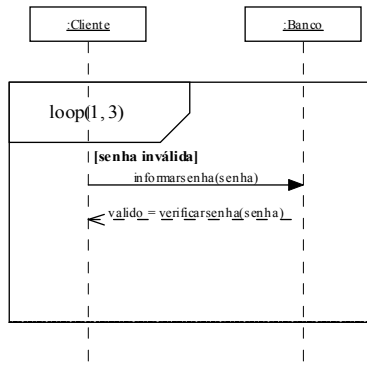
A tag é **par**. O corpo do operador de controle é dividido em várias sub-regiões por linhas horizontais tracejadas. Cada sub-região representa uma **computação paralela** (concorrente).



Execução de loop (Iterativa)

A tag é **loop**. Uma condição de proteção aparece na parte superior de uma linha da vida no corpo. O corpo do loop é executado repetidamente enquanto a condição de proteção é verdadeira, antes de cada iteração. Quando a condição de proteção é falsa na parte superior do corpo, o controle sai do operador de controle.

Faculdade Pilares - Bacharelado em Informática
Execução de loop (iterativa)



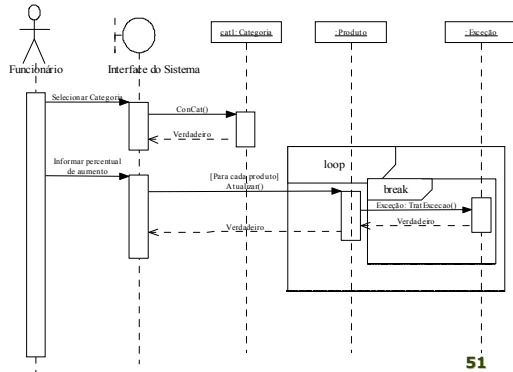
49

Faculdade Pilares - Bacharelado em Informática
Quebra (Break)

Este Operador de Interação indica uma " quebra " na execução do processo. É usado principalmente para modelar o **tratamento de exceções**.

50

Faculdade Pilares - Bacharelado em Informática
Quebra (Break)



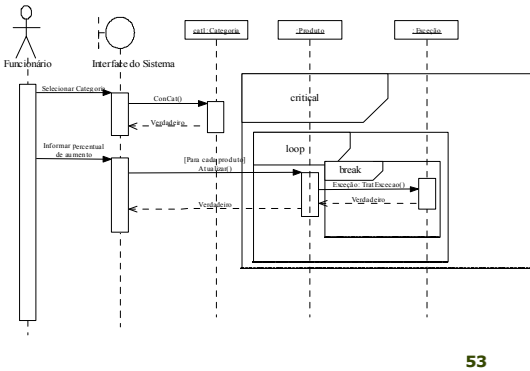
51

Faculdade Pilares - Bacharelado em Informática
Região Crítica (Critical Region)

Este operador de interação identifica uma operação atômica que não pode ser interrompida por outro processo até ser totalmente concluída.

52

Faculdade Pilares - Bacharelado em Informática
Região Crítica (Critical Region)



53

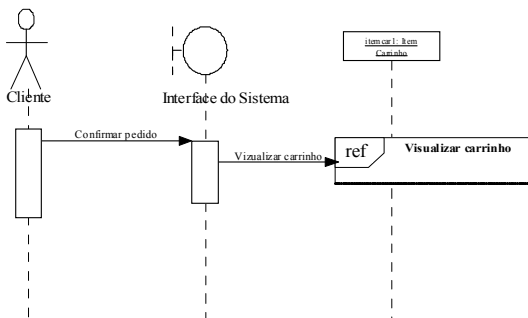
Faculdade Pilares - Bacharelado em Informática
Fragmentos de Interação

Os Fragmentos de Interação são noções abstratas de unidades de interação geral. Um fragmento de interação é uma parte de uma interação, no entanto, cada fragmento de interação é considerado uma interação independente.

Uma das principais vantagens do uso de fragmentos de Interação caracteriza-se pela possibilidade de se poder referenciá-los por meio do operador **Ref**, que é a abreviatura de **Referred** (referido) e significa que se deve procurar por um diagrama cujo nome é o mesmo do nome apresentado após o operador **Ref**, ou seja, o fragmento faz referencia a outro diagrama, não, detalhado no diagrama em questão e que deve ser inserido neste. A isto chama-se **Ocorrência de Interação** e esta inovação permite que se montem diagramas mais complexos que fazem referencia a outros diagramas como se fossem sub-rotinas, detalhadas em separado.

54

Faculdade Pilares - Bacharelado em Informática
Fragmentos de Interação



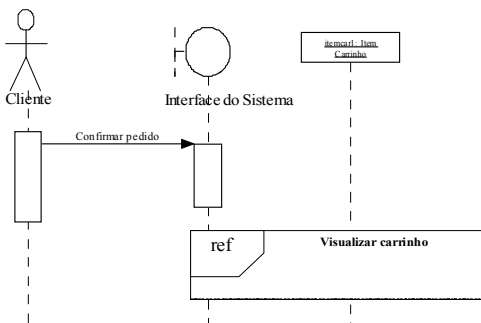
55

Faculdade Pilares - Bacharelado em Informática
Fragmentos de Interação

É bastante comum encontrar Ocorrências de Interação simplesmente sobrepostas às linhas de vida dos objetos que fazem parte dos do processo, sem nem ao menos chamá-las por meio de uma mensagem, como se as instruções contidas nas Ocorrências de Interação fossem adicionadas automaticamente ao diagrama.

56

Faculdade Pilares - Bacharelado em Informática
Fragmentos de Interação



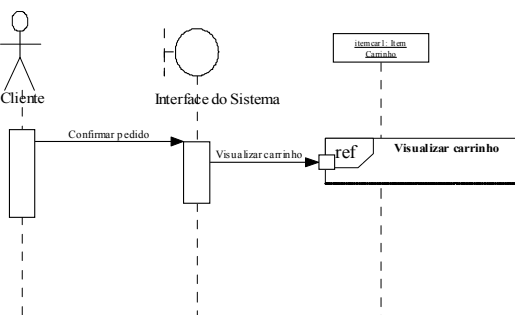
57

Faculdade Pilares - Bacharelado em Informática
Portões (Gates)

Um portão é uma interface entre fragmentos, um ponto de conexão para relacionar uma mensagem fora de um fragmento de interação com uma mensagem dentro do fragmento de interação. Portões são representados simplesmente pelo encontro da seta da mensagem no retângulo do fragmento de interação, ou, em algumas ferramentas CASE, por pequenos quadrados posicionados na linha vertical do retângulo do fragmento, na altura em que a mensagem deveria atingir a linha de vida do objeto a que ele se refere. O propósito de um Portão é simplesmente estabelecer quem está enviando e quem está recebendo uma mensagem, quando esta mensagem não está contida em único fragmento de interação.

58

Faculdade Pilares - Bacharelado em Informática
Portões (Gates)



59