PUC-Rio

Tópico 6:
Diagrama de Sequência – Parte 1

Luiz Antônio M. Pereira

lpereira@uninet.com.br

lpereira@luizantoniopereira.com.br

Colaborar é Preciso

- Em uma organização, os processos são tipicamente executados por indivíduos especializados, que exercem suas responsabilidades e atuam de forma colaborativa:
  - Trocam informações (conversas telefônicas e na copa, no cafezinho, e-mails, sinais visuais etc.);
  - Trocam objetos tangíveis (documentação impressa, informação em CD etc.).

Colaborar é Preciso

- Nas organizações maduras, a colaboração se dá de forma organizada, conforme a formação dos indivíduos;
- A forma é estabelecida na cultura da organização.

Colaborar é Preciso

- A cultura da organização é resultante observações e análises de especialistas e analistas do negócio e determinações de seus gestores;
- Essa cultura é documentada em manuais de procedimentos.

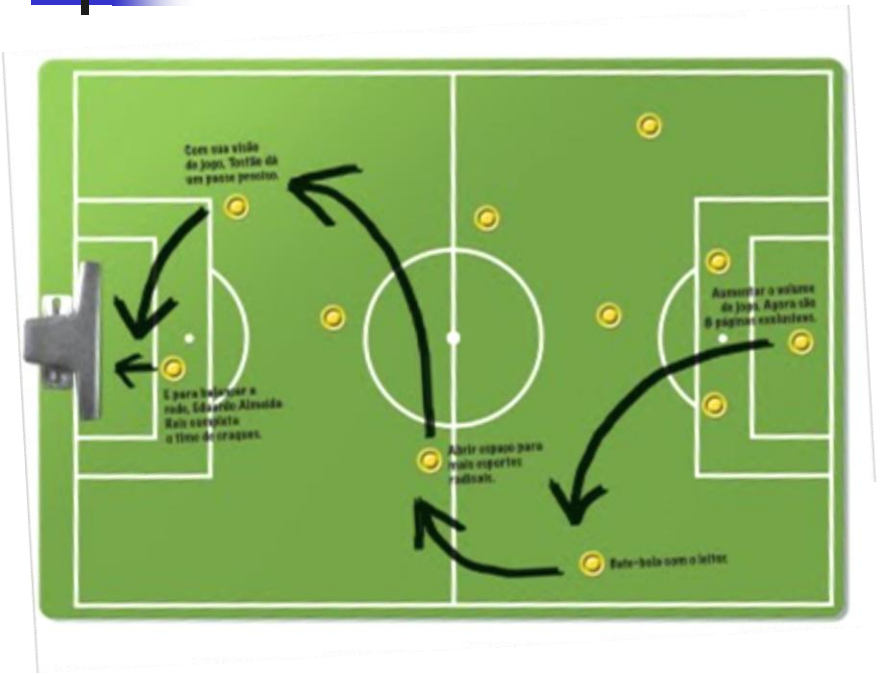
Colaborar é Preciso

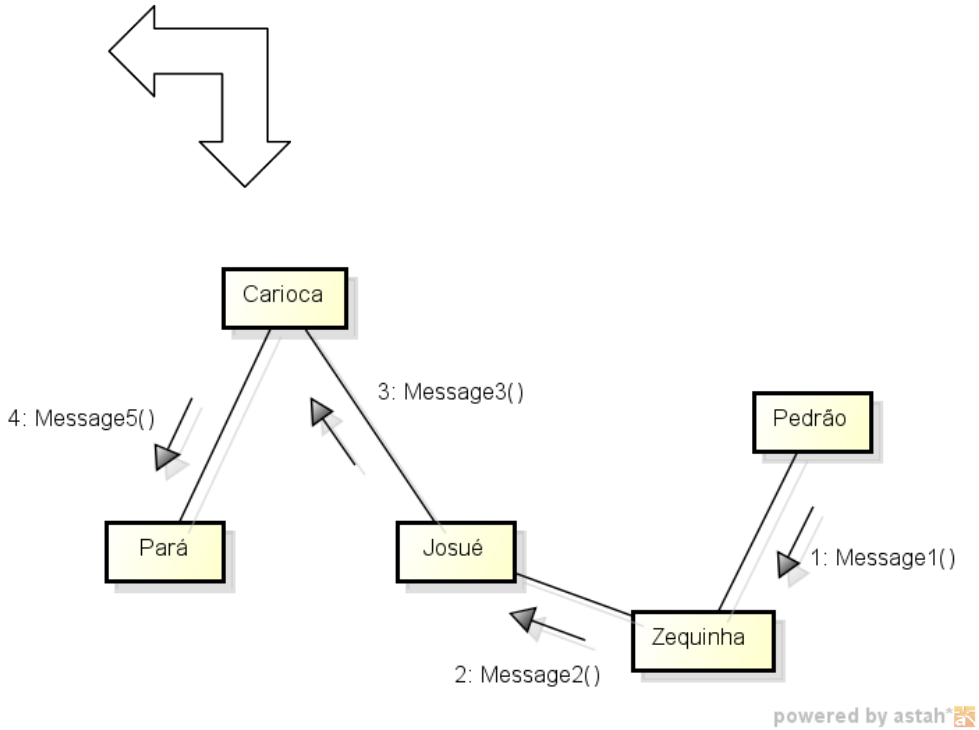
- Sistemas desenvolvidos segundo a OO entendem que objetos colaboram para a realização dos objetivos do sistema.
  - Objetos têm responsabilidades;
  - Objetos executam operações segundo sequências definidas pelos projetistas.

Colaborar é Preciso

Processos de Negócio	Programas OO
Indivíduos especializados	Objetos
Nome de um indivíduo	Identificador de um objeto
Profissões/especialidades dos indivíduos	Classes de objetos
Trocas de informações e sinais	Chamadas de operações
Alocação de um profissional de uma profissão específica a uma tarefa	*Instanciação* (criação) de um objeto de uma classe específica na memória
Operações que um profissional executa para realizar suas responsabilidades	Métodos (ou operações) que compõem a programação de um objeto
Gestores e especialistas no negócio	Analistas e programadores
Manual de procedimentos da organização	Diagramas de interação

Colaborar é Preciso



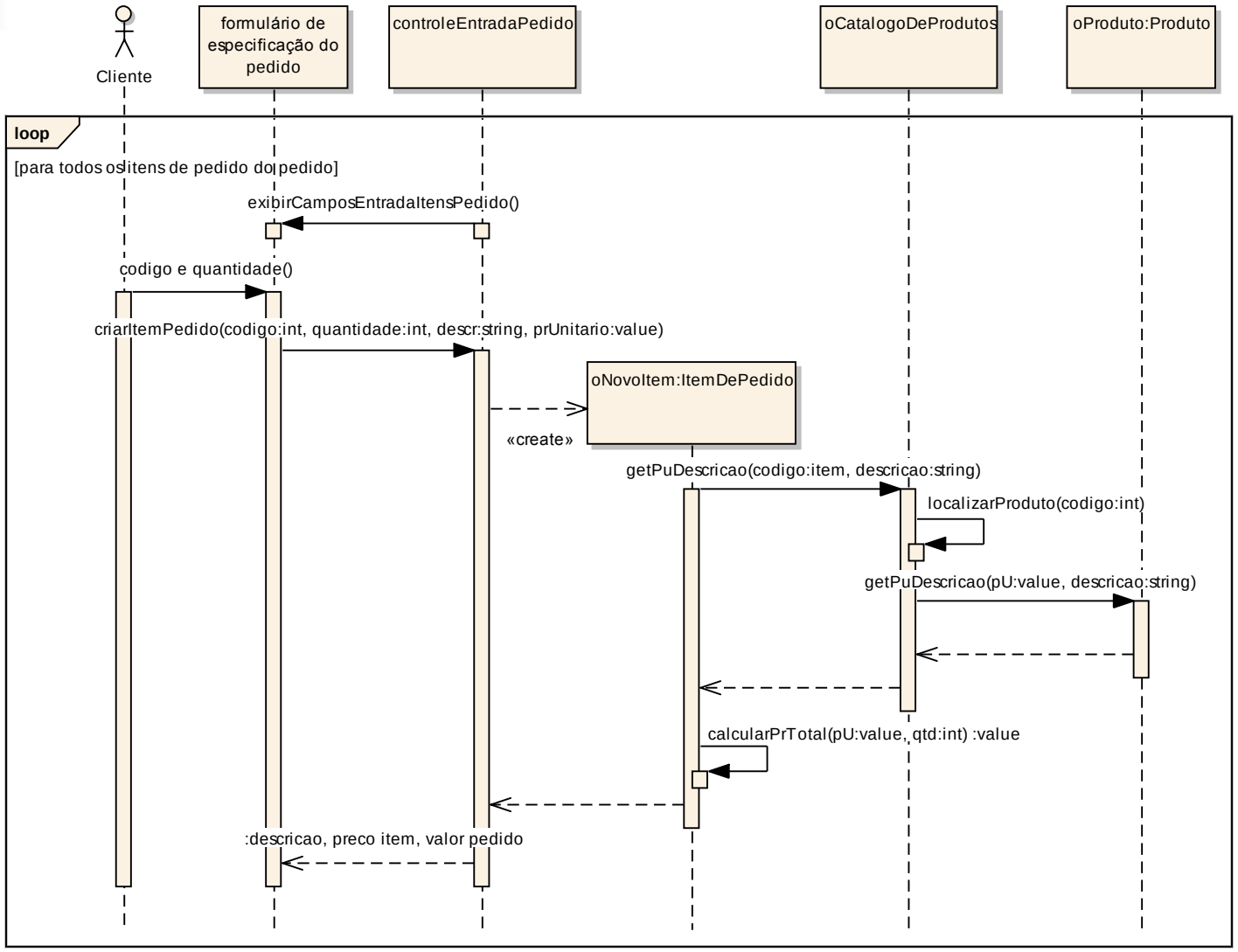


Diagramas de Interação da UML

- Diagrama de Temporização
- Diagrama de Comunicação
- Diagrama de Sequência
- Diagrama de Visão Geral da Interação

PUC-Rio

Diagrama de Sequência



Diagramas de Sequência

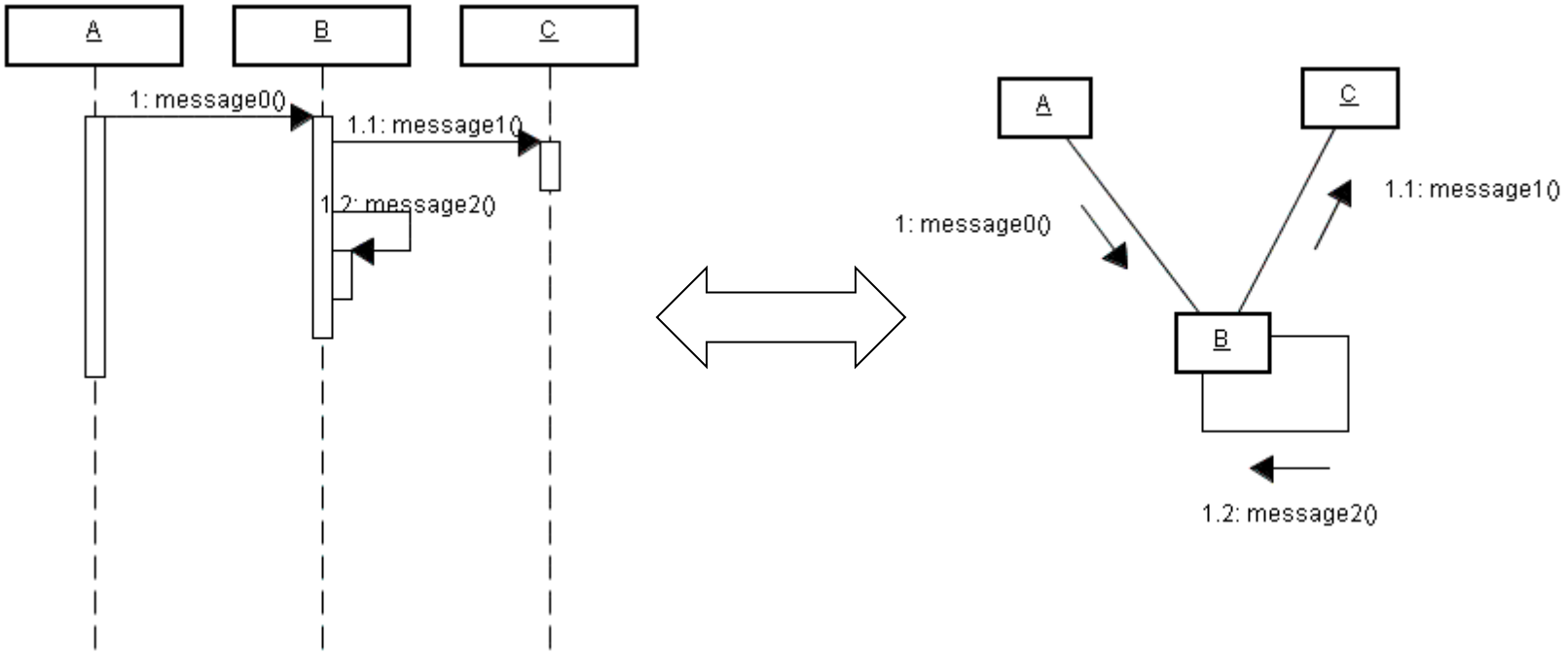
- Diagramas de sequência descrevem como grupos de objetos colaboram em algum comportamento do sistema, ou seja, como os objetos colaboram para realizar alguma função do sistema.

Diagramas de Sequência

- Usados para a atribuição de responsabilidades aos objetos do sistema;
- Úteis para a descoberta de operações dos objetos e para a modelagem da interação entre eles;
- Servem, portanto, para modelar o "funcionamento" do sistema;

Diagramas de Sequência

- Melhores que o Diagrama de Comunicação para apresentar as responsabilidades de cada objeto, especialmente quando o aspecto da ordenação temporal é relevante.



Diagramas de Sequência

- Podemos gerar código automaticamente a partir de um DS (engenharia direta);
- Boas ferramentas CASE também realizam engenharia reversa (código $\rightarrow$ DS).

Cenários

- Um diagrama de sequência tipicamente captura o comportamento de um único cenário de um caso de uso, pois,
- DSs ficam muito complicados visualmente quando representam múltiplos cenários.

→Usar mais de um diagrama por caso de uso (um para cada cenário).

Cenários

Definição de cenário:

- Instância de um caso de uso;
- Um caminho único em um caso de uso;
- Um fluxo de informação;
- Um fluxo relacionado com o objetivo com começo, meio e fim;

Cenários

Cenários podem ser:

- OTIMISTAS : quando tudo dá certo
- EXCEÇÃO : erros e casos de falha
- ALTERNATIVOS : opções de sequência

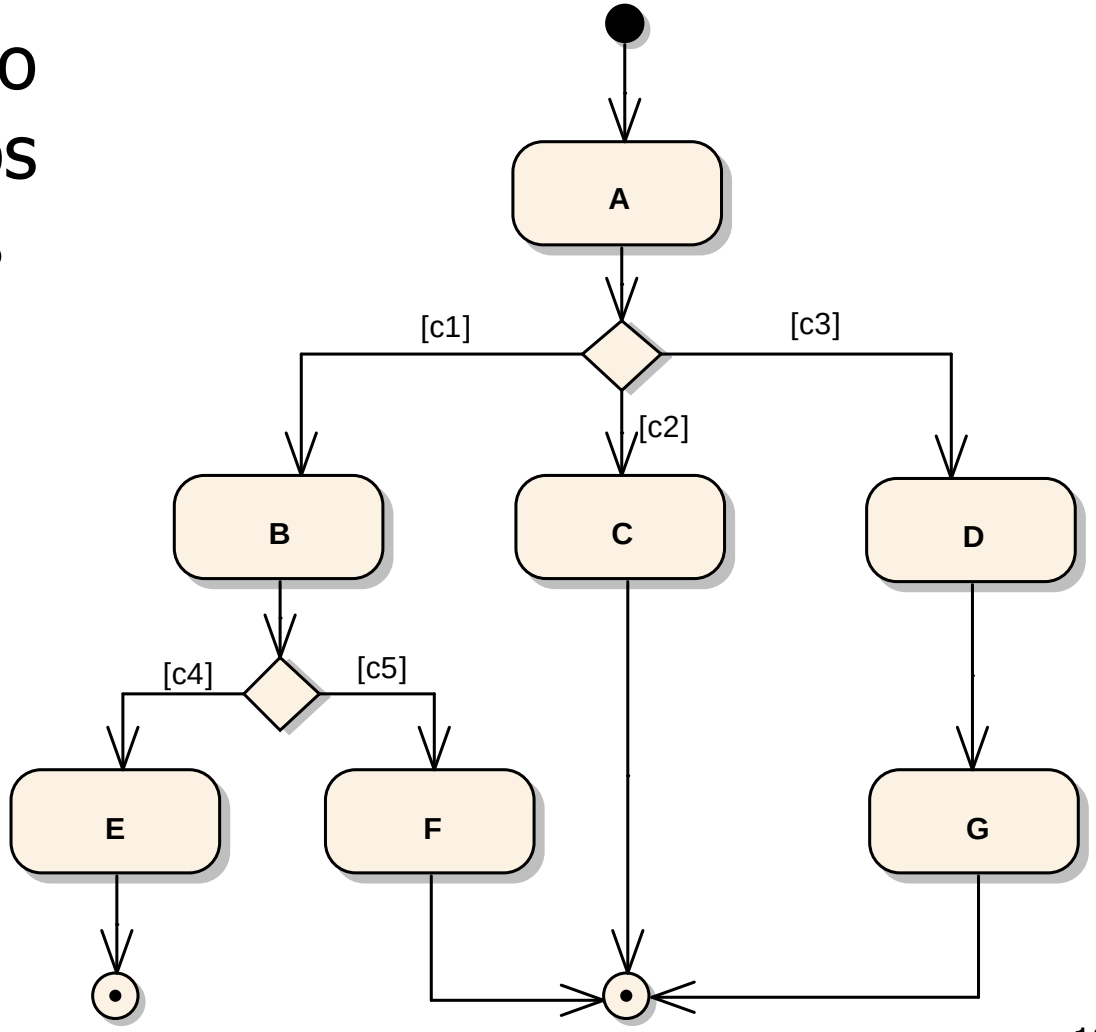
Cenários

Exemplos de cenários:

- O cliente tem o dinheiro e saca o que quer;
- O cliente não tem o dinheiro, mas pode usar o limite do cheque especial e saca o que quer;
- O cliente não tem o dinheiro todo e saca o que pode;
- O cliente não tem nada em conta, não tem cheque especial e nada saca;
- O cliente tem o dinheiro em conta, mas não saca o que quer porque já passou das 22h.

Cenários

- Casos de uso especificados por DAs nos ajudam a identificar cenários.



Cenários

- O curso típico de um caso de uso caracteriza um cenário;
- Os demais cenários podem ser caracterizados pelas ações executadas ou pelas condições que determinam os desvios.

O Ciclo de Vida dos Objetos

- O ciclo de vida de um objeto compreende tudo que acontece com ele durante o tempo decorrido entre a sua instanciação e a sua destruição;
- Objetos de dois tipos:
  - Persistentes;
  - De vida efêmera.
- Ambos precisam estar na memória para colaborar.

O Ciclo de Vida dos Objetos

- Objetos persistentes:
  - Tipicamente instâncias de classes conceituais;
  - Sobrevivem à desativação do sistema;
  - Iniciam o ciclo de vida quando são instanciados;
  - Morrem quando são removidos do BD.
- De vida efêmera:
  - Tipicamente instâncias de *classes utilitárias*;
  - Instanciados para colaborar em algum ponto da interação;
  - São descartados em seguida.

PUC-Rio

C

C E

Responsabilidades X Atributos e Operações dos Objetos

- Engenheiros, contadores, etc., têm responsabilidades em uma organização;
- Classes têm responsabilidades em um sistema.
- Responsabilidades são exercidas por meio de
  - Operações;
  - Atributos.

PUC-Rio

Responsabilidades X Atributos e Operações dos Objetos

- Responsabilidades atribuídas aos objetos devem ser baseadas na capacidade deles as exercerem:
  - Têm todos ou parte dos atributos?
  - Têm todas ou parte das operações?
  - Podem delegar toda ou parte das responsabilidades?

PUC-Rio

Responsabilidades X Atributos e Operações dos Objetos

- Perguntas:
  - Qual é o sinal de que pedimos a um colaborador algo que ele não é o mais indicado a executar?
  - Qual é o sinal de que pedimos a um objeto algo que ele não é o mais indicado a executar?

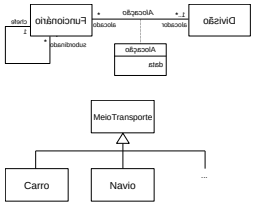
O Tripé da Análise

- DSs completam o tripé da análise:
  - Casos de Uso - comportamento externo (funcional)
  - Diagramas de Classes - visão estática
  - Diagramas de Sequência - visão dinâmica

} Internos

O Tripé da Análise

Diagrama de Classes



Operações (4)

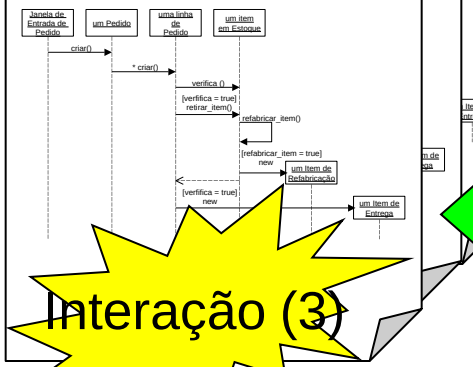
Caso de Uso

Nome: Reservando Passagem
Atores: Agente
Curso de Eventos
1- Ujfsaj jfklsdj jfdkkj fl als ;a f a;
2- jfaskdjf lj kl;k kdfjasdkl lkssss
3- jsdkfklk lkkkk lopjfa[pokfsao opw
4- skdjfl)kkk;'PIO lkkfapp kjadfp
5- lkLKO oeppae fokkzp;xp pokf ;lp[
Alternativas
1- ljfksa kJFKJ a;lkj ;kjfklasojk;a

Diagrama de Colaboração

Diagrama de Atividade

Diagrama de Sequência



Objetos (2)

Cenário (1)

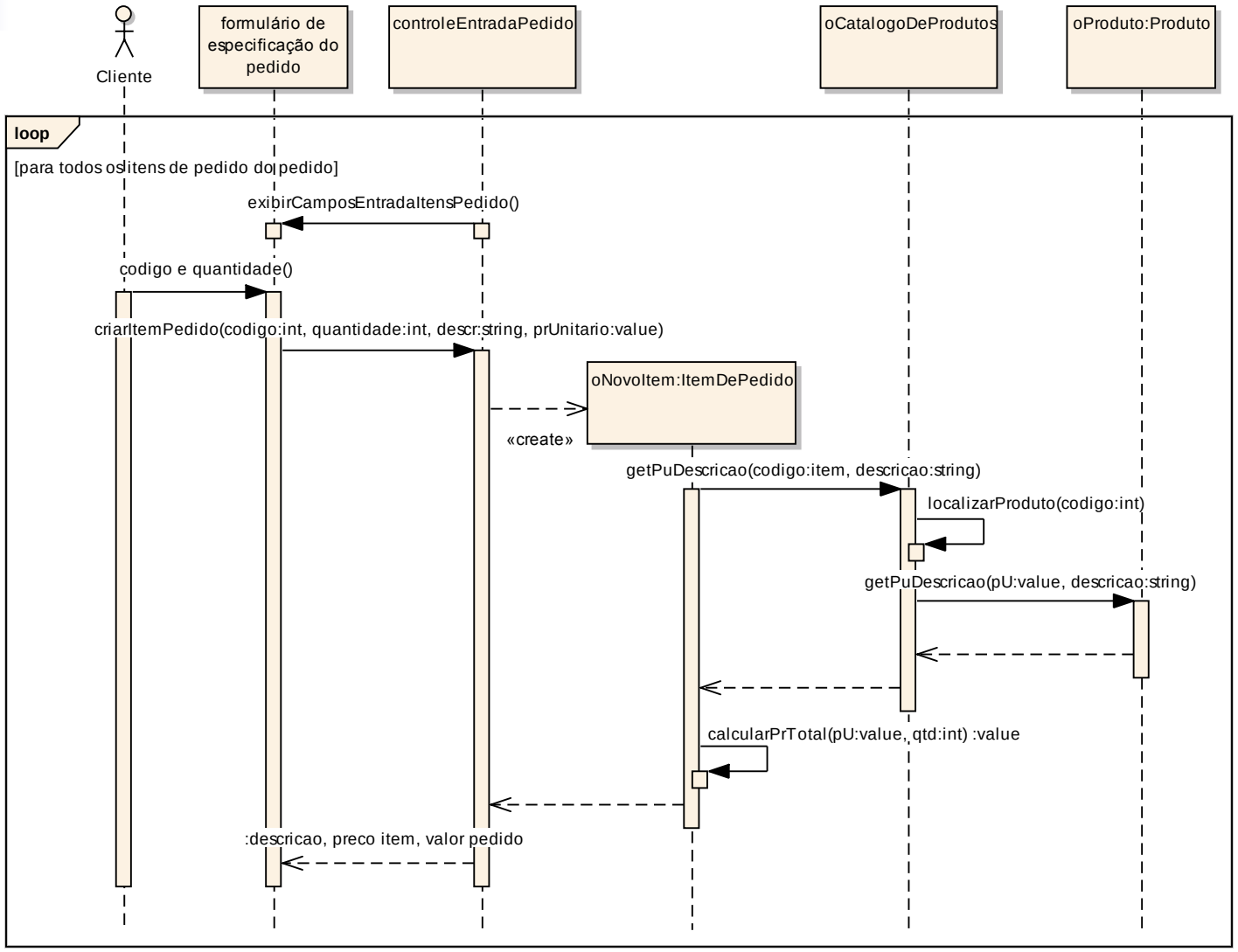
Interação (3)

PUC-Rio

Dimensões Horizontal e Vertical

H

V

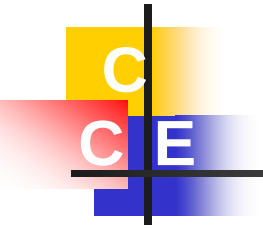


Dimensões Horizontal e Vertical

- Dimensão Horizontal:
  - Onde relacionamos os objetos que participam da colaboração.
- Dimensão Vertical:
  - Representa a passagem do tempo.

Nível de Detalhamento dos Diagramas de Sequência

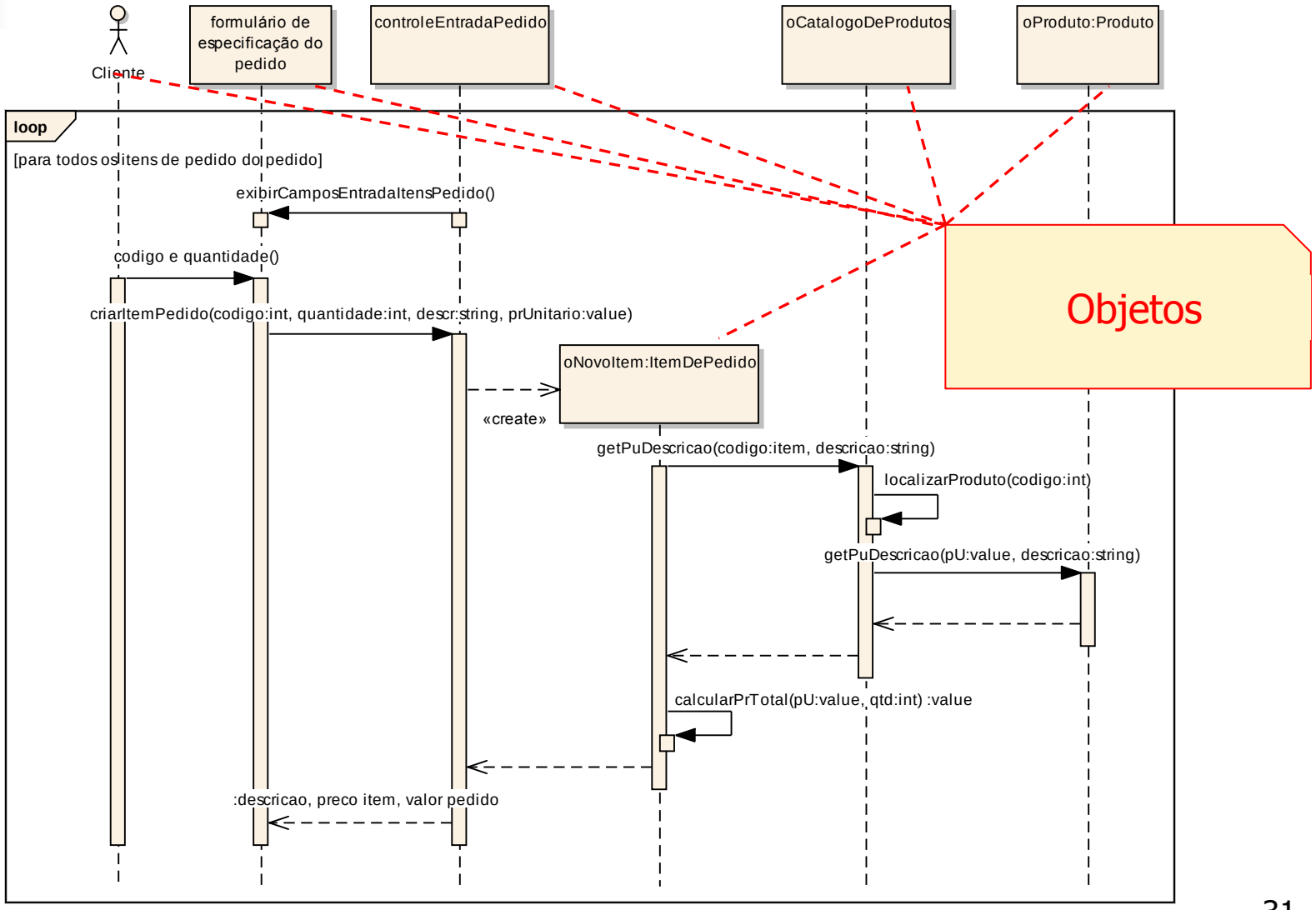
- Definir objetivo: DS para documentação ou geração de código?
  - Detalhamento prejudica compreensão;
  - Alto nível de abstração se contrapõe à geração de código.
- DSs nos aguçam a busca por detalhes
  - Cuidado com a perspectiva!



Elementos Básicos da Notação

PUC-Rio

Objetos



Diagramas de Sequência

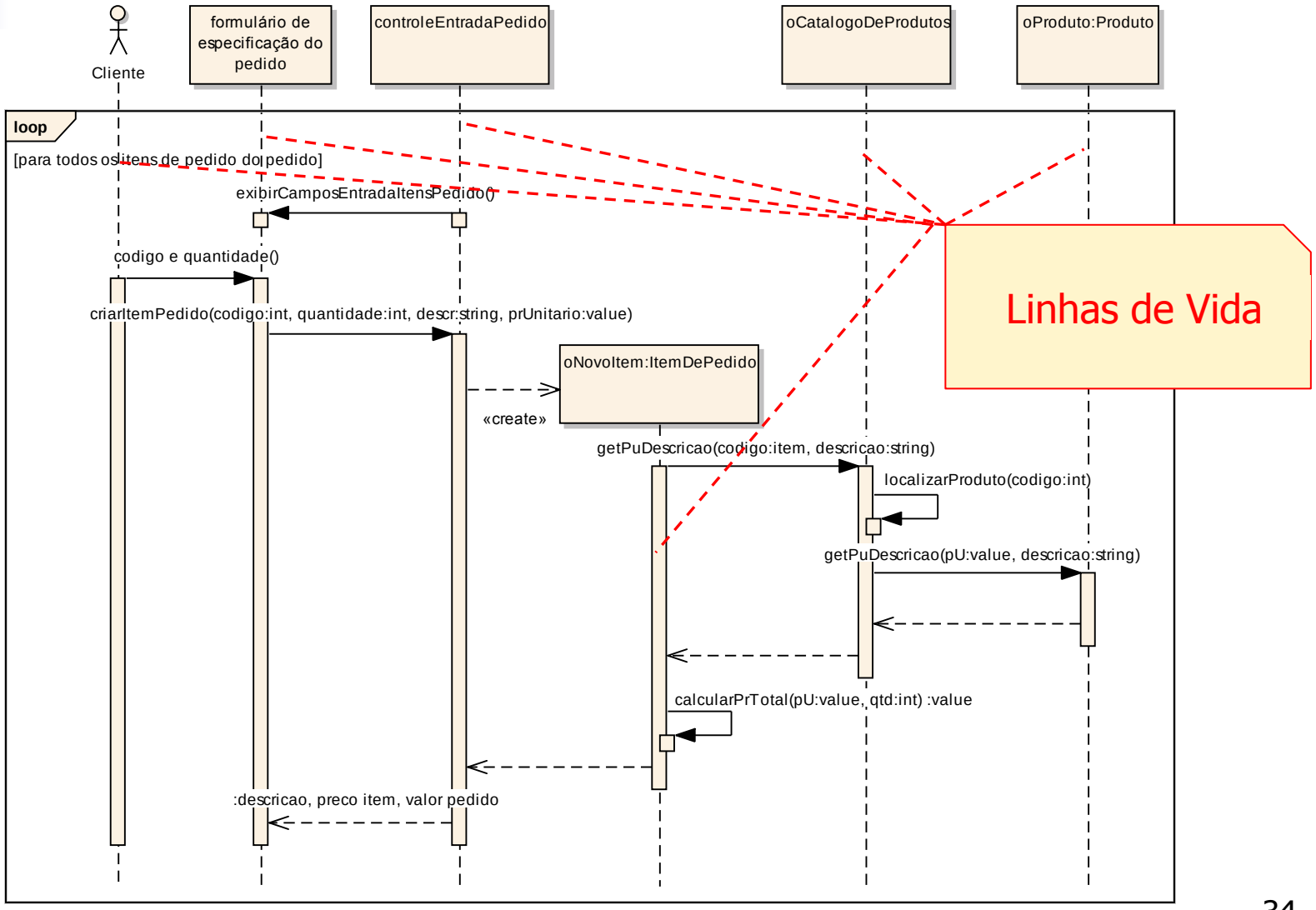
- Objetos compõem a *dimensão horizontal* ($\rightarrow$) ;
- Ordem da colocação dos objetos na dimensão horizontal não tem significado.

Diagramas de Sequência

- Objetos possuem nome: *obj*:*Classe*
  - O sublinhado denota instanciação (objeto = instância de classe) sendo, portanto, obrigatório;
  - *Obj* OU (exclusivo) *classe* pode ser omitido;
  - Tipicamente: um(a)*Classe*.
  - Ex.: umPedido, umItemPedido, umFuncionário, :Funcionário, :Pedido, Joaquim:Funcionário, etc.
- O sublinhado deixou de ser obrigatório na versão 2

PUC-Rio

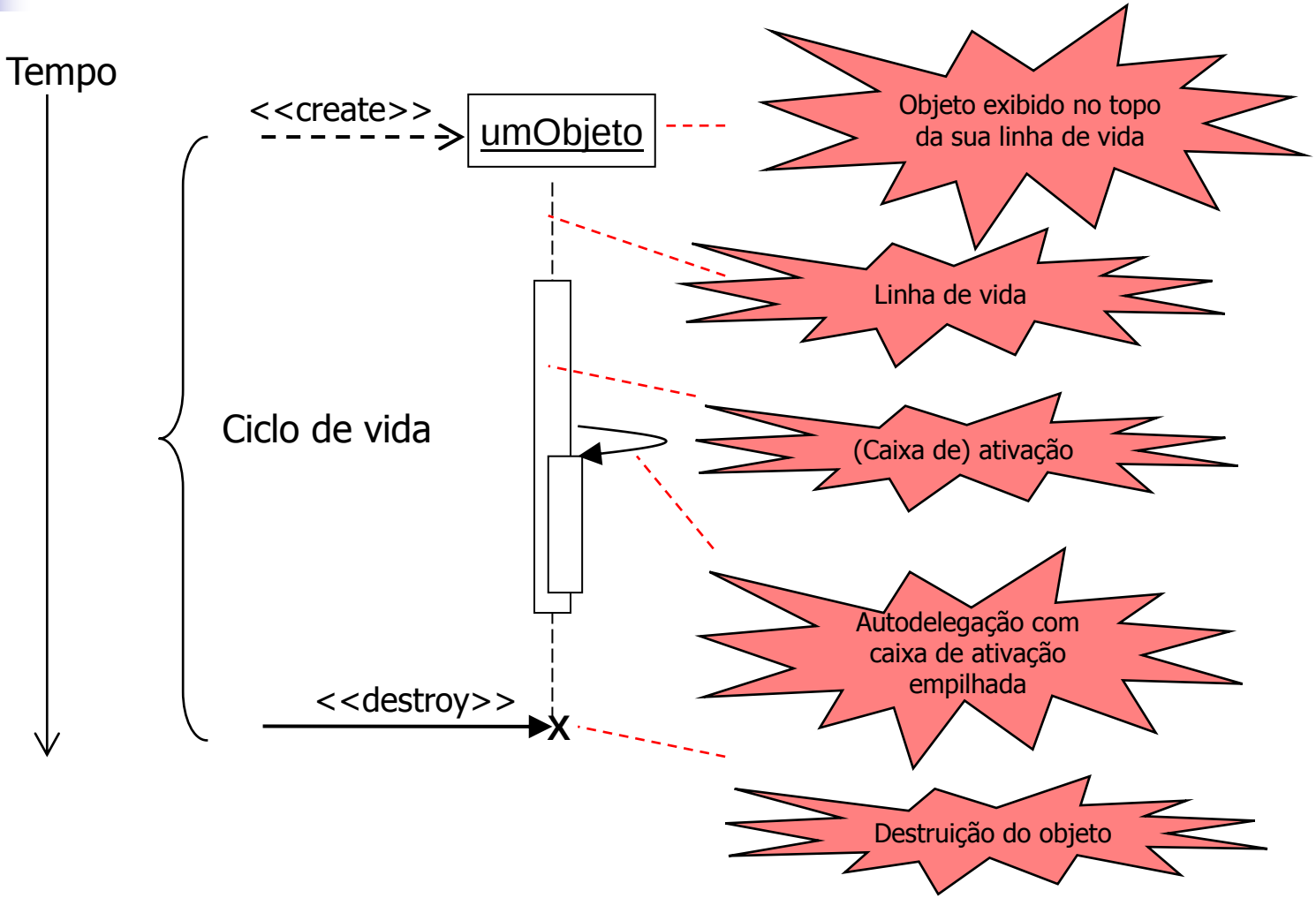
Linha de Vida



Linha de Vida

- Linhas de vida compõem a dimensão vertical (tempo);
- Dimensão vertical = sequência, onde a vida do objeto durante a interação é representada;
- Pode apresentar a ativação e a desativação de objetos (foco de controle) por meio das caixas de ativação;
- Pode representar criação de objetos e a destruição de objetos

Linha de Vida



Linha de Vida

- Caixas de ativação denotam que o objeto está executando algo (foco de controle) diretamente ou através de outro objeto;
- Caixas de ativação são opcionais (diagramas ficam mais fáceis de se desenhar), mas a ausência dificulta a leitura do diagrama em certos casos;
- A atividade sendo executada pode ser descrita em texto na margem esquerda do diagrama, dentro dos limites da ativação;
- Caixas de ativação podem empilhar-se para denotar chamadas recursivas.

Antes de Prosseguirmos...

A alocação de memória em poucas palavras

- Objetos "moram" na memória.
- A memória é de dois tipos:
  - Principal
  - Secundária
- Objetos se comunicam se estiverem na memória principal.

Antes de Prosseguirmos...
A alocação de memória em poucas palavras

- A alocação de um objeto na memória principal é feita por uma chamada a uma função do SO;
- O SO responde a essa chamada de duas formas:
  - Não há espaço disponível na memória principal para a alocação do objeto; ou
  - O espaço foi alocado e o endereço é *tal*.
- O SO aloca os espaços solicitados de uma forma própria, segundo uma lógica não facilmente compreensível, "aparentemente" sem um padrão.
- Com isso, de um modo geral, objetos não ficam "arrumadinhos" na memória principal.

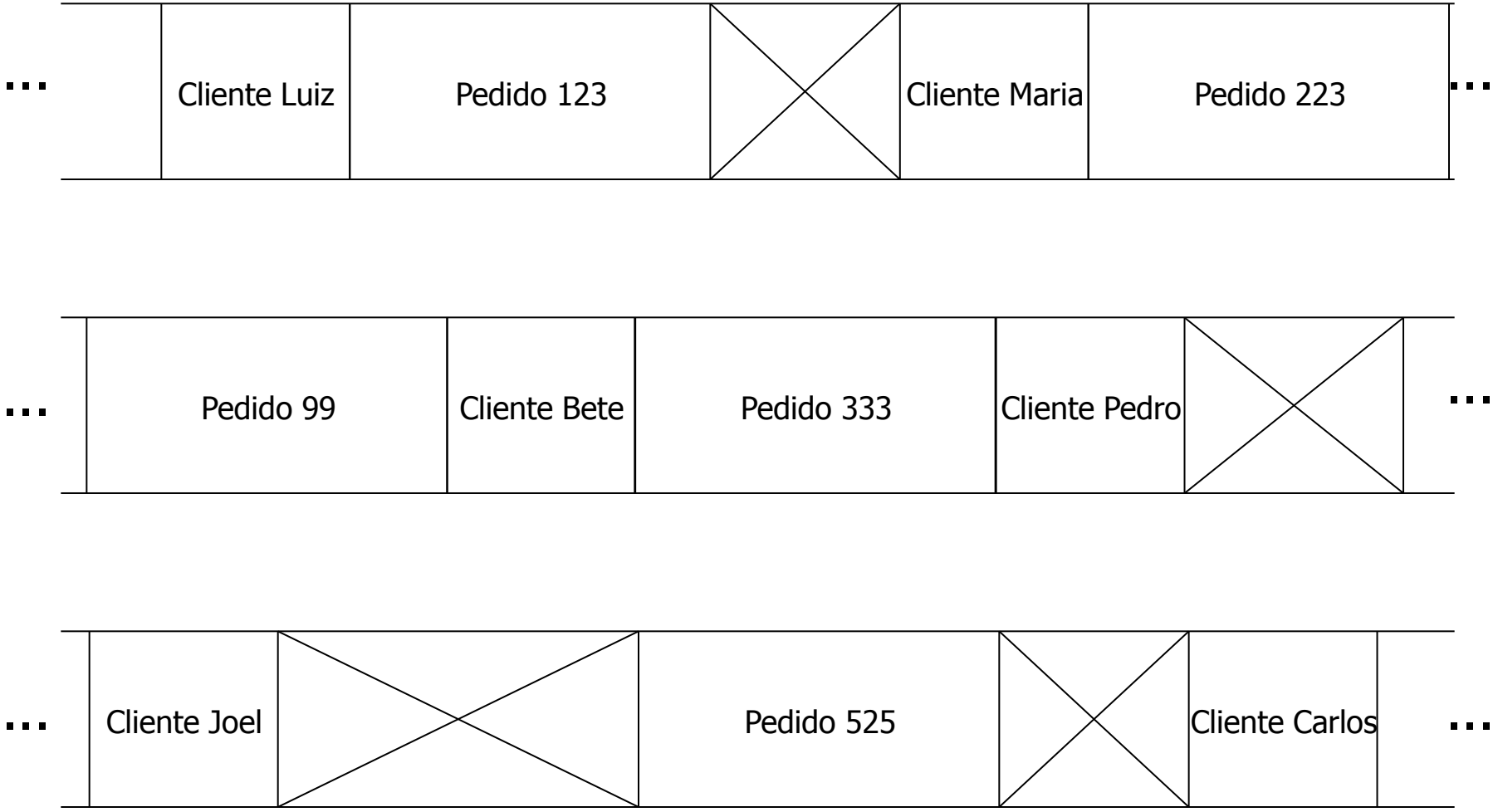
Antes de Prosseguirmos...
A alocação de memória em poucas palavras

- O projetista é o responsável por definir mecanismos para guardar, de forma organizada, os endereços dos objetos na memória principal, pois:
  - Se os endereços forem perdidos, os objetos são inacessíveis;
  - Objeto inacessível não colabora (não há como solicitar a ele a execução de algo) e, portanto,
  - O espaço ocupado por ele é um espaço "morto" (lixo)!

Ver ilustração a diante...

Antes de Prosseguirmos...

A alocação de memória em poucas palavras



PUC-Rio

Antes de Prosseguirmos...

A alocação de memória em poucas palavras

- O projetista normalmente cria objetos para indexar outros objetos;
- Alguns objetos têm a "vocação" natural para gerenciar outros objetos (por exemplo, Clientes gerenciam seus respectivos Pedidos).

Consolidando o Que Foi Dito

- Objetos "colaboram" para a realização das funções de um sistema;
- A colaboração é realizada pela troca de mensagens entre eles;
- As mensagens são passadas por meio de chamadas de métodos (um objeto chama método(s) de outro(s));
- Para o objeto "A" solicitar a execução de um método do objeto "B", é necessário que "A" "conheça" o endereço de "B";
- Além disso, "A" e "B" precisam estar na memória principal;
- Se um objeto não tem seu endereço conhecido por qualquer outro objeto do sistema, ele não pode colaborar e, portanto, pode ser removido da memória.

Consolidando o Que Foi Dito

- Um objeto tem seu ciclo de vida iniciado quando passa a existir no sistema (por uma ação de cadastramento, por exemplo) e tem seu ciclo de vida encerrado quando é expurgado do sistema (remoção do BD, por exemplo);
- Objetos que "sobrevivem" à desativação do sistema, sendo armazenados em algum lugar, são os objetos persistentes;
- Objetos persistentes são, em geral, instâncias de classes conceituais.

Consolidando o Que Foi Dito

- A sequência de trocas de mensagens e quais objetos participam de uma colaboração são definidos pelos projetistas/programadores dos sistemas;
- DSs (um dos diagramas de interação da UML) especificam os objetos participantes e a sequência de trocas de mensagens entre eles.

Consolidando o Que Foi Dito

- DSs são úteis para a descoberta de operações dos objetos. A medida em que desenvolvemos os DSs, vamos "populando" os terceiros compartimentos das classes dos objetos envolvidos;
- DSs podem nos ajudar a gerar código, se forem suficientemente detalhados. Podemos gerar um DS a partir do código (Eng. Reversa), mas o resultado é, em geral, visualmente muito complexo e desorganizado $\rightarrow$ pouca utilidade;
- Poucos projetistas usam DSs para a geração de código com base no modelo.

Consolidando o Que Foi Dito

- Cenários são caminhos únicos em um caso de uso; cada cenário tem seu começo (comum a todos os demais), um meio e um fim;
- Se descrevermos os UCs usando DAs, facilmente identificamos visualmente os cenários;
- Cada cenário é especificado pelas ações executadas ou pelos desvios trilhados no cenário.

Consolidando o Que Foi Dito

- Antes dos "frames" (UML 2.0) construíamos um DS por cenário → dezenas de diagramas por Caso de Uso. Mais uma razão para poucos usarem DS antes dos frames.
- Frames ajudam a reutilizar partes de colaborações em diversos diagramas e modelam paralelismo, condicionalidades, opções, loops, etc.
- Frames tornam os DS visualmente mais simples;
- Veremos frames mais tarde;
- Ao elaborarmos um DS, somos induzidos a pensar nos detalhes (implementação).

Exercícios

1. Identificando Passos de uma Colaboração.

   Exercício 5.1 da apostila de exercícios.

2. Obtendo o Lucro Líquido da ZYX Ltda.

   Exercício 5.2 da apostila de exercícios.