

SC-301 – Programação Orientada a Objetos II

Introdução à

UML

Parte 03

Prof. Dr. Fábio Fagundes Silveira

fsilveira@unifesp.br

<http://fabiosilveira.net>

UNIFESP – Universidade Federal de São Paulo

março/2010



Créditos

- Grade parte dos slides seguintes foram preparados/elaborados pelos Professores:
 - Prof. Dr. Jaelson Freire Brelaz de Castro; e
 - Prof. Dr. Ulrich Schiel



Bibliografia

- *The Unified Modelling Language - User Guide* (G. Booch, J. Rumbaugh, I. Jacobson) - Addison Wesley
- *The Unified Modelling Language - Reference Manual* (J. Rumbaugh, I. Jacobson, G. Booch) - Addison Wesley
- *The Unified Software Development Process* (I. Jacobson, G. Booch, R. Rumbaugh) - Addison Wesley
- *UML Distilled* (Martin Fowler) - Addison Wesley

UML

UML



Conteúdo

- Introdução à UML
- Conceitos Gerais
- Apresentação dos diagramas da UML
- Novos diagramas da UML 2.0



Diagramas de Atividades



Diagramas de Atividades

- Os Diagramas de Atividades mostram o **fluxo entre atividades** (ações não-atômicas)
- São um caso especial dos Diagramas de Estados, com a maioria das transições resultantes do **término das atividades**
- São semelhantes aos antigos **fluxogramas**
- São muito usados para modelar **atividades concorrentes**



Concorrência, Forks e Joins

- **Barras de sincronização** são usadas para especificar **forks** e **joins**
- Um **fork** representa um único fluxo de controle em vários fluxos de controle concorrentes
- Um **join** representa a sincronização de dois ou mais fluxos de controle concorrentes

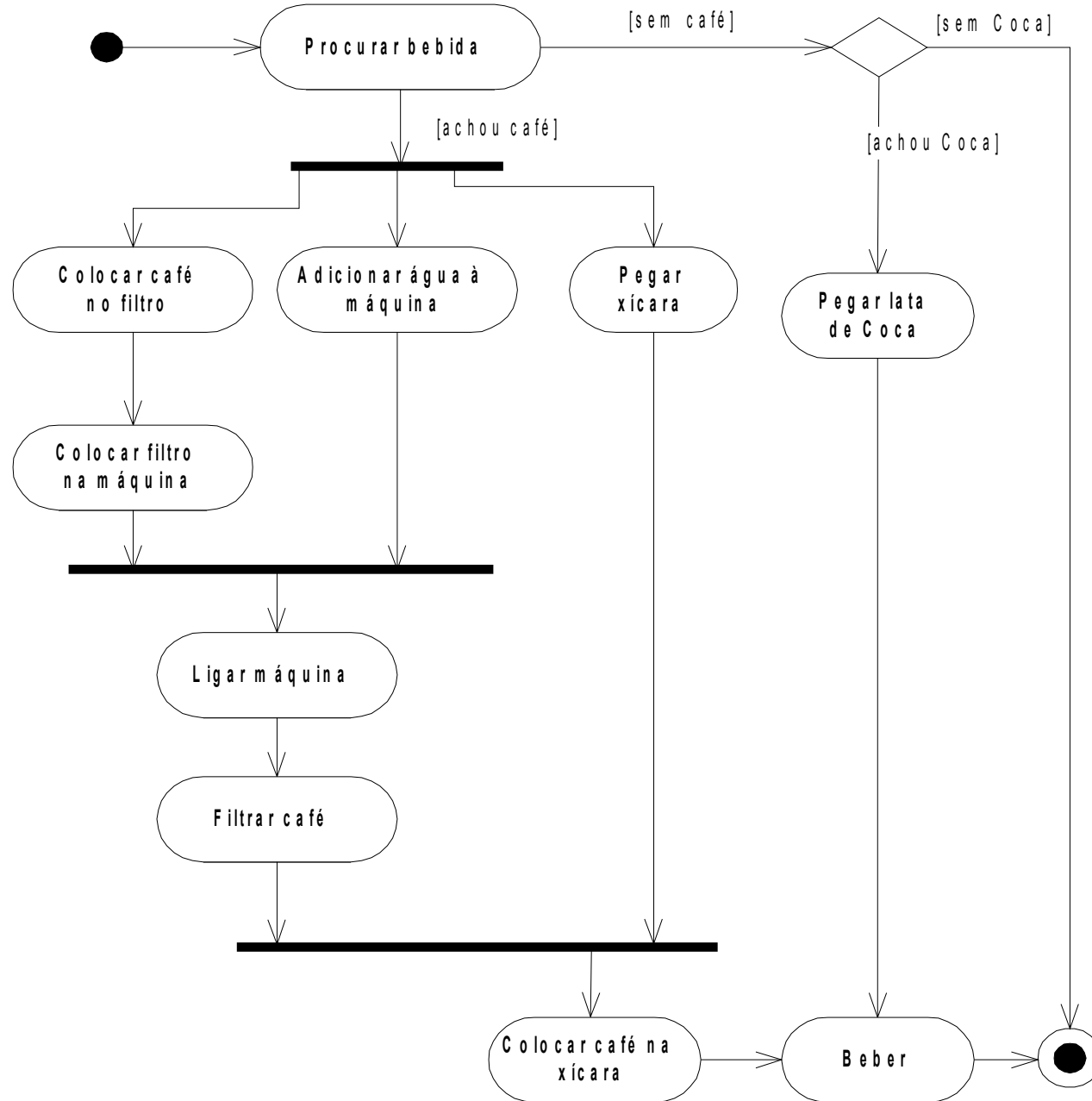


Concorrência, Forks e Joins

- Atividades depois de um **fork** podem ser realizadas em qualquer ordem, ou ao mesmo tempo
- Para que as atividades depois de um **join** possam ser realizadas, todas as atividades **antes** do join devem ser concluídas



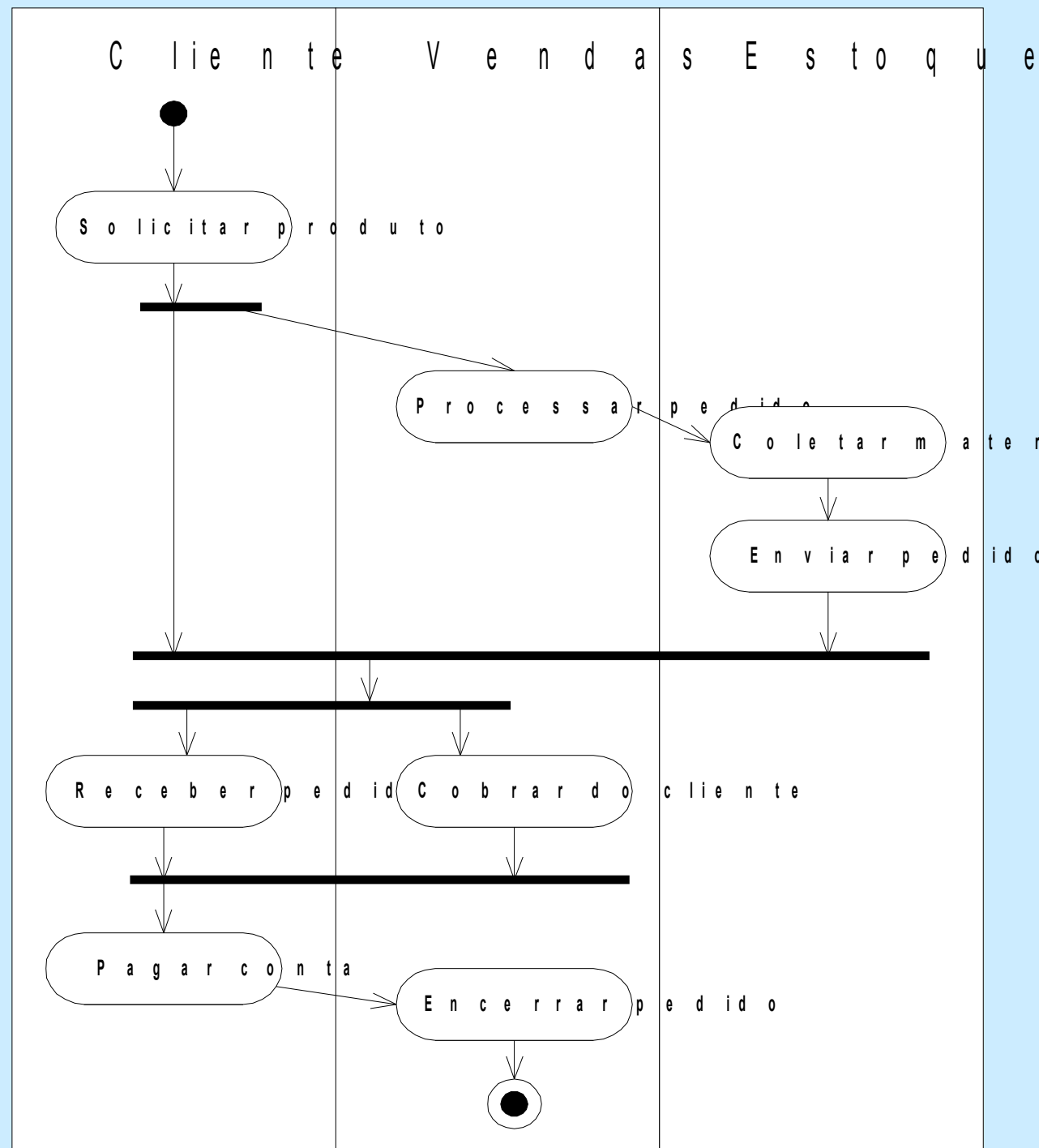
Pessoa





Swimlanes (raias)

- **Swimlanes** (raias) são usadas para definir quais são as classes (ou conjuntos de classes) responsáveis pela realização de cada atividade
- Swimlanes são especialmente úteis para a modelagem de processos empresariais
- Em muitos casos, os swimlanes implicam **concorrência**, ou pelo menos **independência**, das atividades





Diagramas de Componentes



Componentes

- Os componentes existem no mundo material, de bits
- São um elemento importante na modelagem dos **aspectos físicos** de um sistema
- Um componente é uma **parte física** e **substituível** de um sistema, que **realiza** um conjunto de **interfaces**

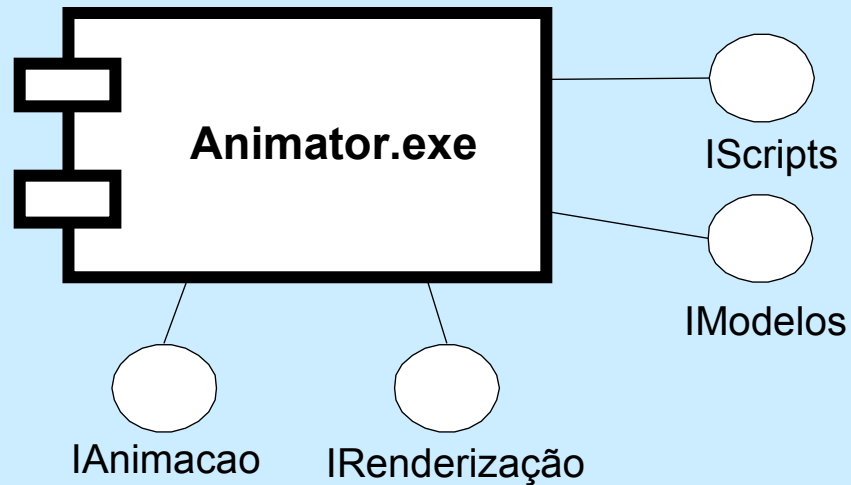


Componentes

- Componentes são coisas que podem ser **executadas** em nós (processadores, dispositivos, etc.)
- Exemplos de componentes são executáveis, bibliotecas, tabelas, arquivos e documentos
- Um componente, tipicamente, é uma versão física de elementos lógicos, como classes e interfaces

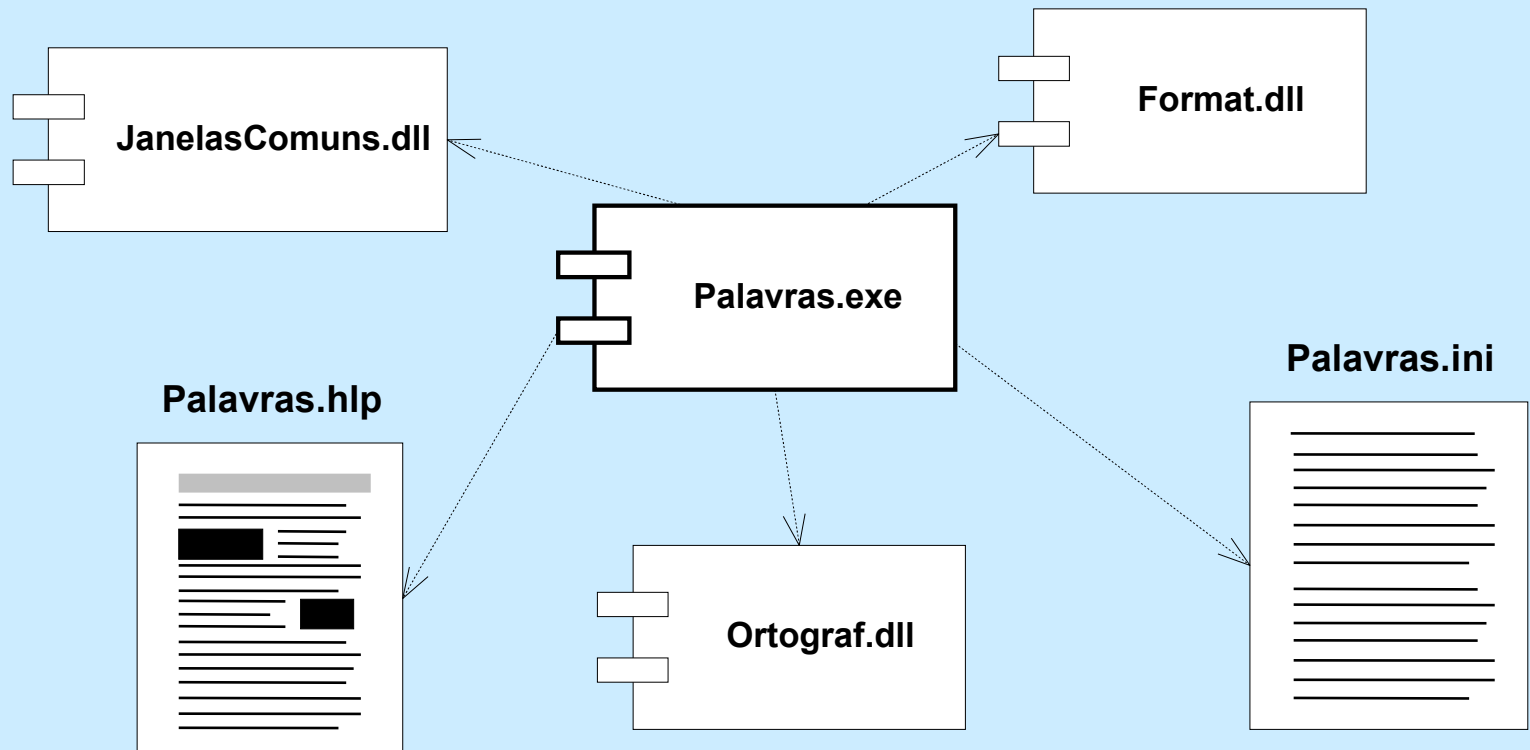


Componentes- Notação





Componentes e Dependências





Diagramas de Componentes

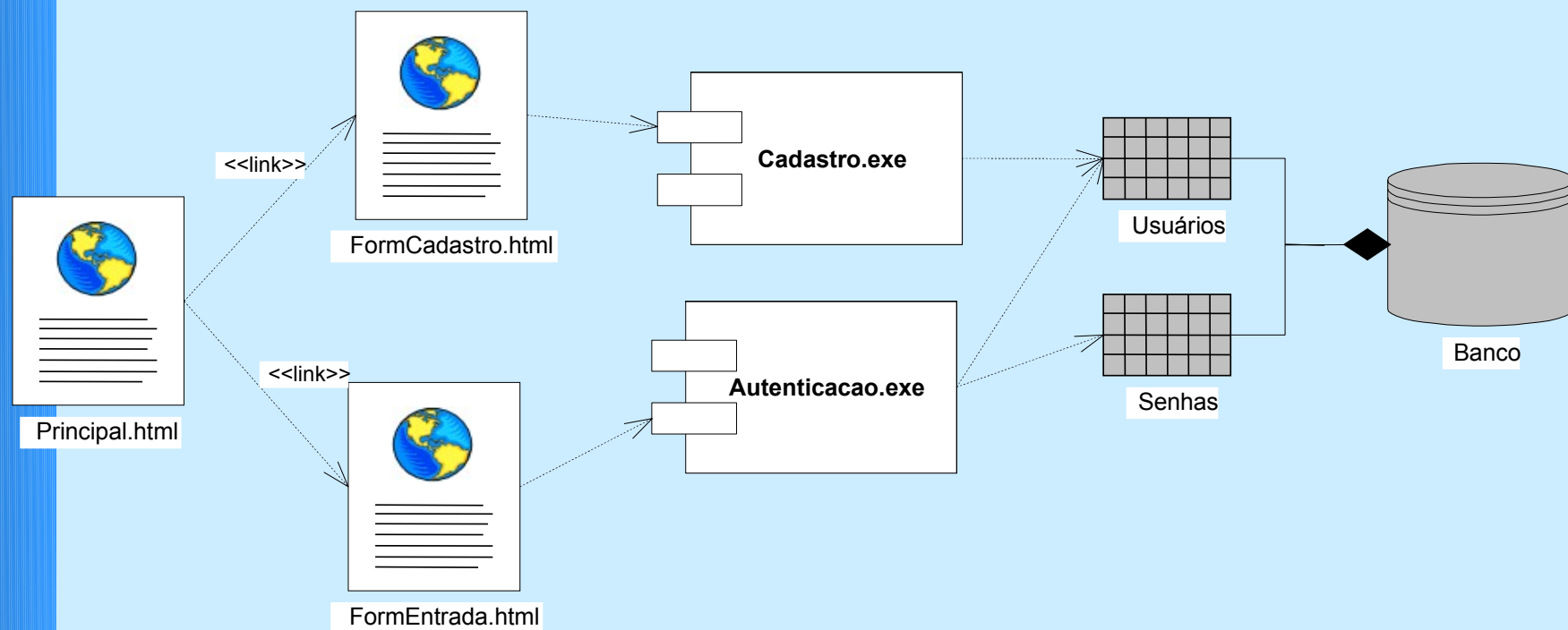
- Diagramas de Componentes são usados para modelar os **aspectos físicos** de um sistema
- Nos diagramas de componentes, são mostrados **componentes** e os **relacionamentos** entre eles



Diagramas de Componentes

- Diagramas de componentes podem ser usados para modelar
 - os aspectos físicos de um banco de dados
 - o código fonte de um aplicativo
 - uma API
- A única restrição é que o que está sendo modelado deve ser físico (formado por bits) e não conceitual (ou lógico)

Exemplo





Diagramas de Implantação



Diagramas de Implantação

- Os diagramas de implantação são usados para modelar o ambiente em que o sistema será executado
- São compostos por **nós** e **associações**(relacionamentos de comunicação)
- Um nó pode ser, por exemplo, um computador, uma rede, um disco rígido, um sensor, etc.



Diagramas de Implantação

- Os diagramas de implantação podem ser usados para descrever a **topologia** do ambiente no qual o software será executado
- Os diagramas de implantação geralmente só fazem sentido para sistemas que rodam em várias máquinas ou dispositivos
- Para sistemas que rodam em uma única máquina e se comunicam apenas com dispositivos comuns, como o teclado, monitor, os diagramas de implantação não são necessários



Diagramas de Implantação e Estereótipos

- Como nós representamos **elementos físicos** (tangíveis) de um sistema, eles são os elementos mais estereotipados da UML
- O recurso de **estereótipos** permite estender a linguagem UML com novos símbolos e nova semântica
- **Símbolos** como PCs, Workstations, Servidores e Dispositivos são muito usados em diagramas de implantação, para tornar os diagramas mais claros



Nós e Associações

- Um nó é um elemento físico que existe em **tempo de execução** e representa algum **recurso computacional**. Um nó, geralmente, possui **memória** e, muitas vezes, capacidade de **processamento**
- Uma **associação** entre dois nós representa uma **conexão física** entre os nós, como um conexão Ethernet, uma linha serial ou um link de satélite



Diagramas de Implantação

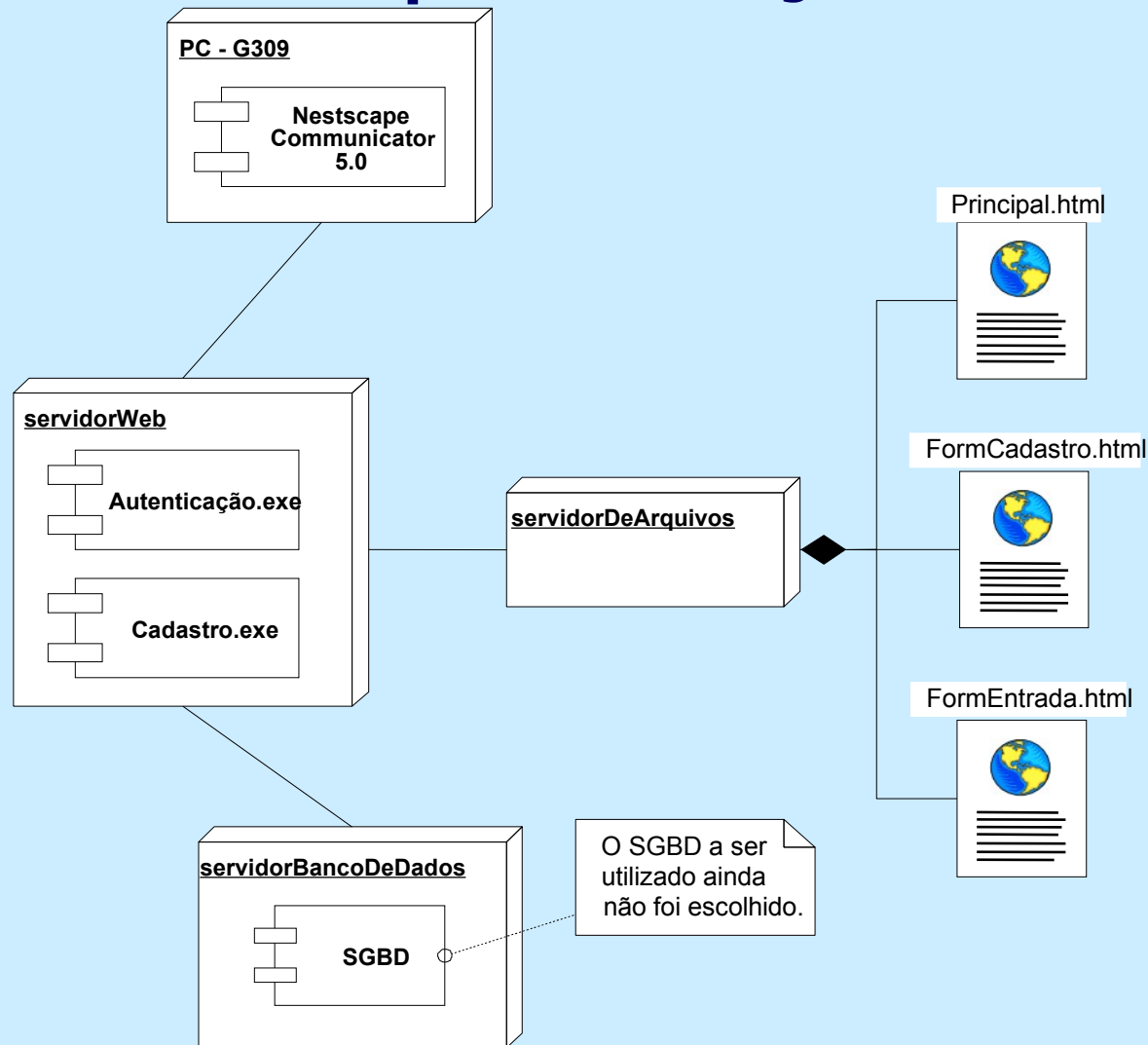




Diagrama de Estrutura Composta

Descreve a estrutura interna de um classificador, classe ou componente, detalhando suas partes internas e como essas partes se comunicam e colaboram

Descreve a colaboração de um conjunto de instâncias na realização de um tarefa

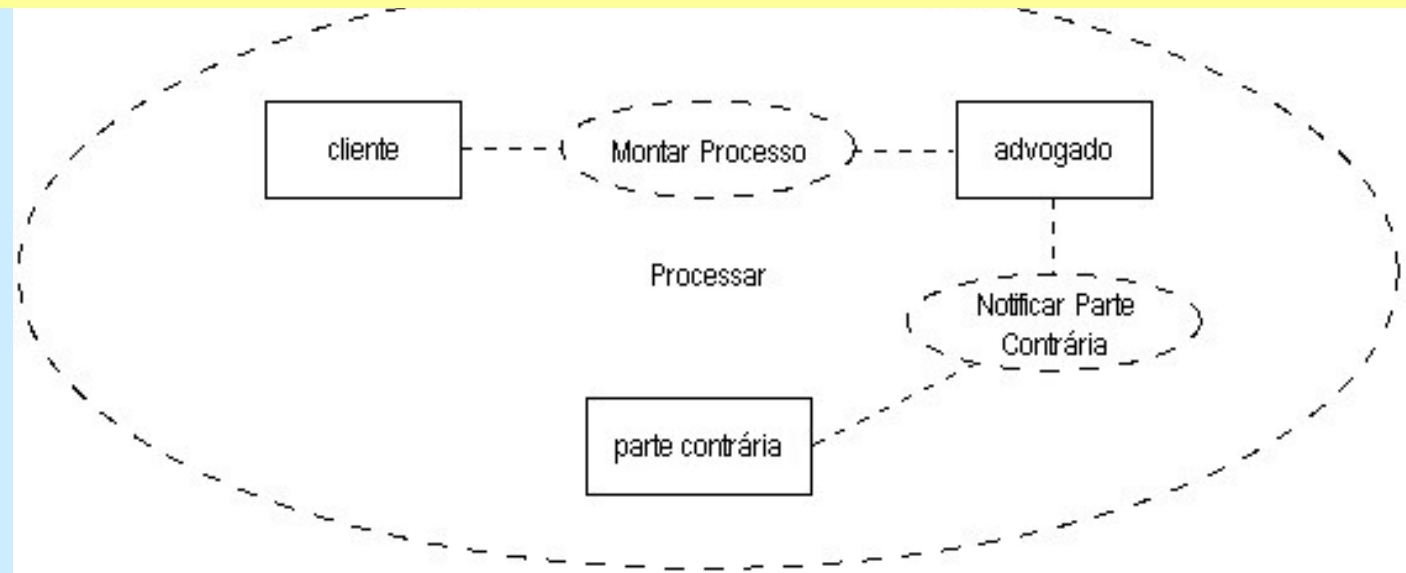




Diagrama de Interação Geral

Variação do Diagrama de Atividades

Fornece uma visão geral dentro de um sistema ou processo de negócio

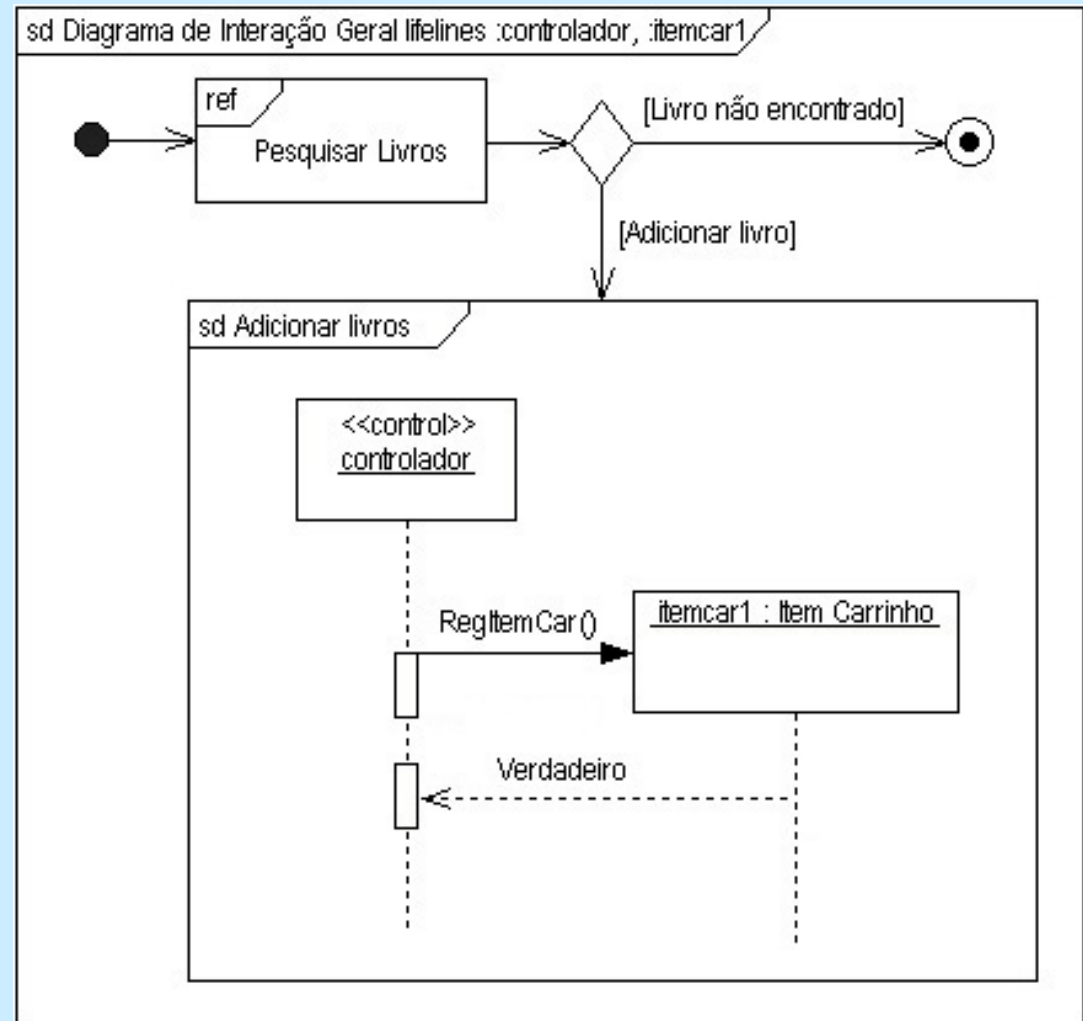
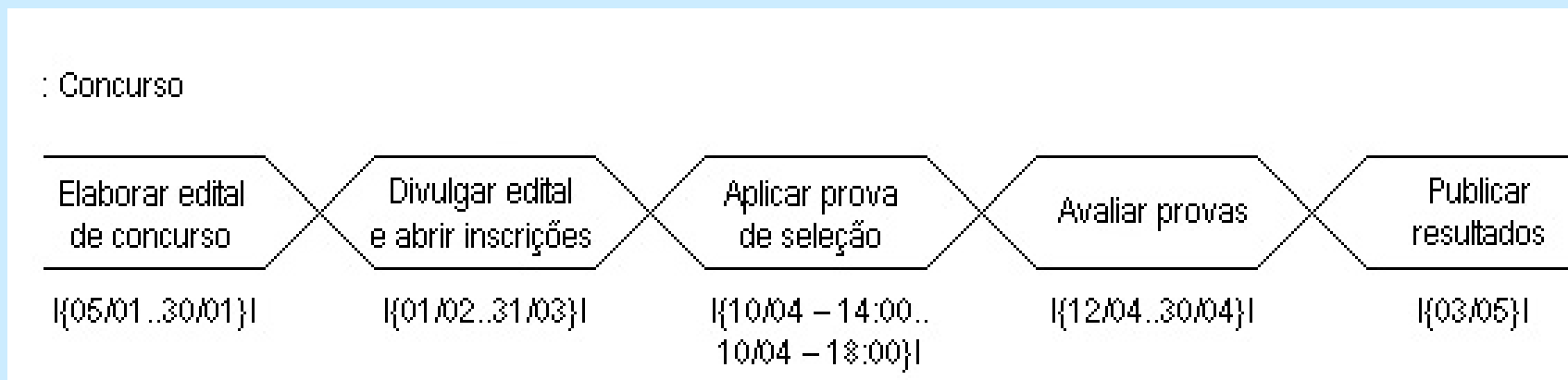




Diagrama de Tempo

Descreve a mudança no estado ou condição de uma instância de uma classe ou seu papel durante um certo tempo

Demonstra a mudança no estado de um objeto no tempo em resposta a eventos externos





Bibliografia

- The Unified Modelling Language User Guide (Grady Booch)
- The Unified Modelling Language Reference Manual (James Rumbaugh)
- The Unified Software Development Process (Ivar Jacobson)
- UML Distilled (Martin Fowler)