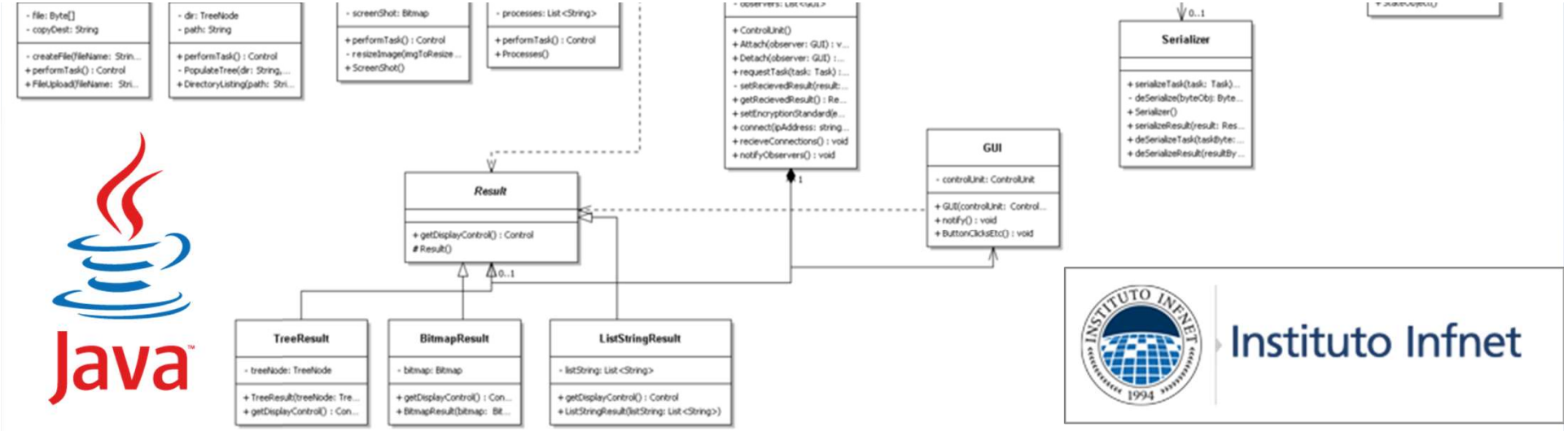
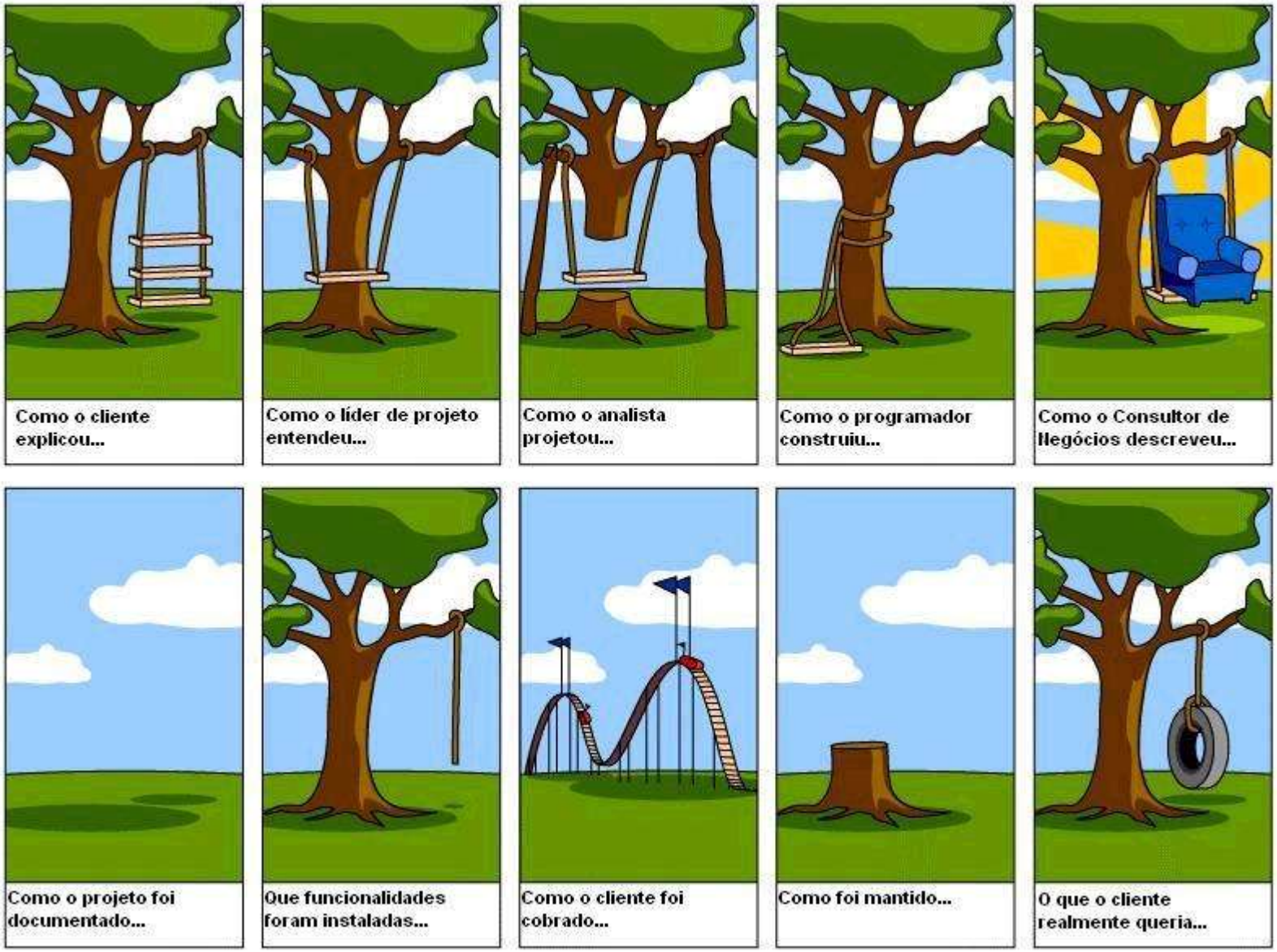


UML e Desenvolvimento Java

Engenharia de Requisitos e Visão Geral de UML

Etapa 1

Gustavo de Miranda Gonçalves
gustavo.miranda@prof.infnet.edu.br


Como o cliente explicou...
Como o líder de projeto entendeu...
Como o analista projetou...
Como o programador construiu...
Como o Consultor de Negócios descreveu...
Como o projeto foi documentado...
Que funcionalidades foram instaladas...
Como o cliente foi cobrado...
Como foi mantido...
O que o cliente realmente queria...


QUEM SOMOS?
USUÁRIOS
O QUE QUEREMOS?
NÃO SABEMOS!
QUANDO QUEREMOS?
AGORA!

Engenharia de Requisitos

- Concepção
- Levantamento
- Elaboração
- Negociação
- Especificação
- Validação
- Gestão

Engenharia de Requisitos

- **Concepção**
  Estabelecer um entendimento do problema e das pessoas envolvidas.

- **Levantamento**
  Coleta de Requisitos

- **Elaboração**
  As informações coletadas durante a concepção e o levantamento são expandidas e refinadas. Construção de um modelo técnico das características do software tendo como produto final um modelo de análise que define o domínio do problema.

- **Negociação**
  Clientes, usuários e outros interessados são solicitados a ordenar os requisitos e discutir os conflitos de prioridade. Riscos dos requisitos devem ser identificados e analisados. O impacto de cada requisito no custo do projeto e no prazo de entrega também deve ser validado.

- **Especificação**
  Na especificação, um documento escrito, combinando descrições em linguagem natural e modelos gráficos pode ser a melhor abordagem. Para sistemas menores, casos de uso podem ser suficientes.

- **Validação**
  Na validação, um exame sobre a especificação é realizado para garantir que todos os requisitos de software tenham sido declarados de modo não ambíguo. Além disso, omissões e erros devem ser detectados e corrigidos.

- **Gestão**

  A gestão de requisitos é um conjunto de atividades que ajuda a equipe de projeto a identificar, controlar e a rastrear requisitos e suas modificações em qualquer momento do projeto.

  As tabelas de rastreamento relacionam os requisitos identificados a um ou mais aspectos do sistema, ou de seu ambiente. Alguns tipos de tabela de rastreamento são de:

  - **Características**
  - **Fontes**
  - **Dependência**
  - **Subsistemas**
  - **Interface**

UML

Unified Modeling Language (UML)

Linguagem de Modelagem Unificada

É uma linguagem de modelagem para a elaboração da estrutura de projetos de software.

Tem como **objetivo** especificar, documentar e estruturar para sub-visualização e maior visualização lógica do desenvolvimento completo de um sistema de informação.

Tipos de Diagramas do UML

UML 2 possui 14 tipos de diagramas divididos em duas categorias:

- **Diagramas Estruturais**
  Enfatizam os elementos que precisam estar presentes no sistema modelado.

- **Diagramas Comportamentais**
  Enfatizam o que precisa acontecer no sistema modelado.

Tipos de Diagramas do UML

UML 2 possui 14 tipos de diagramas divididos em duas categorias:

- **Diagramas Estruturais**
  Enfatizam os elementos que precisam estar presentes no sistema modelado.

  - Diagrama de Classes
  - Diagrama de Objetos
  - Diagrama de Componentes
  - Diagrama de Instalação ou de Implantação
  - Diagrama de Pacotes
  - Diagrama de Estrutura Composta
  - Diagrama de Perfil

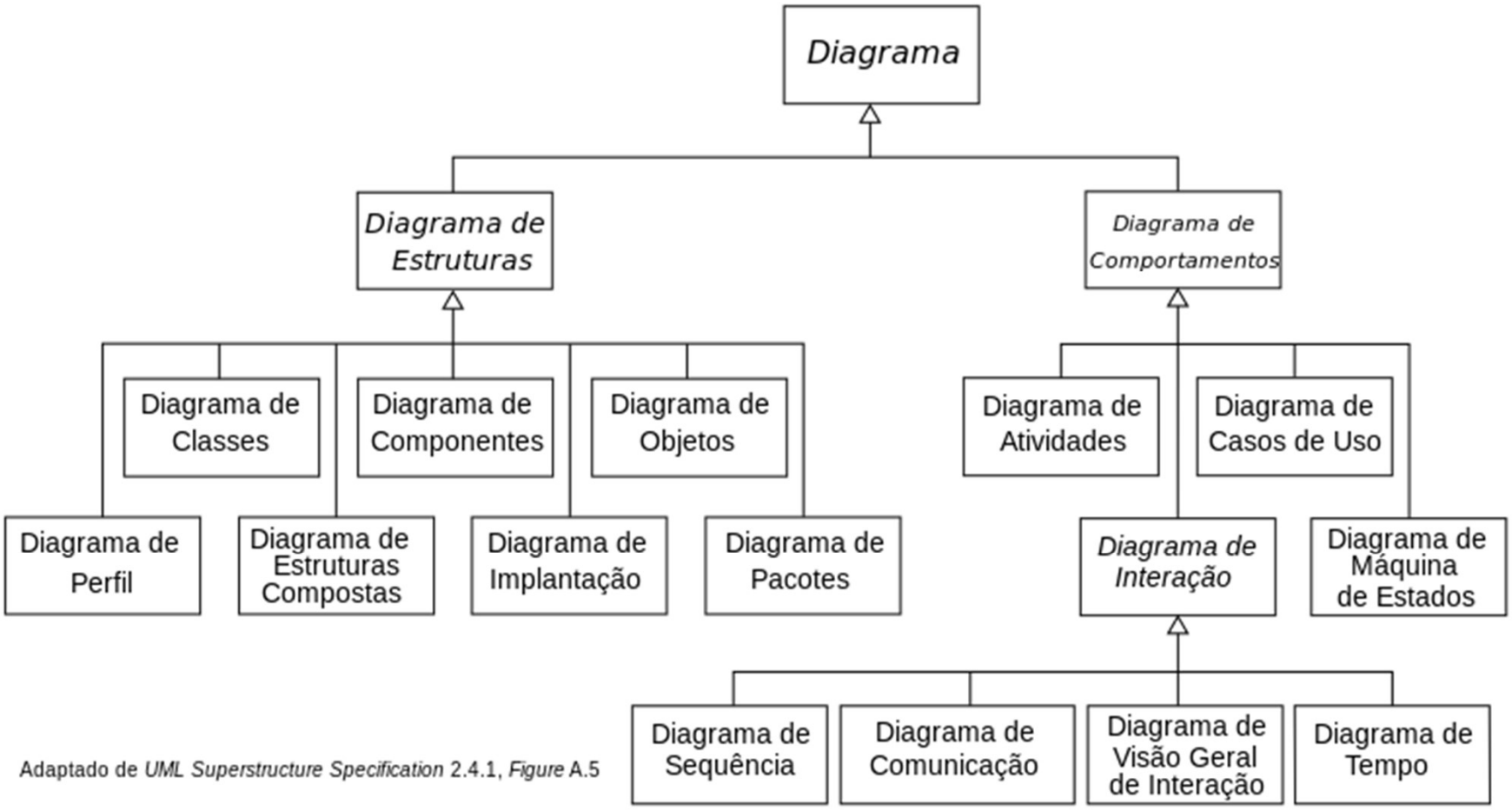
Tipos de Diagramas do UML

UML 2 possui 14 tipos de diagramas divididos em duas categorias:

- **Diagramas Comportamentais**
  Enfatizam o que precisa acontecer no sistema modelado.

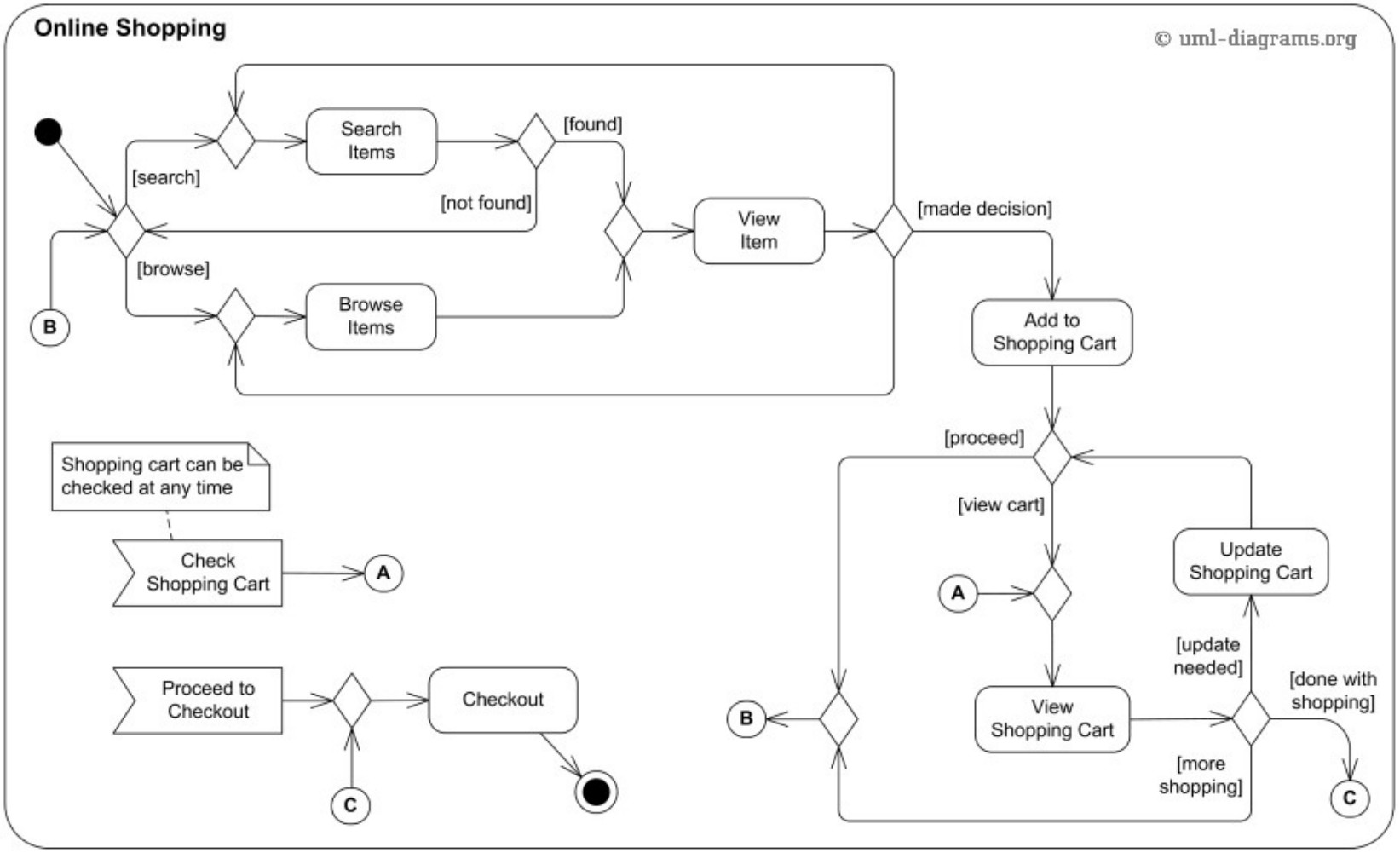
  - Diagrama de Caso de Uso
  - Diagrama de Transição de Estados (ou de Estados)
  - Diagrama de Atividade
  - Diagrama de Sequência
  - Diagrama Visão Geral de Interação (ou de Interação)
  - Diagrama de Colaboração (ou Comunicação)
  - Diagrama de Tempo (ou Temporal)

Tipos de Diagramas do UML



Adaptado de *UML Superstructure Specification* 2.4.1, *Figure* A.5

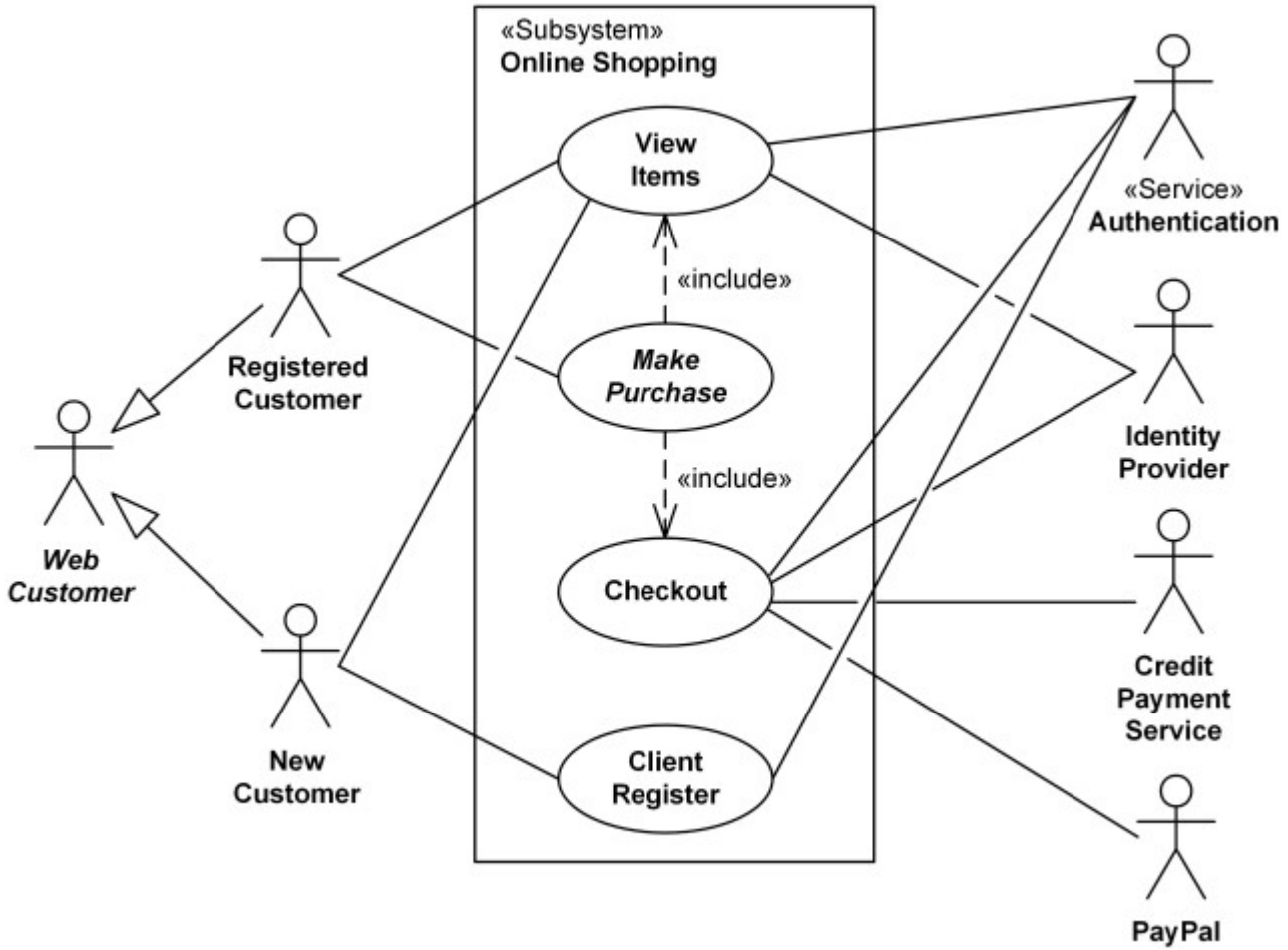
UML: ***Diagrama de Atividades***



http://www.uml-diagrams.org/online-shopping-uml-activity-diagram-example.html?context=activity-examples

UML: ***Diagrama de Caso de Uso***

Exemplo Online Store - *Top Level*



http://www.uml-diagrams.org/examples/online-shopping-use-case-diagram-example.html?context=uc-examples

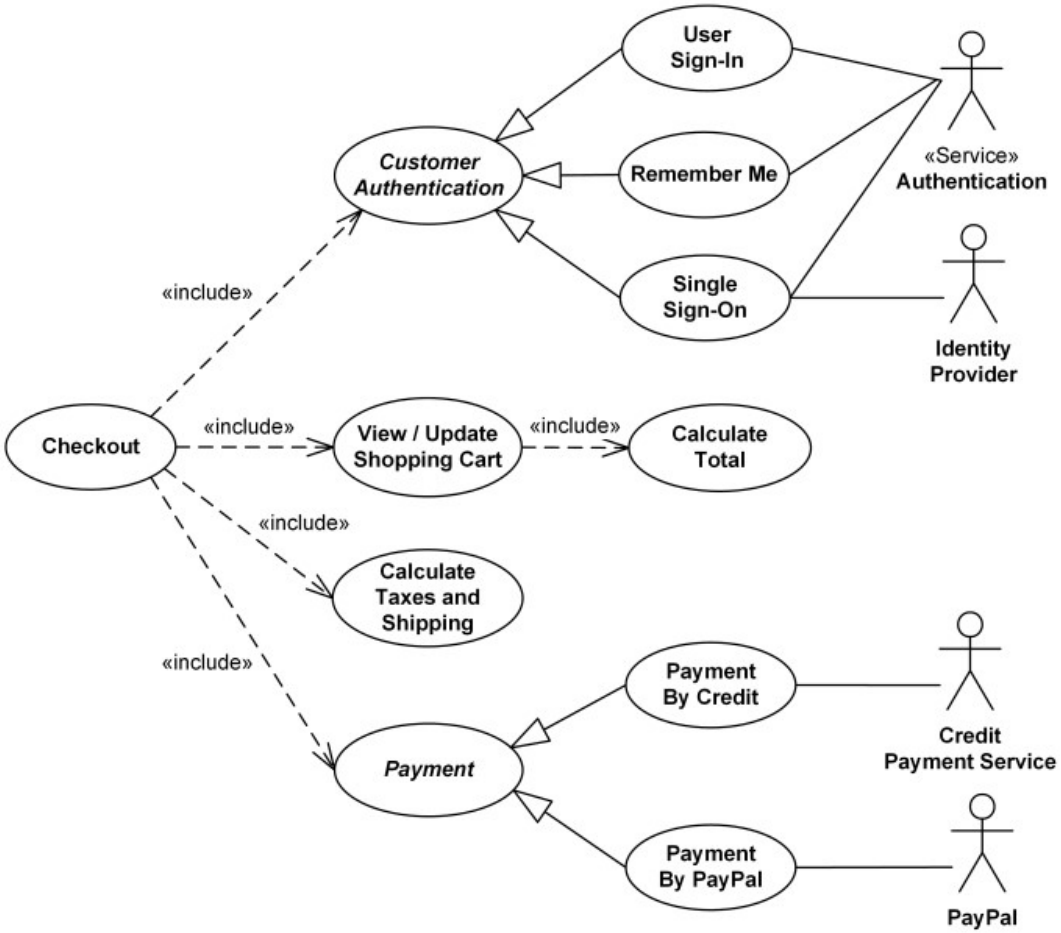
UML: ***Diagrama de Caso de Uso***

Exemplo Online Store - *View Items*

http://www.uml-diagrams.org/examples/online-shopping-use-case-diagram-example.html?context=uc-examples

UML: ***Diagrama de Caso de Uso***

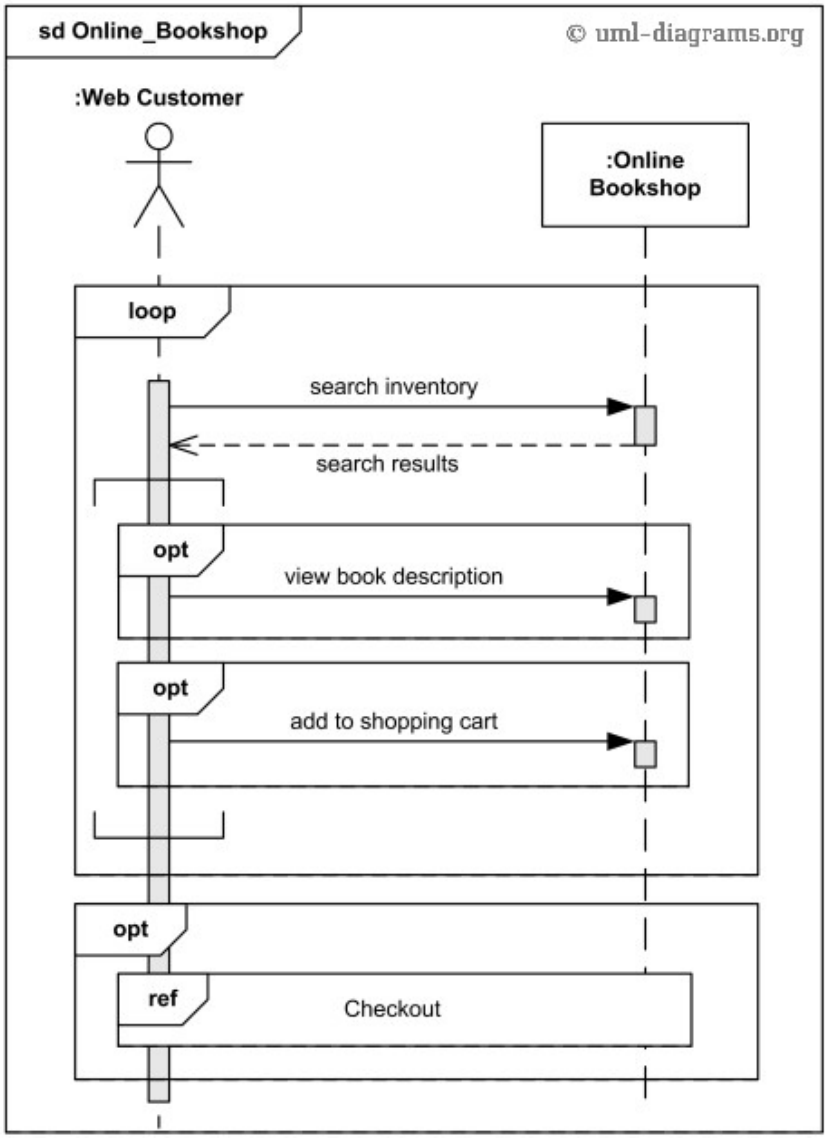
Exemplo Online Store – *Checkout, authentication and payment use cases*



http://www.uml-diagrams.org/examples/online-shopping-use-case-diagram-example.html?context=uc-examples

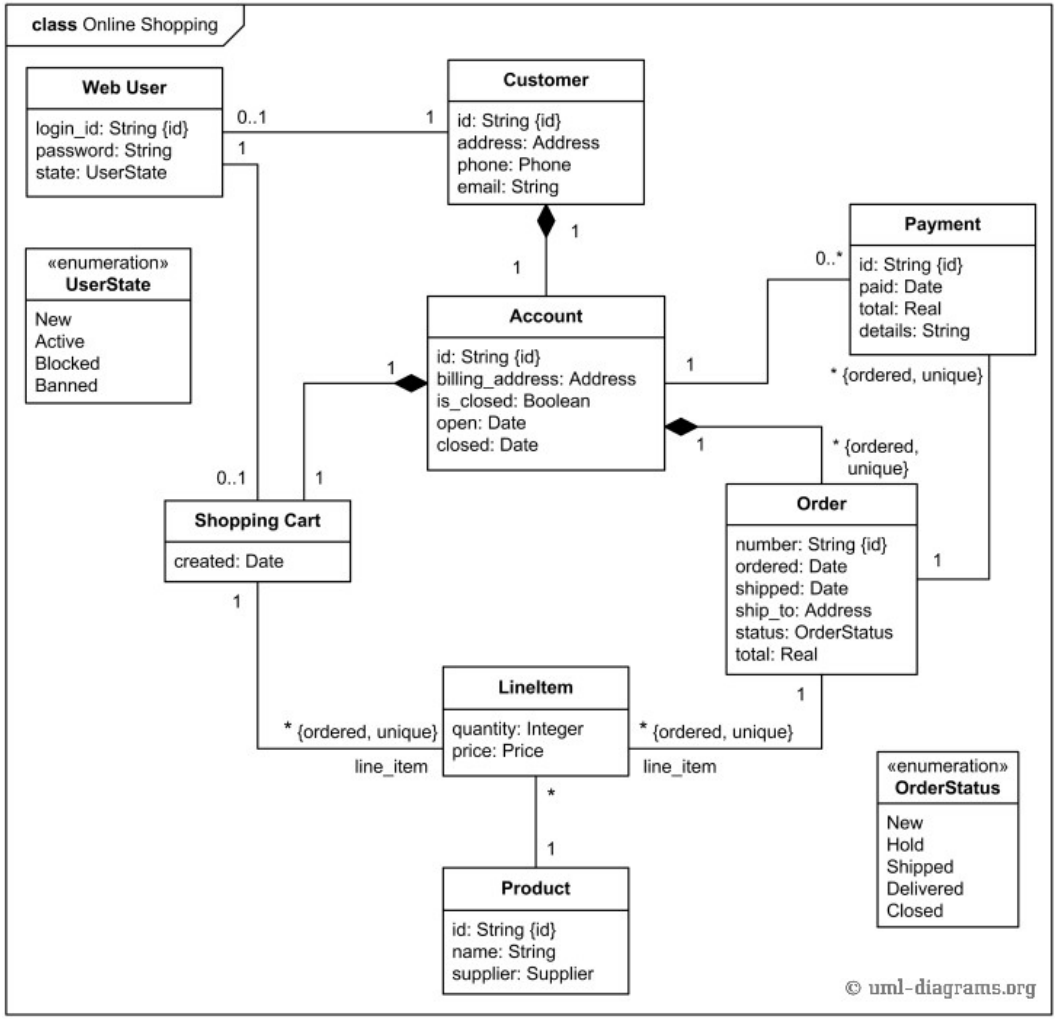
UML: ***Diagrama de Estados***

Exemplo Online Bookshop



http://www.uml-diagrams.org/examples/online-shopping-use-case-diagram-example.html?context=uc-examples

UML: ***Diagrama de Classes***



http://www.uml-diagrams.org/examples/online-shopping-domain-uml-diagram-example.html?context=cls-examples


392 PR
BMW
M3 Coupé 1999
558 PR
SUBARU
Impreza WRX STI
621 PR
DODGE
Challenger SRT8
GEN
509 PR
MAZDA
RX-7 FD
607 PR
PORSCHE
911 (993) Carrera
675 PR
NISSAN
Skyline GT-R BNR32
Choose a car to modify

Design Patterns

Padrões de Projeto (*Design Patterns*)

Padrões de projeto são soluções gerais para problemas que ocorrem com frequência dentro de um determinado contexto no projeto de software.

Não são projetos finalizados que podem ser transformados diretamente em código fonte. **São descrições ou modelos (*templates*) que se propõem a resolver diferentes problemas.**

São práticas formalizadas que o programador pode utilizar para resolver esses problemas comuns.

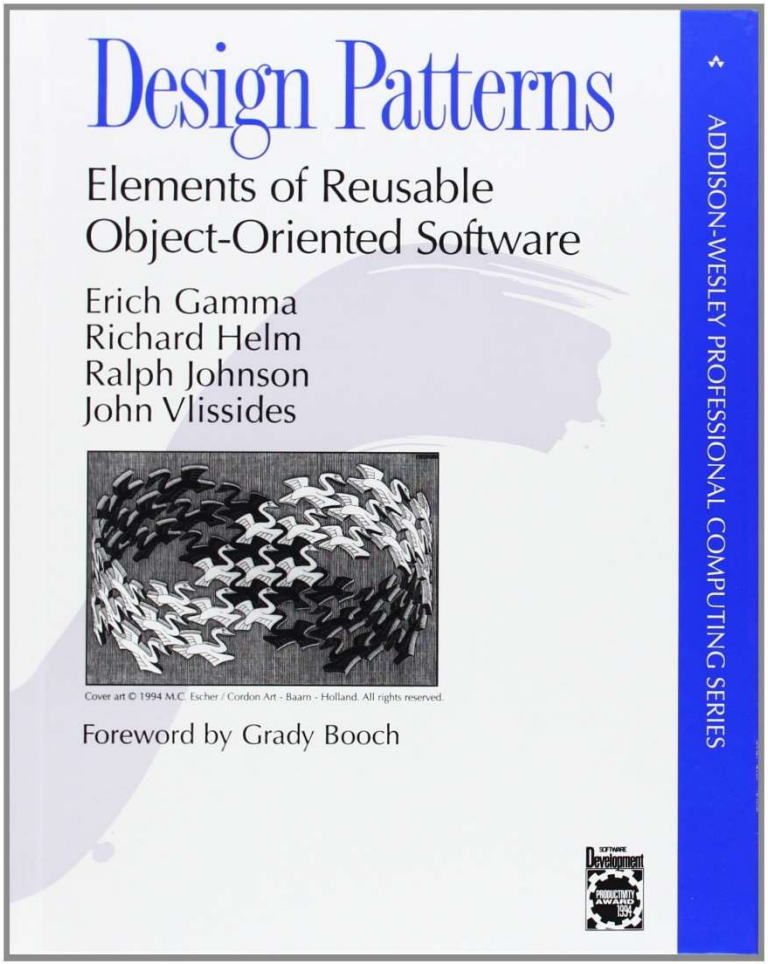
Padrões de Projeto (*Design Patterns*)

- **Padrões GOF**
  G*ang of Four*

- **Padrões GRASP**
  General Responsibility Assignment Software Patterns (or Principles)

Design Patterns: **Tipos de Padrões GOF (Gang of Four)**

- Padrões de Criação
- Padrões Estruturais
- Padrões Comportamentais



Design Patterns: **Tipos de Padrões GOF (Gang of Four)**

Criação
Abstract Factory
Builder
Factory Method
Prototype
Singleton

Estrutural
Adapter
Bridge
Composite
Decorator
Facade
Flyweight
Proxy

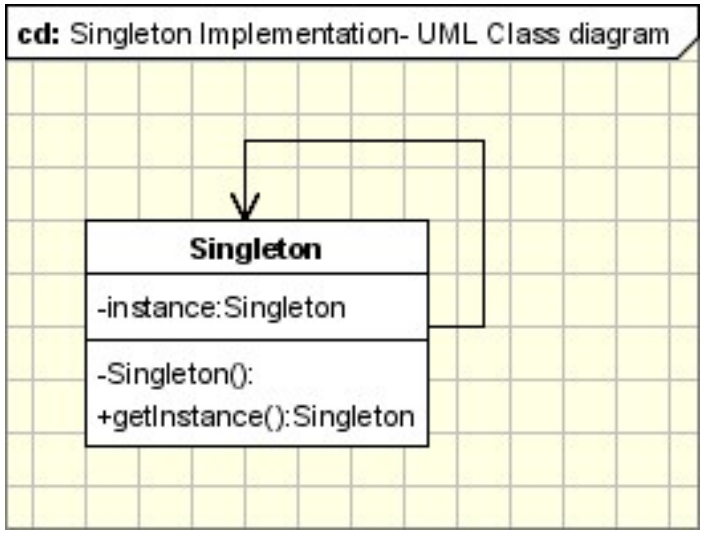
Comportamental	
Chain of Responsibility	State
Command	Strategy
Interpreter	Template Method
Iterator	Visitor
Mediator	
Memento	
Observer	

Design Patterns: ***Singleton***

Problema
Uma classe precisa ter uma única instância.

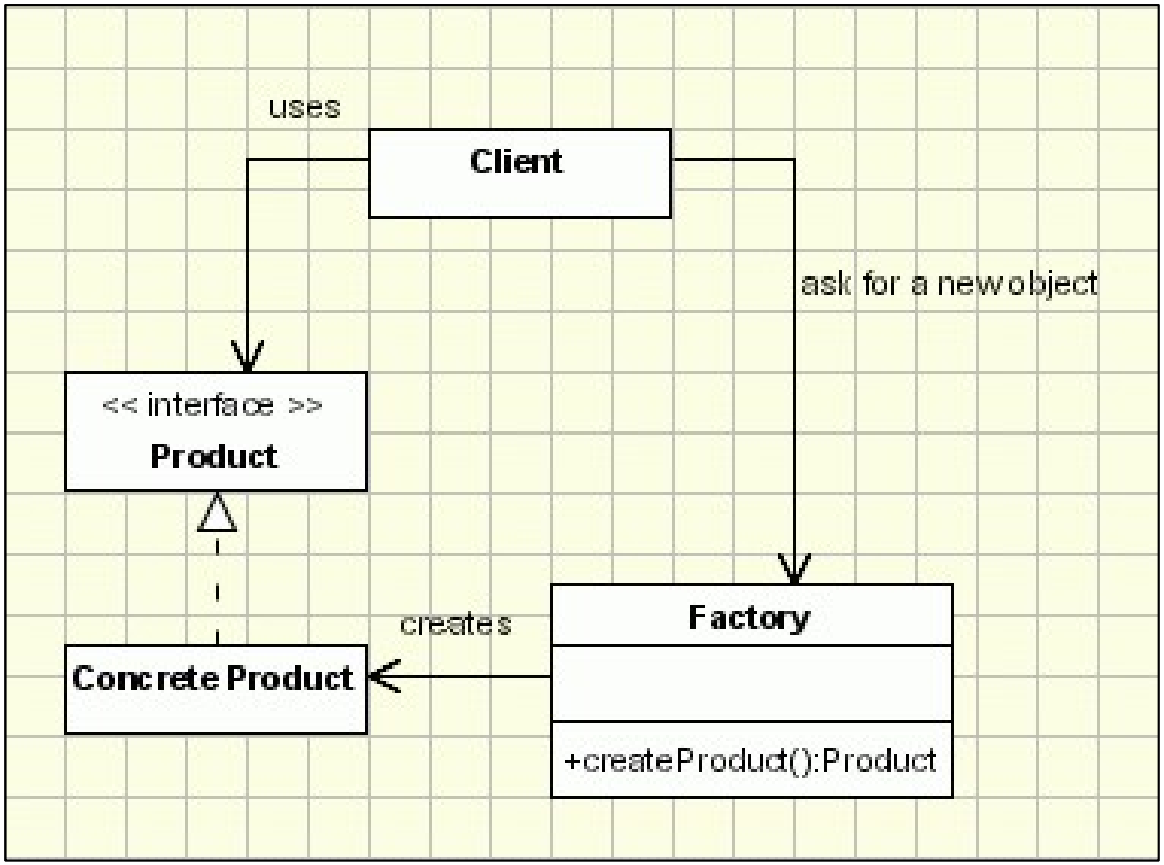
Solução
Garante que uma classe terá apenas uma instância.

Consequência
Fácil acesso a gerência de recursos compartilhados, como variáveis globais.



http://www.oodesign.com/singleton-pattern.html

Design Patterns: ***Factory Method***



http://www.oodesign.com/factory-pattern.html

Design Patterns: **Tipos de Padrões GRASP**
General Responsibility Assignment Software Patterns (or Principles)

Assim como os padrões de projeto do GOF, os padrões GRASP são utilizados para resolução de problemas comuns e bastante típicos de desenvolvimento de software orientado a objeto. Portanto, tais técnicas apenas documentam e normatizam as práticas já consolidadas, testadas e conhecidas no mercado.

Design Patterns: **Tipos de Padrões GRASP**
General Responsibility Assignment Software Patterns (or Principles)

GRASP	
Controller	Polymorphism
Creator	Protected Variations
Indirection	Pure Fabrication
Information Expert	
High Cohesion	
Loose Coupling	

Importância dos Design Patterns

Design Patterns

- *Visam facilitar a reutilização de soluções de desenho – isto é, soluções na fase de projeto do software.*

- *Estabelecem um vocabulário comum de desenho, facilitando comunicação, documentação e aprendizado dos sistemas de software.*