USP - Universidade de São Paulo

IME - Instituto de Matemática e Estatística

MAC0332

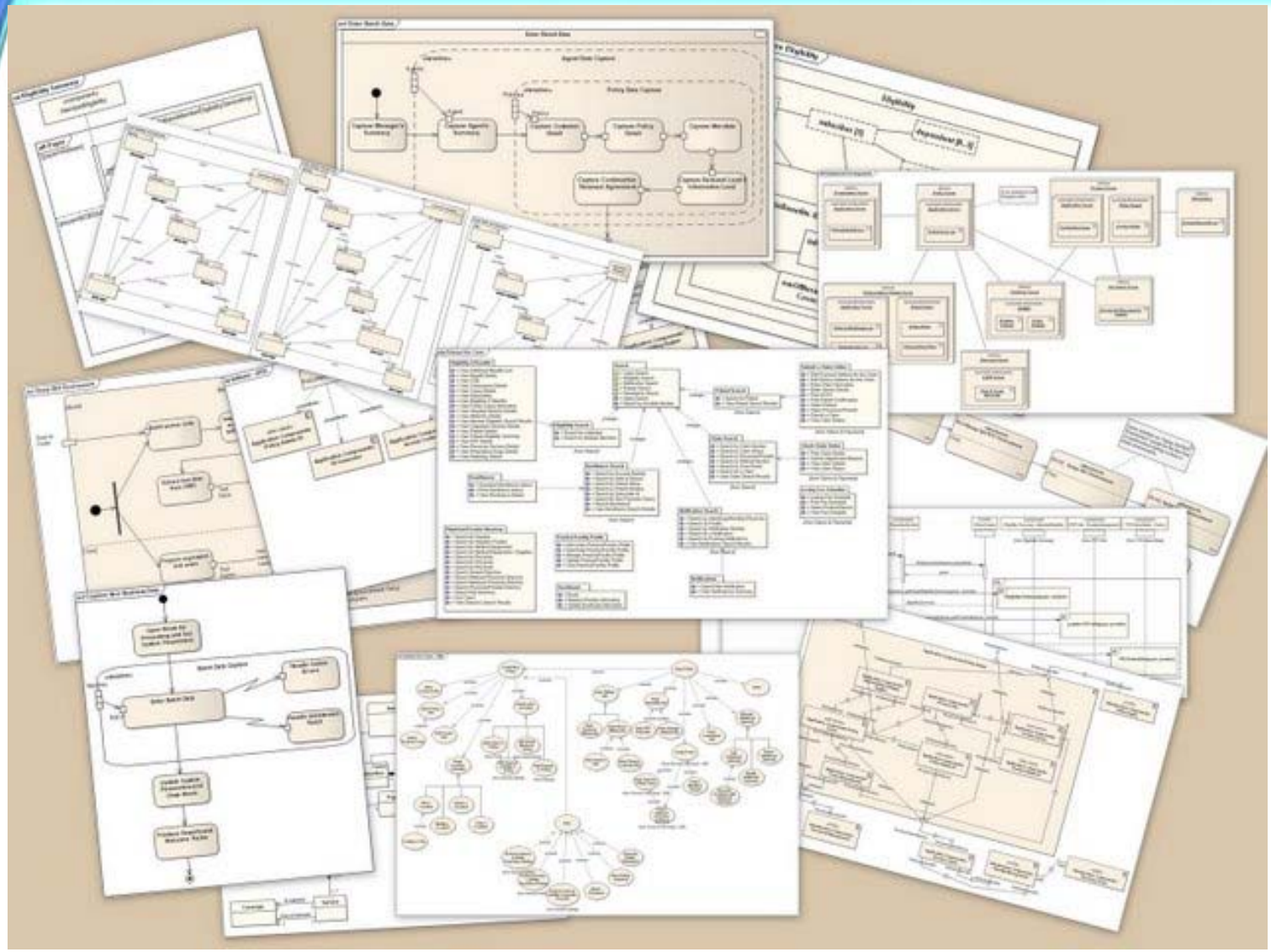
Engenharia de Software

Análise Orientada a Objetos

Marco Aurélio Gerosa

gerosa@ime.usp.br

UML

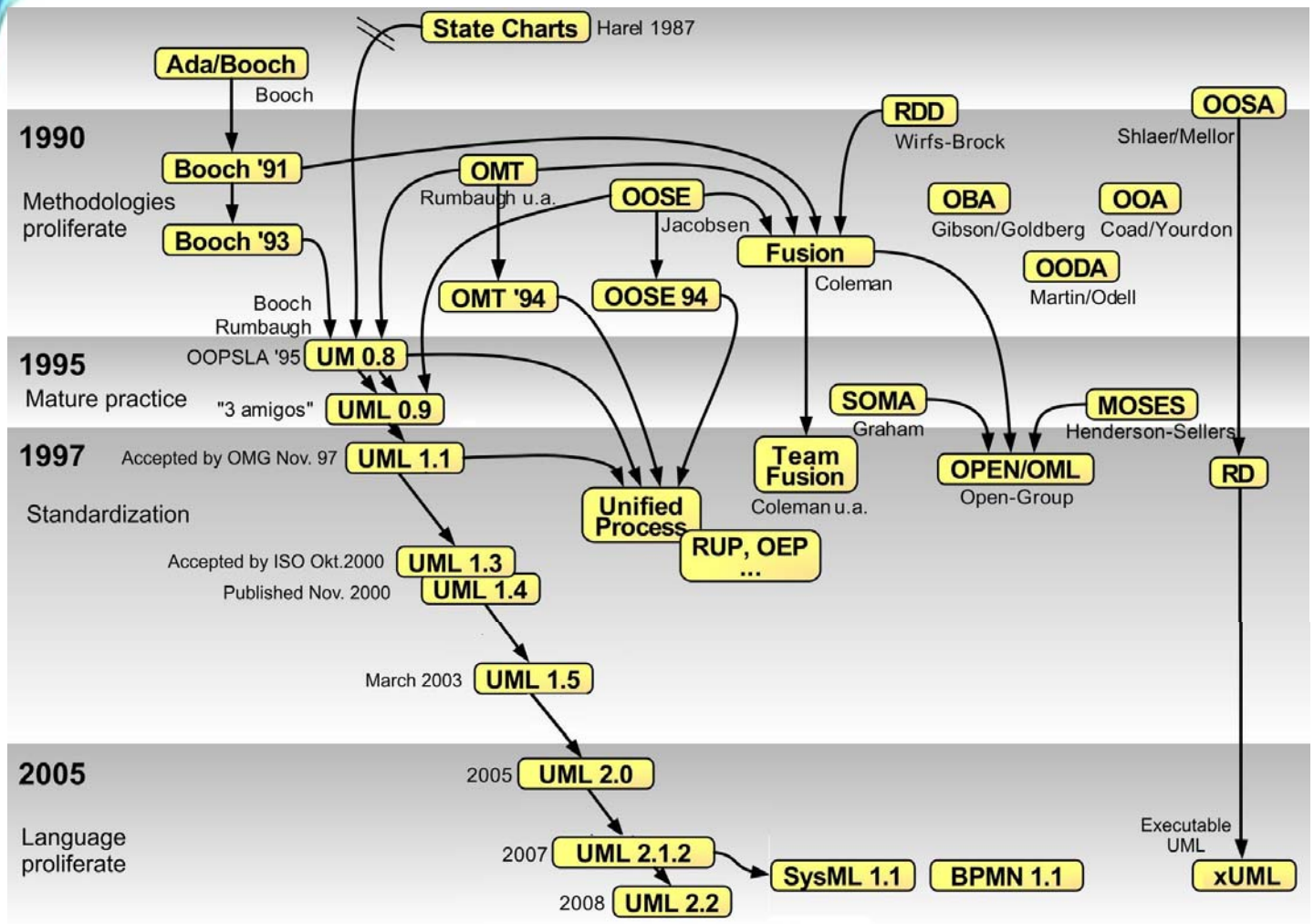


http://en.wikipedia.org/wiki/Unified_Modeling_Language#mediaviewer/File:UML_Diagrams.jpg

UML

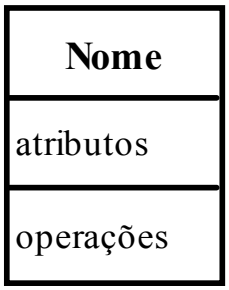
- 1994 – Grady Booch e Jim Rumbaugh unificam os métodos Booch e OMT (Object Modeling Technique)
- 1995 – Versão 0.8 UML (Unified Modeling Language)
- 1996 – Ivar Jacobson, criador do método Objectory, junta-se ao grupo.
- Os três fundam a Rational Software e lançam a versão 0.9 da linguagem
- Em 1997 lançaram a linguagem como proposta de padronização à OMG (Object Management Group), que homologou-a como padrão.
- Grupo RTF (Revision Task Force) para revisão da linguagem
- 1998 – versões 1.2 e 1.3
- 2001 – versão 1.4
- 2005 – versão 2.0
- (...)
- 2010 – versão 2.3
- 2011 – versão 2.4.1
- 2012 – versão 2.5 (in process)

História

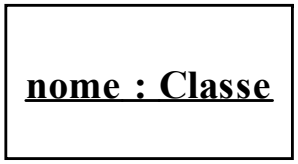


http://en.wikipedia.org/wiki/File:OO-historie.jpg

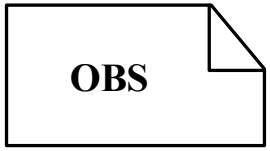
Elementos da UML



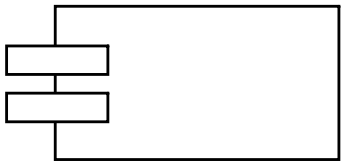
Classe (*class*)



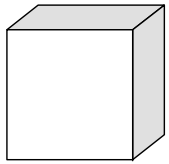
objeto (*object*)



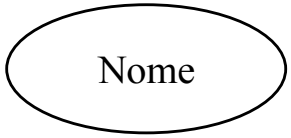
Nota (*note*)



componente (*component*)



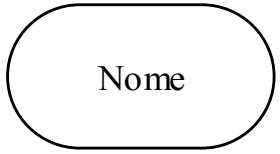
Nó (*node*)



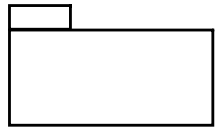
Caso de uso (*use case*)

ator (*actor*)

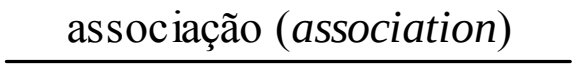
estado (*state*)

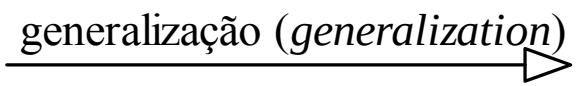


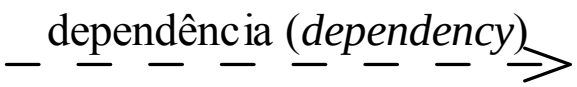
atividade (*activity*)

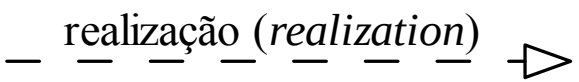


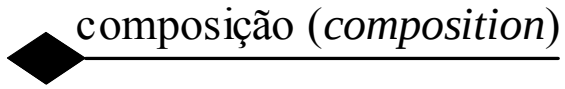
pacote (*package*)

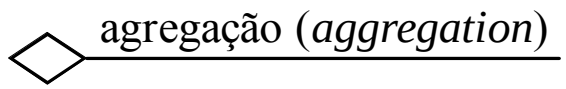




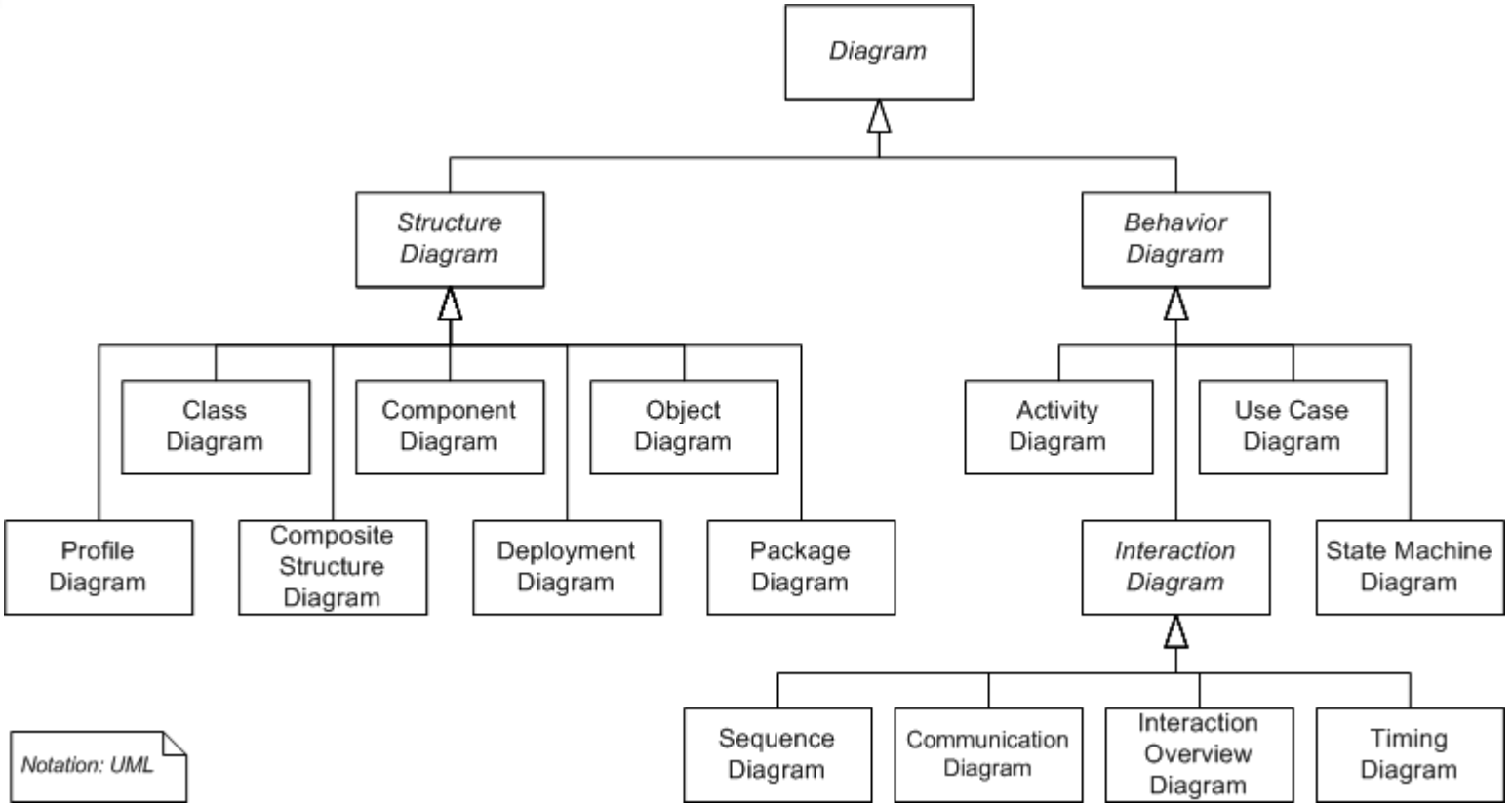








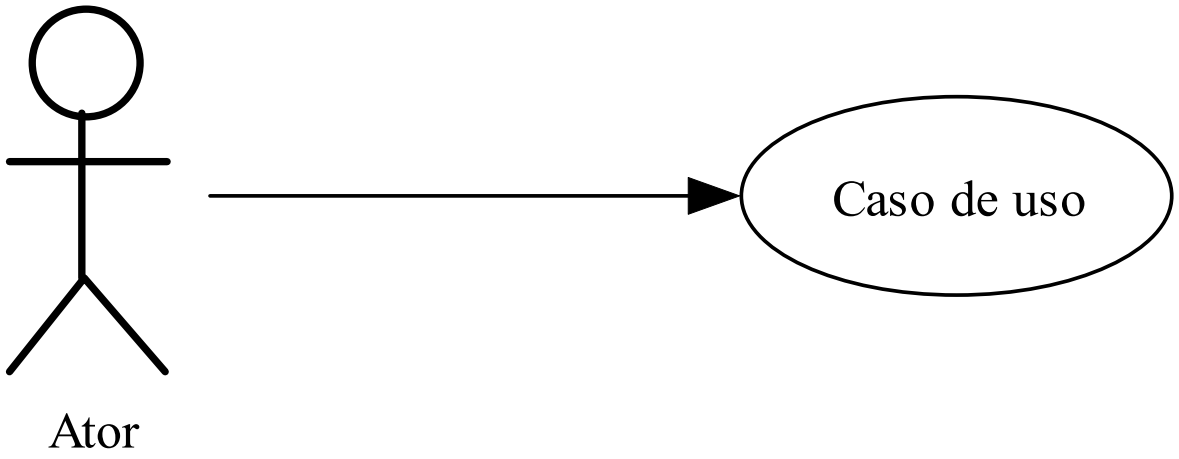
Diagramas UML



http://en.wikipedia.org/wiki/File:Uml_diagram2.png

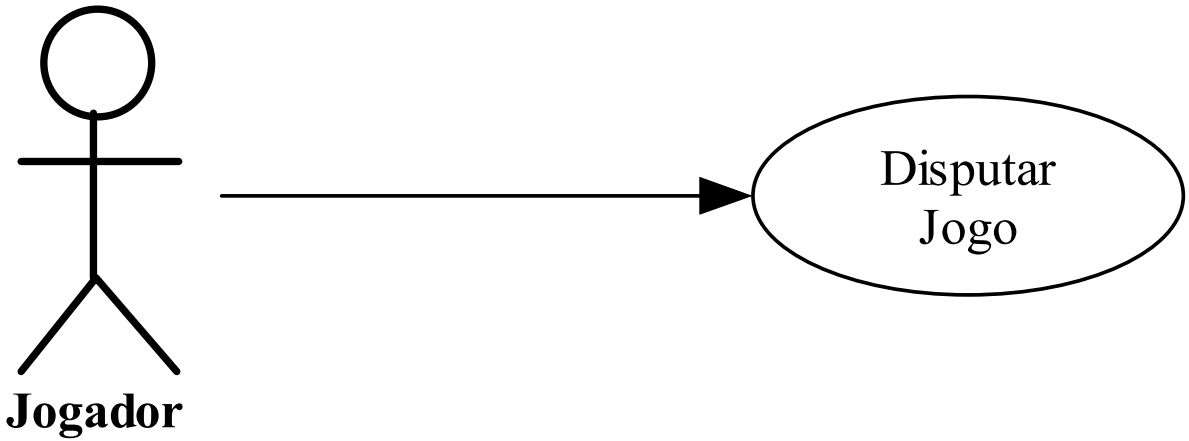
Diagrama de caso de uso

- (já visto em aula anterior):



Exemplo

- *Descrição do problema*
- O sistema deverá simular um jogo de dados entre um jogador e o computador. Cada um joga dois dados e quem somar mais pontos ganha a partida. Ganha o jogo quem ganhar três partidas.

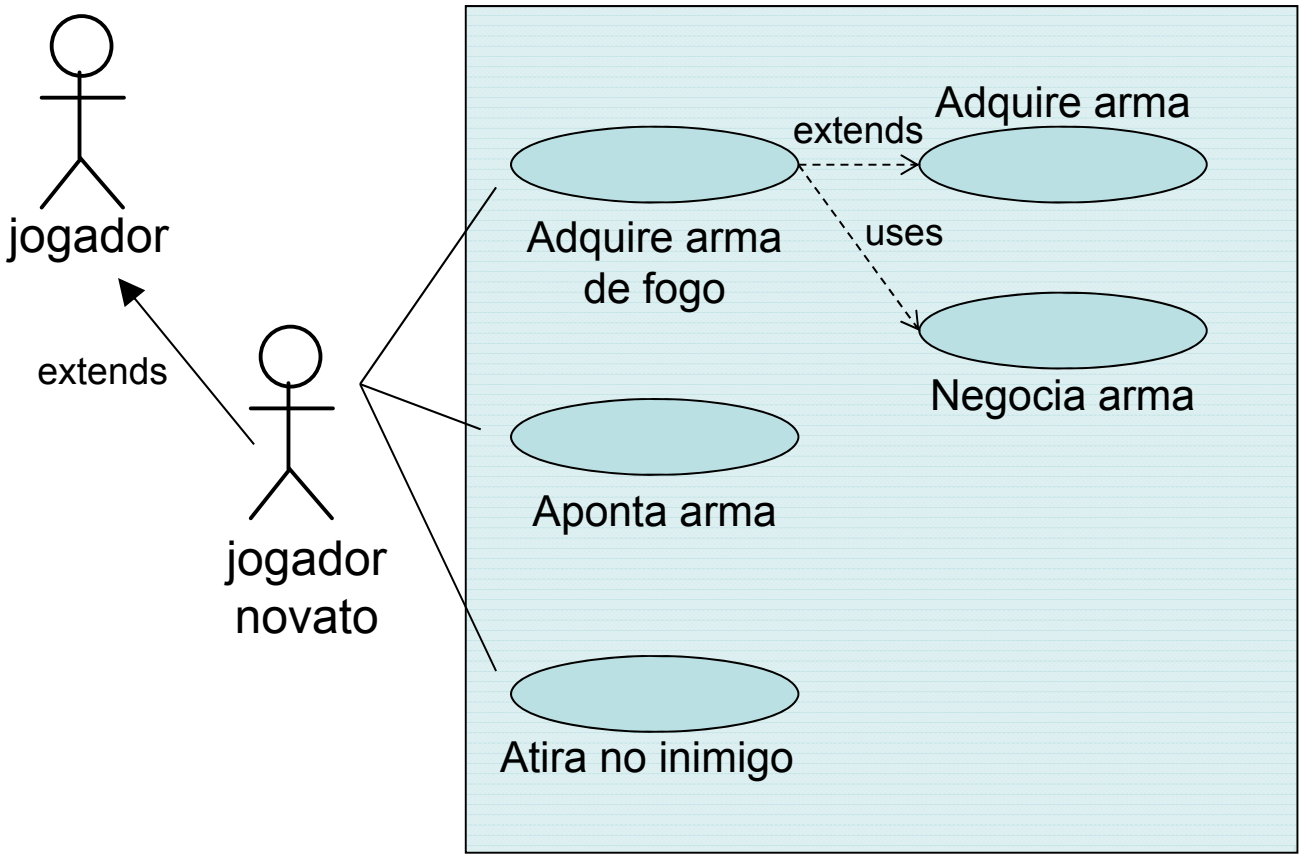


Descrição do caso de uso

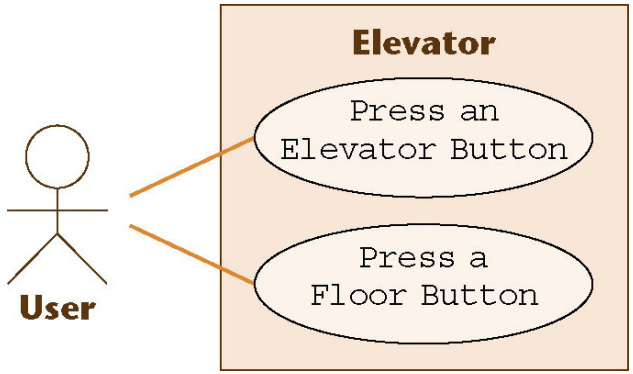
- **Caso de Uso:** Disputar Jogo
- **Ator**: Jogador
- **Descrição**: O jogador disputa o jogo com o computador.
- **Fluxo principal**
- 1. O jogador informa seu nome.
- 2. O sistema registra o nome e dá início ao jogo.
- 3. O jogador informa que quer lançar os dados.
- 4. O sistema sorteia o valor de face dos dados do jogador e exibe o resultado.
- 5. O sistema sorteia o valor de face dos dados para o computador e exibe o resultado.
- 6. O sistema verifica quem obteve a maior soma de pontos e exibe o ganhador da partida.
- 7. Se nem o jogador e nem o computador obtiveram três vitórias, volta para o passo 3.
- 8. O sistema exibe o ganhador do jogo.
- **Fluxo alternativo**
- *a. A qualquer momento, o usuário decide sair.
- 1. O sistema finaliza o jogo.
- 3a. O jogador demora mais do que 3 minutos para lançar os dados
- 1. O sistema finaliza o jogo

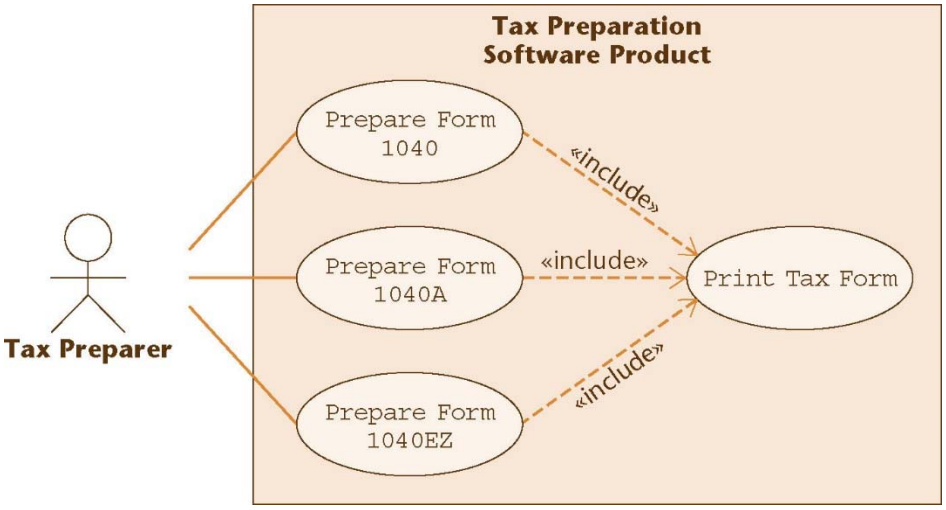
Outro exemplo

- Diagrama de caso de uso:



Outro exemplo





Cenário (instância dos casos de uso)

1. User A presses the Up floor button at floor 3 to request an elevator. User A wishes to go to floor 7.
2. The Up floor button is turned on.
3. An elevator arrives at floor 3. It contains User B, who has entered the elevator at floor 1 and pressed the elevator button for floor 9.
4. The elevator doors open.
5. The timer starts.
   User A enters the elevator.
6. User A presses the elevator button for floor 7.
7. The elevator button for floor 7 is turned on.
8. The elevator doors close after a timeout.
9. The Up floor button is turned off.
10. The elevator travels to floor 7.
11. The elevator button for floor 7 is turned off.
12. The elevator doors open to allow User A to exit from the elevator.
13. The timer starts.
    User A exits from the elevator.
14. The elevator doors close after a timeout.
15. The elevator proceeds to floor 9 with User B.

UML – um breviário

- Diagrama de classes:

- Classe = **nome** + atributos + métodos
  - Nome: identificador da classe
  - Atributo = nome + visibilidade + tipo
  - Método = nome + visibilidade + tipo
    - Visibilidade = público, privado, etc.
    - Tipo = tipo de dados (inteiro, string, etc.)

Diagrama de classes

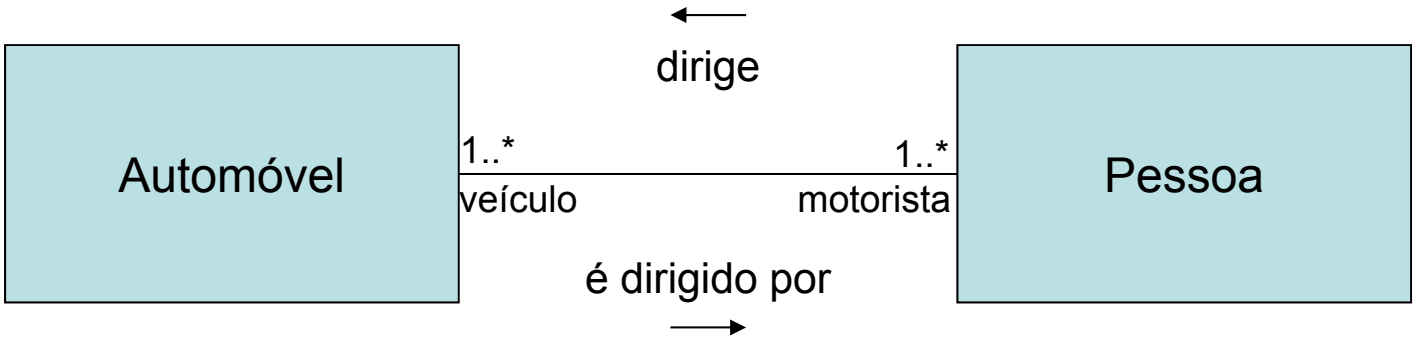
Bank Account
− accountBalance
+ deposit ( ) + withdraw ( )

NOME
Visib atrib: tipo Visib atrib: tipo Visib atrib: tipo (...)
Visib método(par_1: tipo,... par_n:tipo): tipo Visib método(par_1: tipo,... par_n:tipo): tipo Visib método(par_1: tipo,... par_n:tipo): tipo (...)

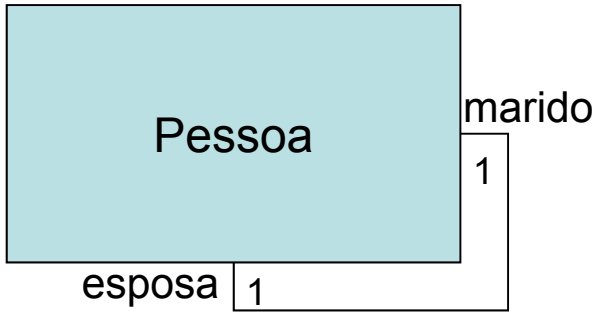
Relacionamentos entre classes

- Associações:
  - Direcionadas
  - Com cardinalidades
  - Com papéis

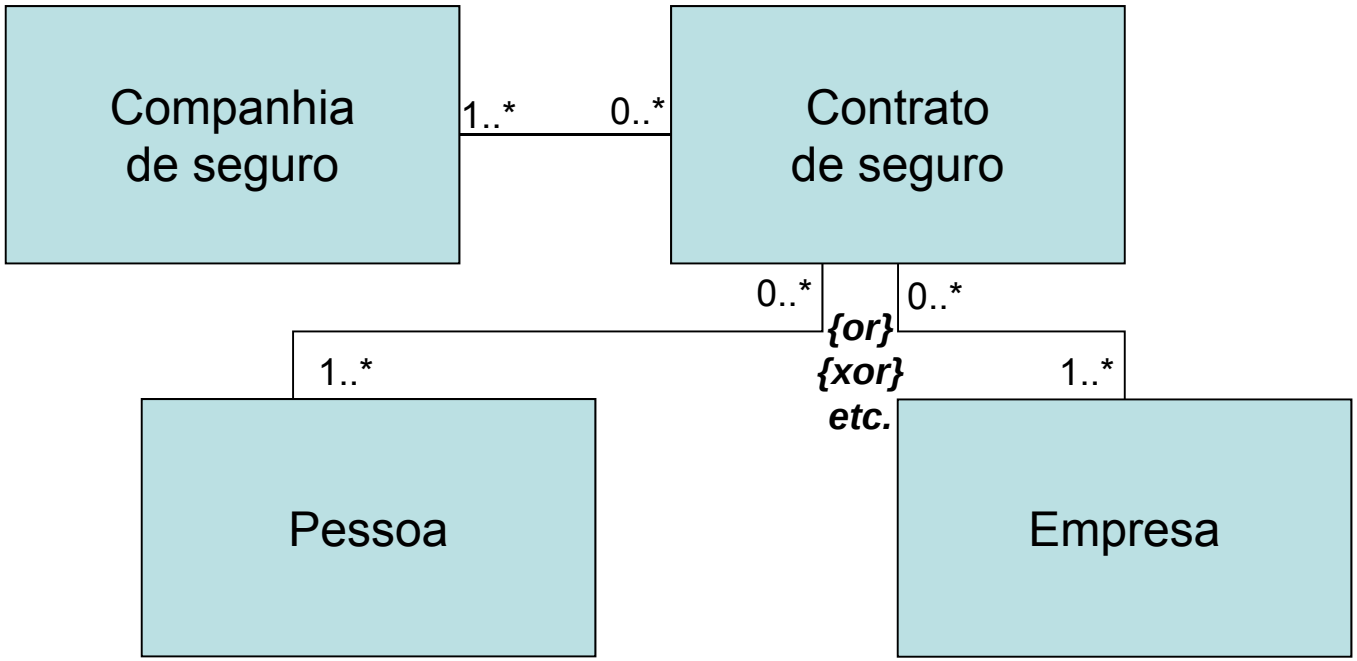
Associações



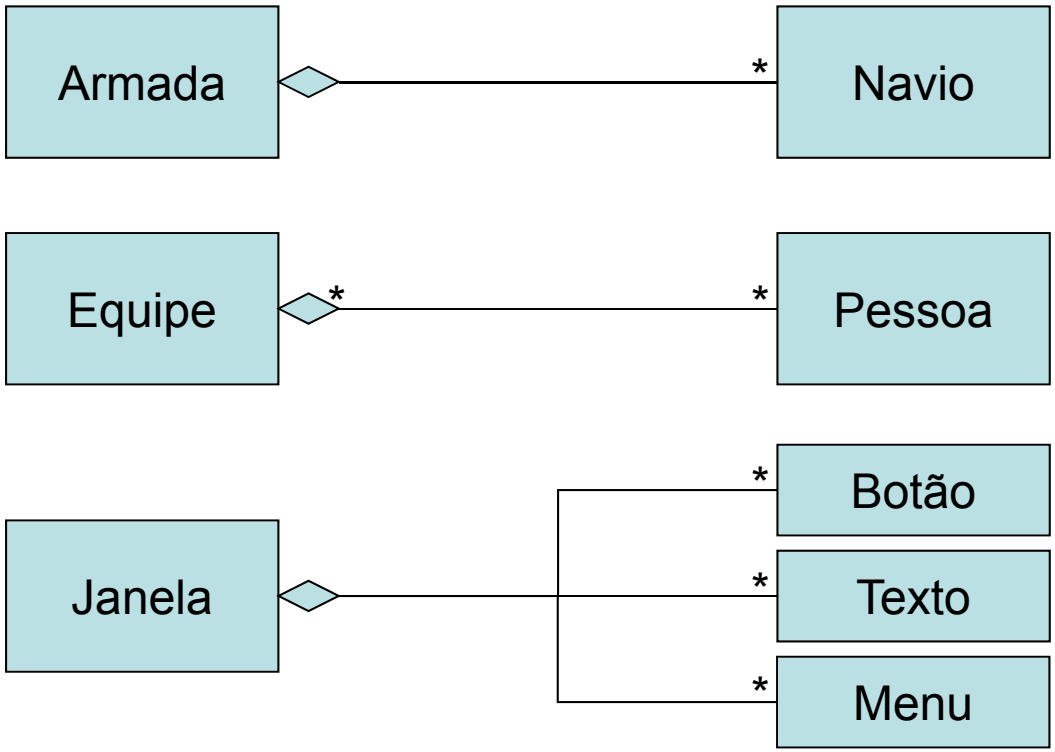
Associações



Associações

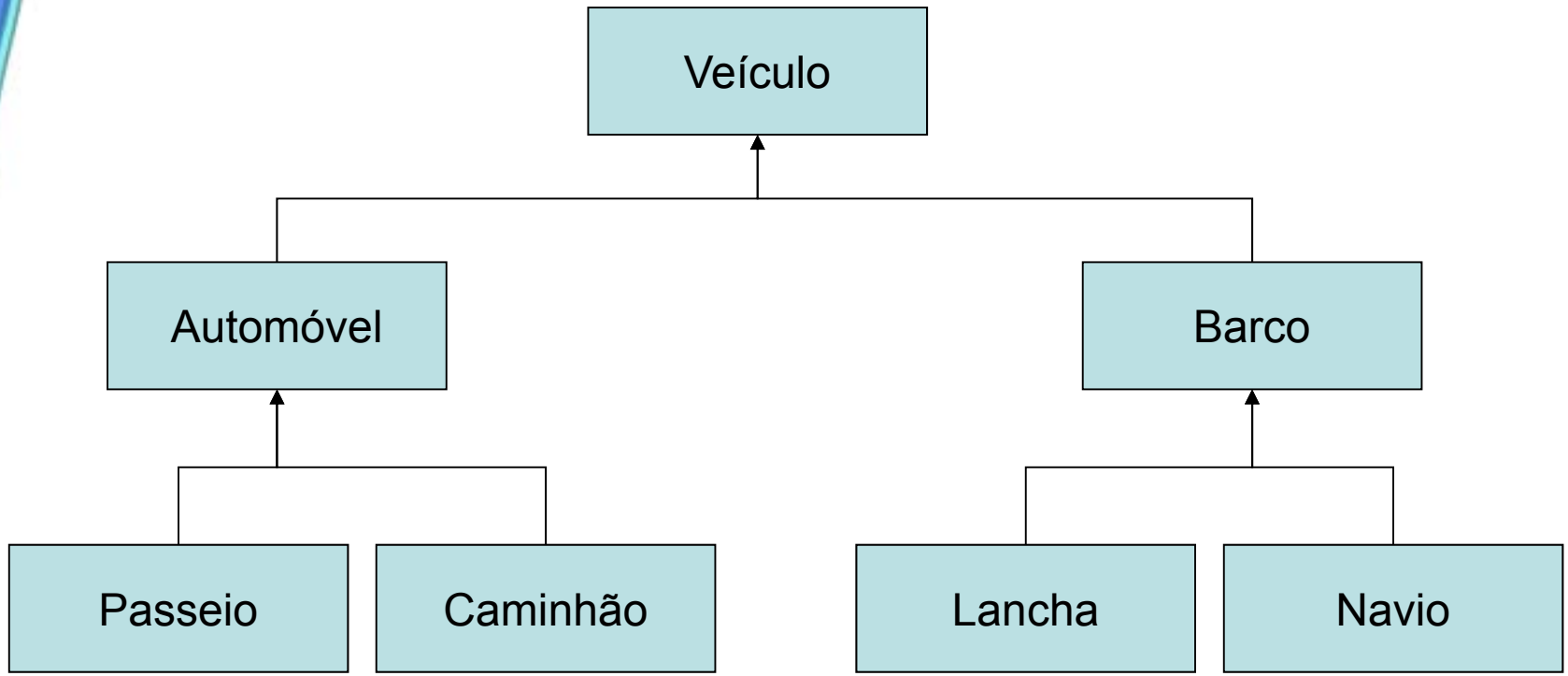


Agregados

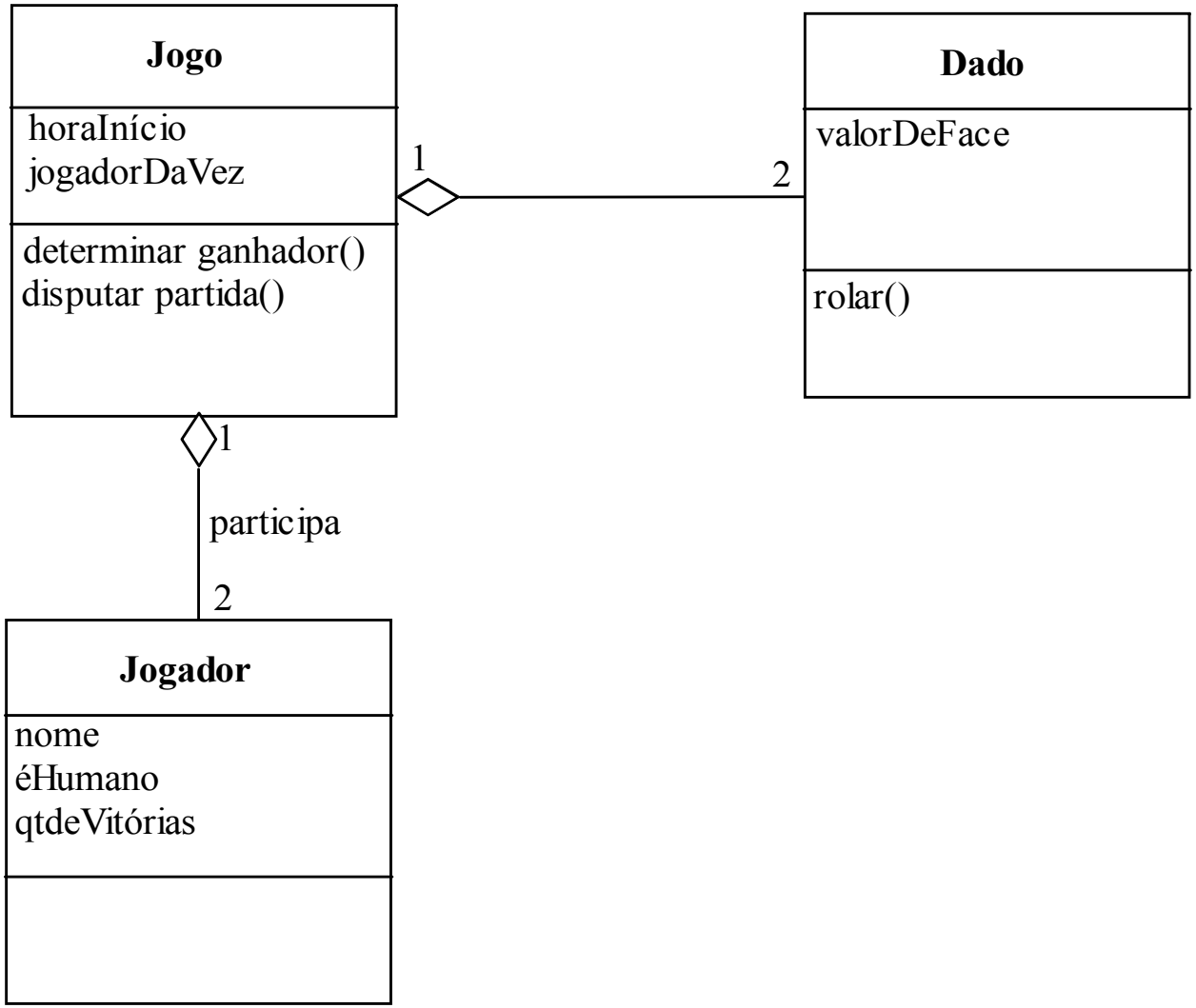


UML – um breviário

- Generalizações:



Exemplo: modelagem conceitual



Exemplo: design

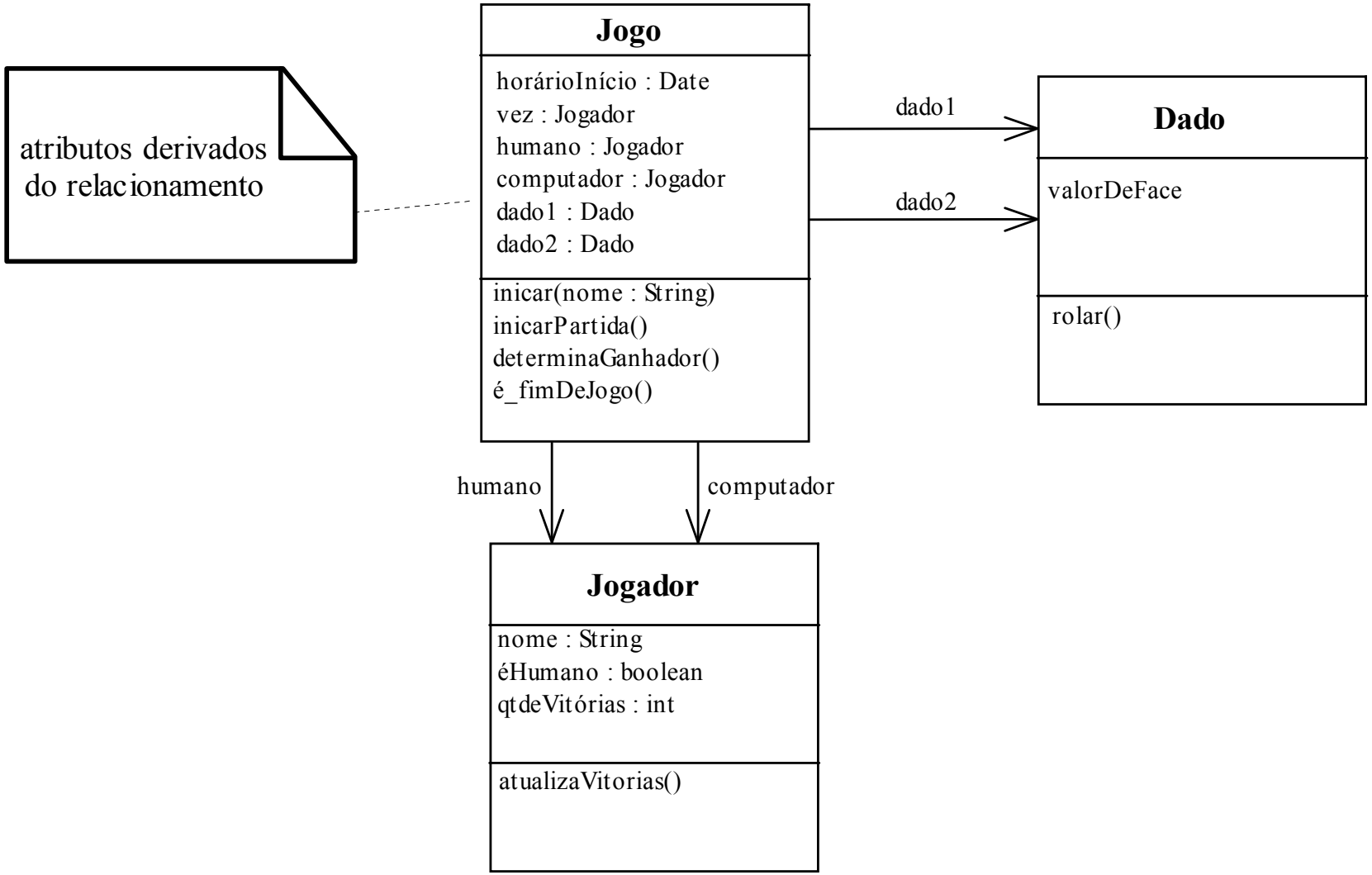
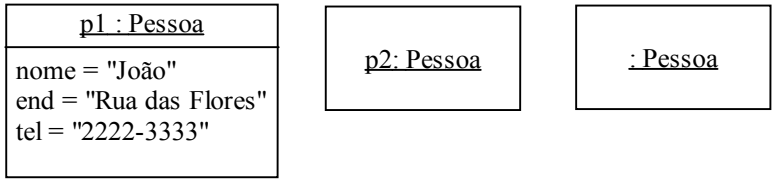
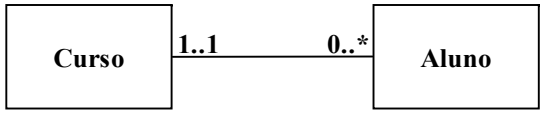


Diagrama de objetos

- Diagrama de objetos:
  - Similar ao diagrama de classes, mas serve para caracterizar como as instâncias das classes – ou seja, objetos específicos – se relacionam





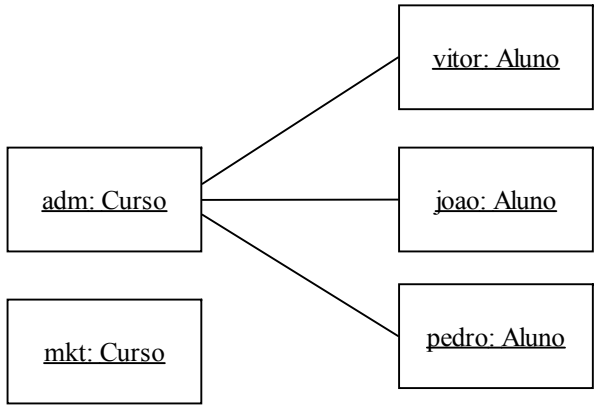
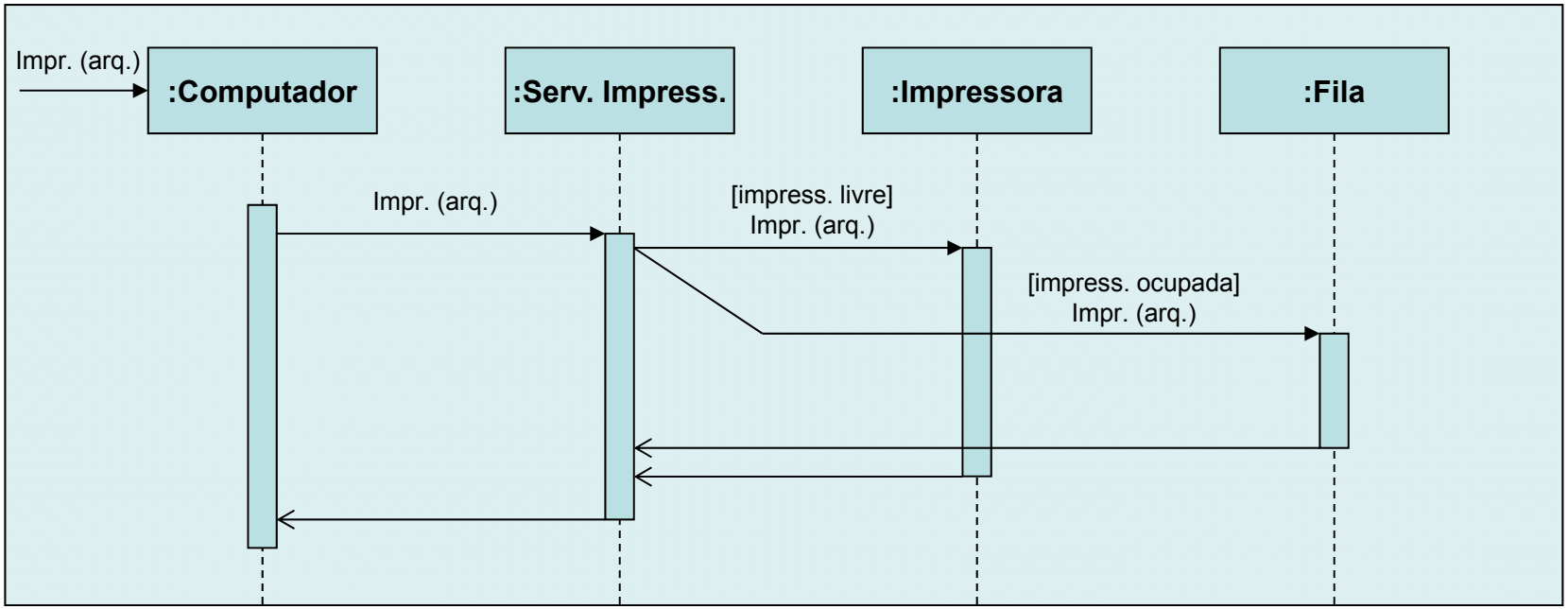
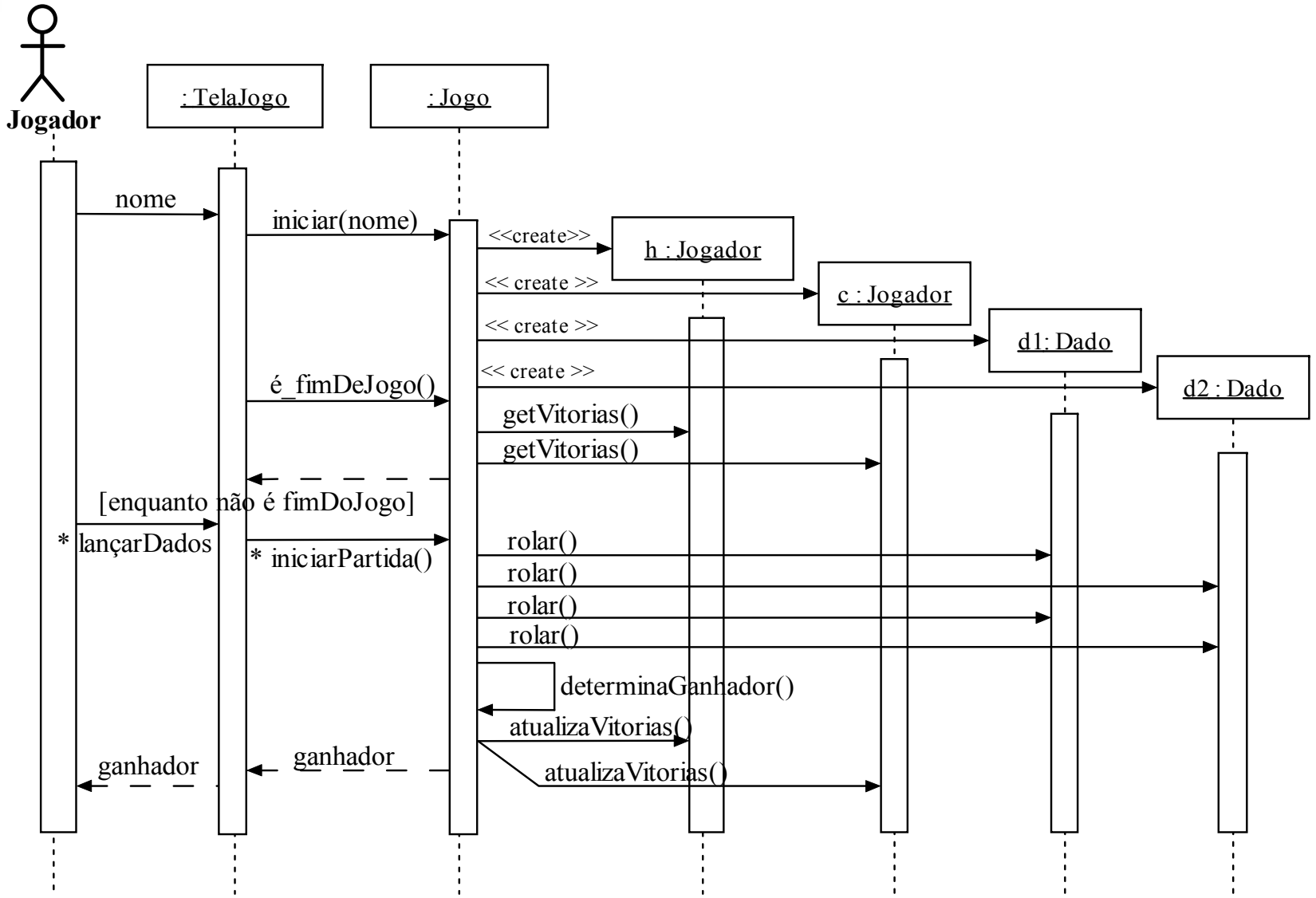


Diagrama de sequência

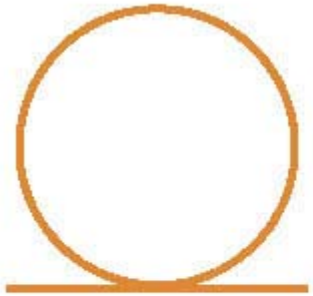
- Diagrama de sequência: organização de troca de mensagens ao longo do tempo



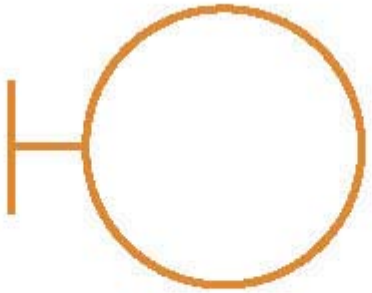
Projeto dinâmico



Tipos de classes e estereótipos



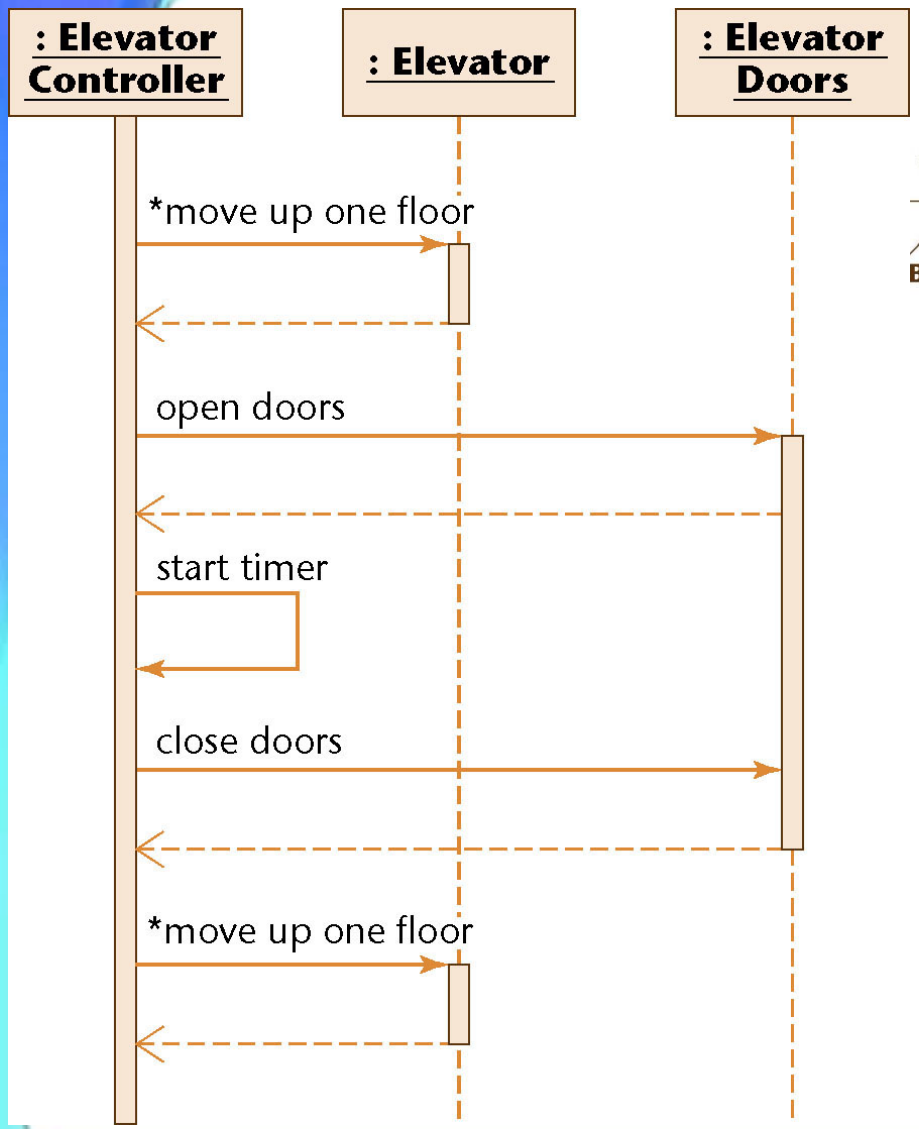
Entity Class



Boundary Class

Control Class

Diagrama de Sequência



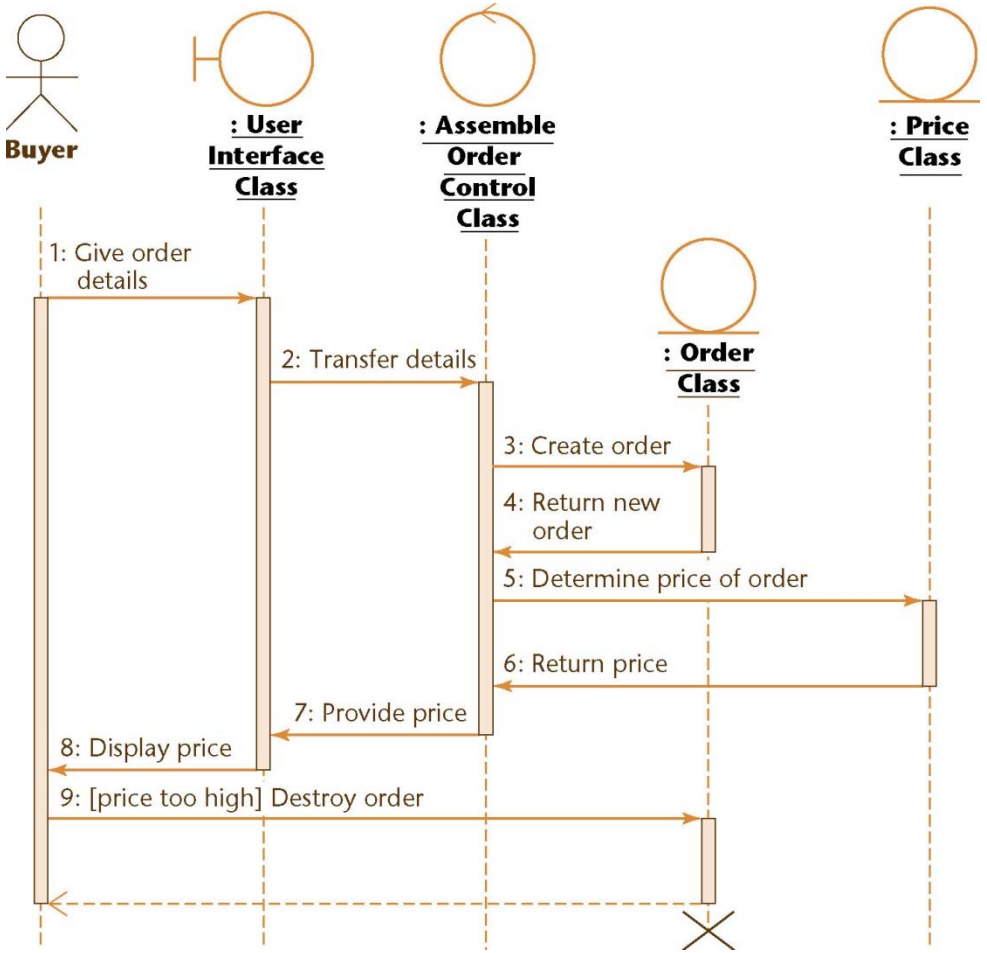


Diagrama de Sequência

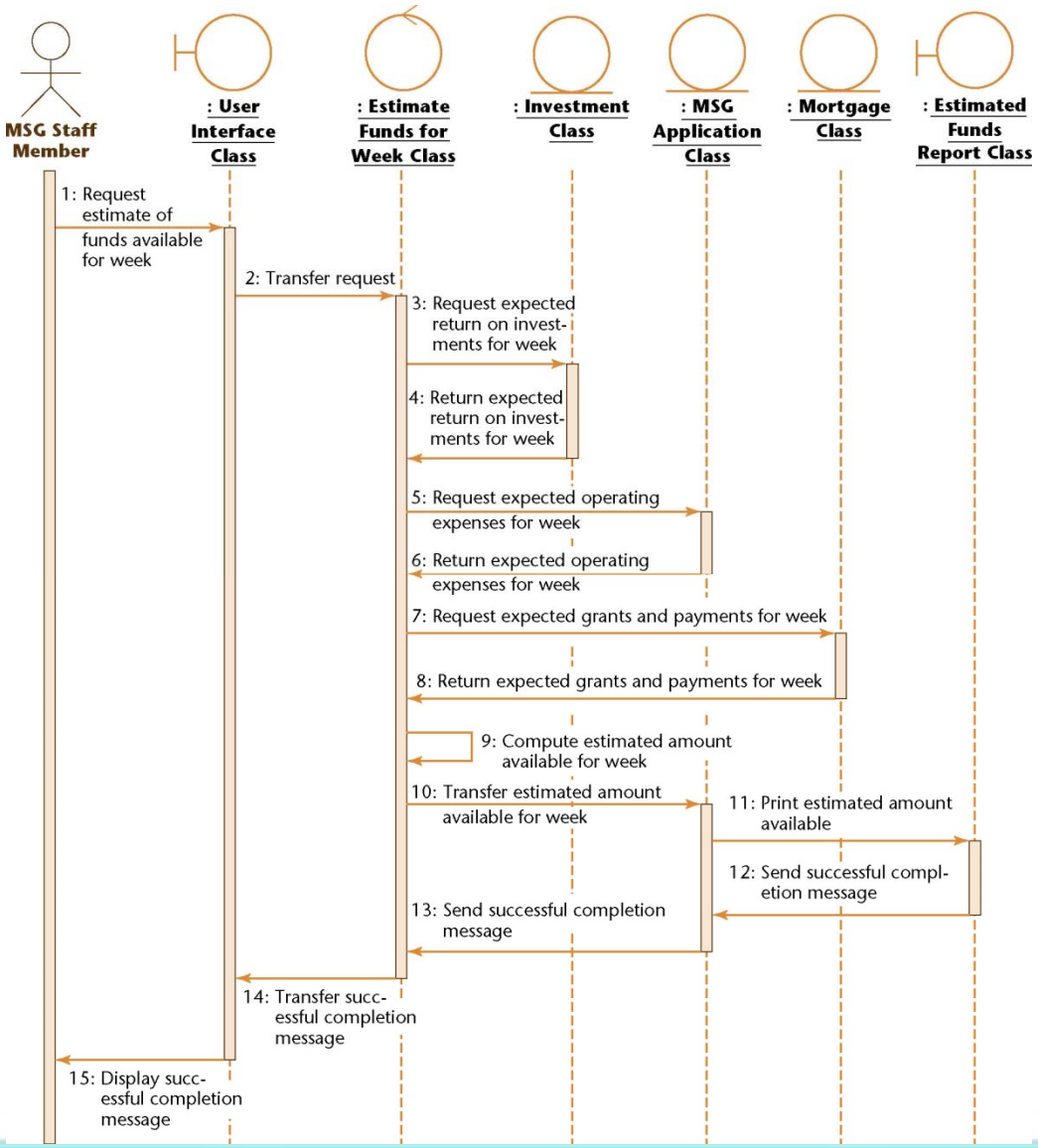


Diagrama de colaboração

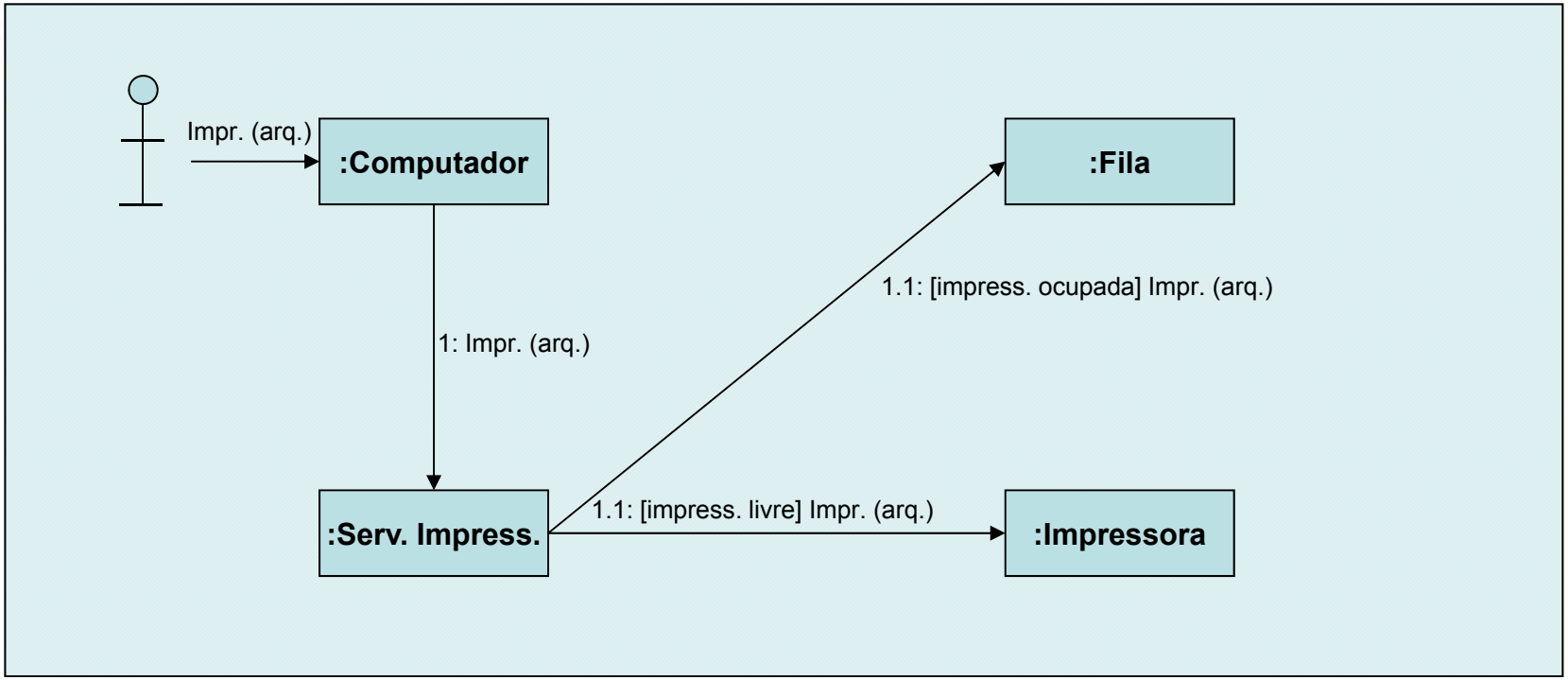


Diagrama de colaboração

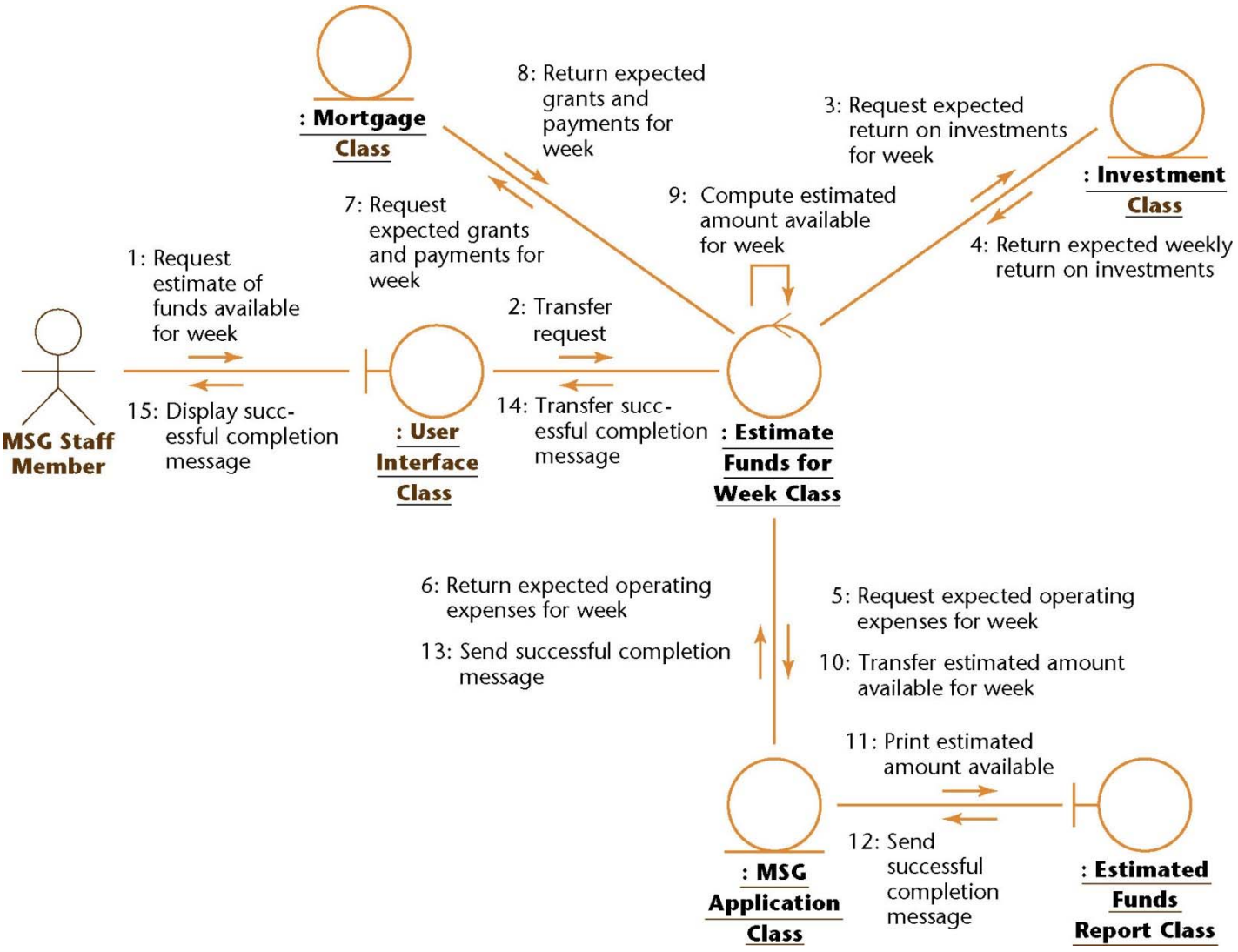


Diagrama de estados

Diagrama de estados: um elevador

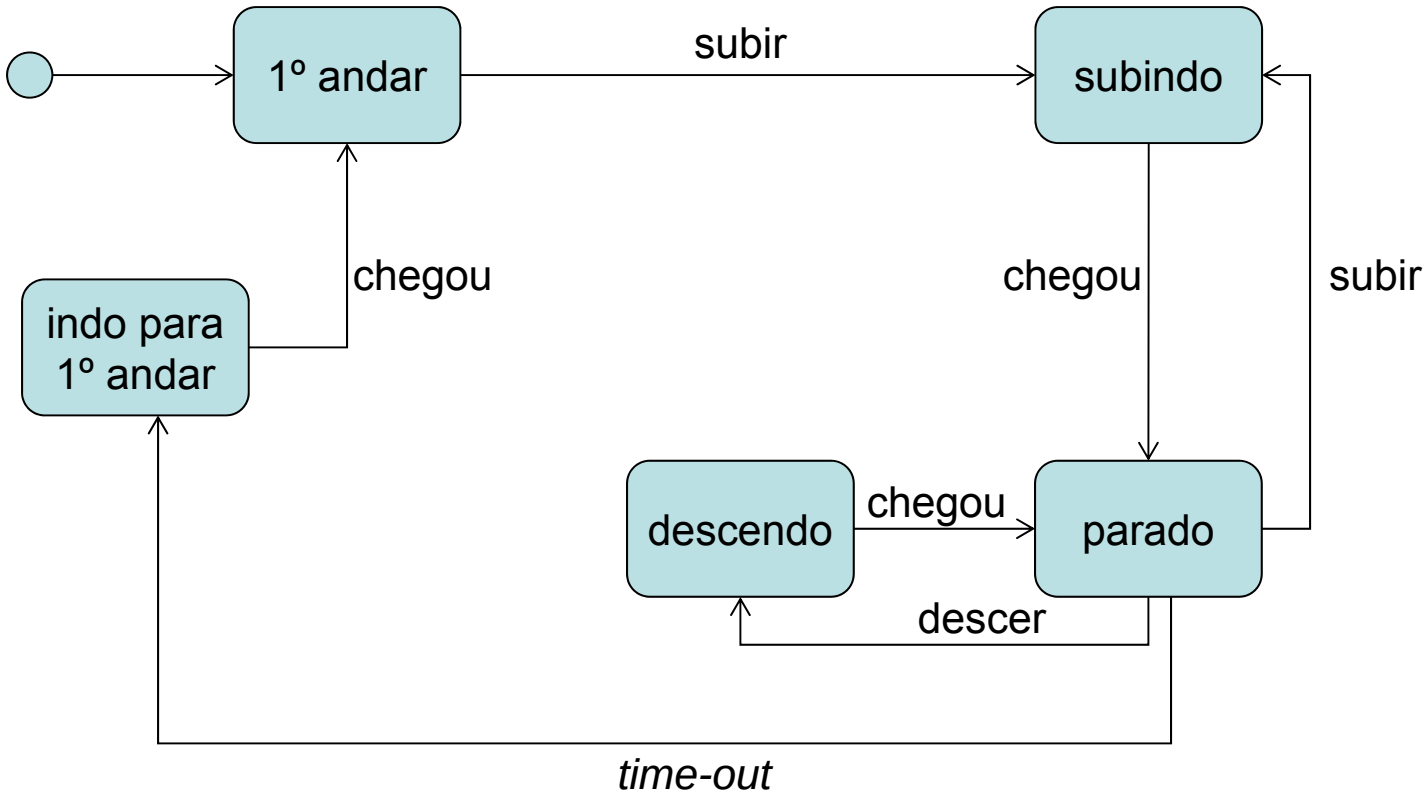


Diagrama de estados

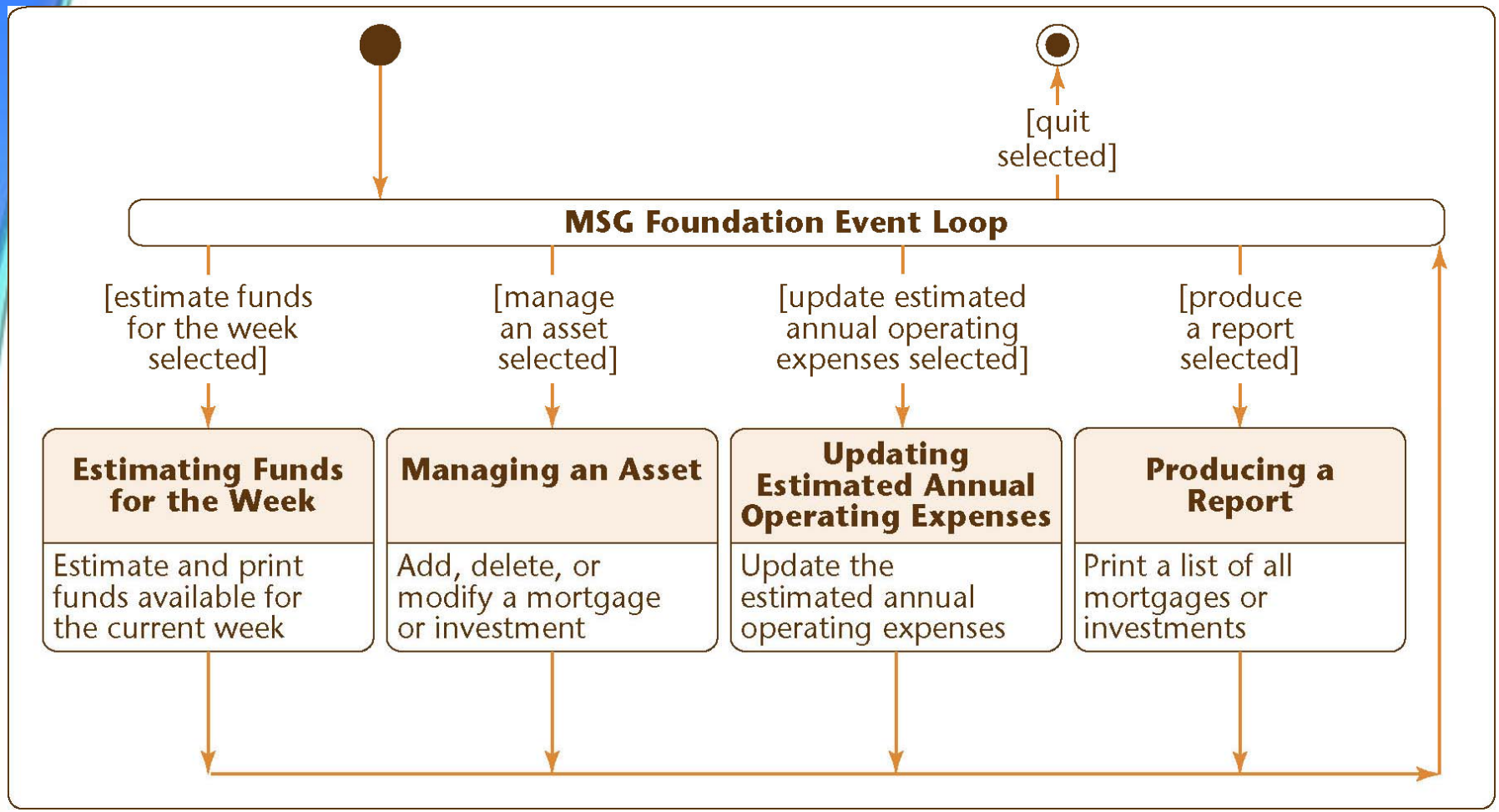


Diagrama de estados

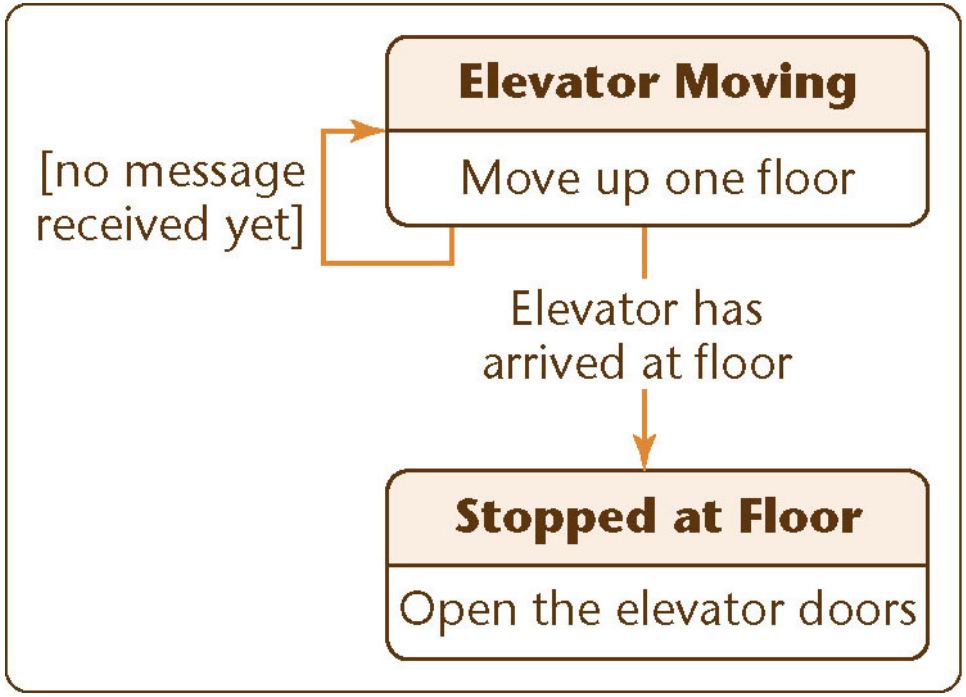


Diagrama de estados mais elaborado

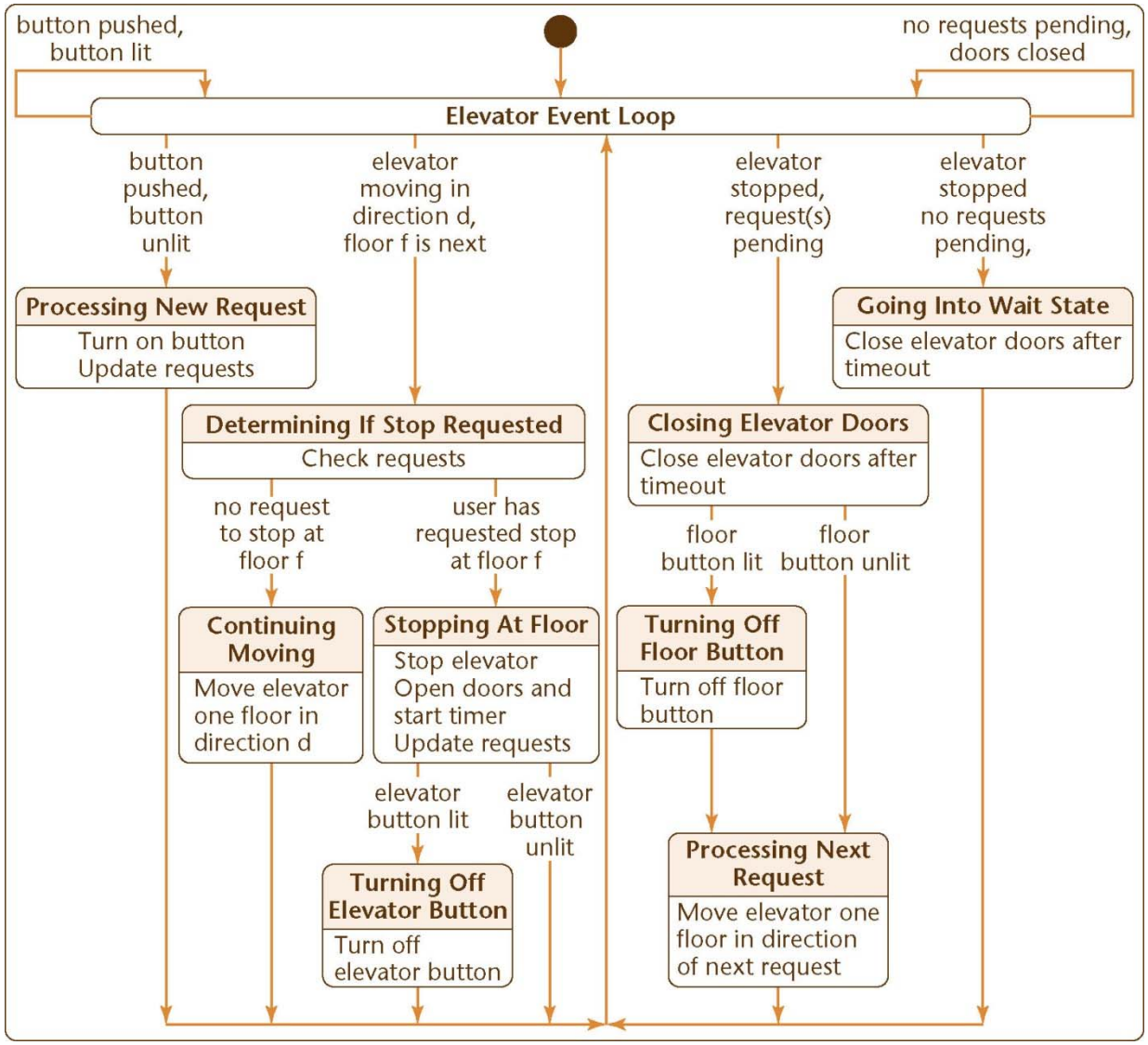


Diagrama de atividades

- Diagrama de atividades: fluxo de operações de um método específico

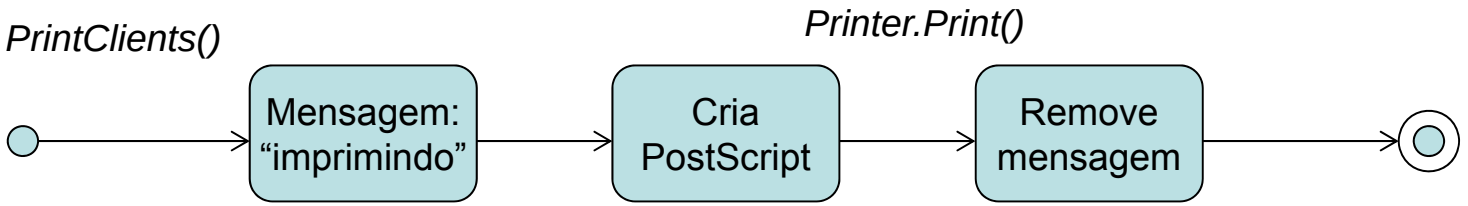


Diagrama de atividades

- *threads* e concorrência

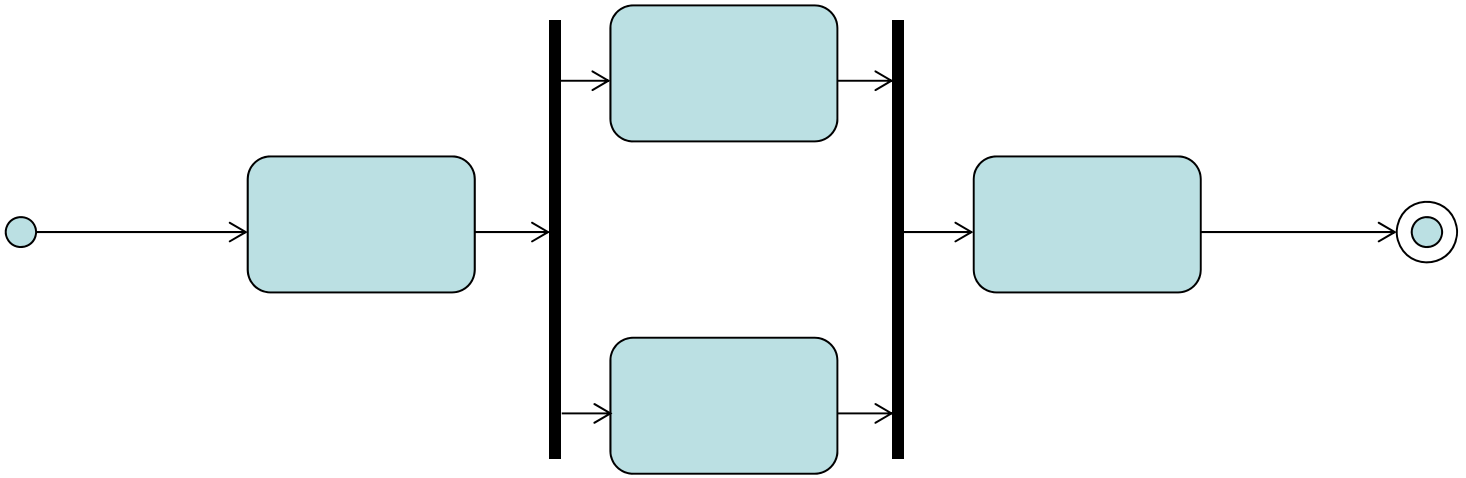


Diagrama de atividades

- *threads* e concorrência – sincronização

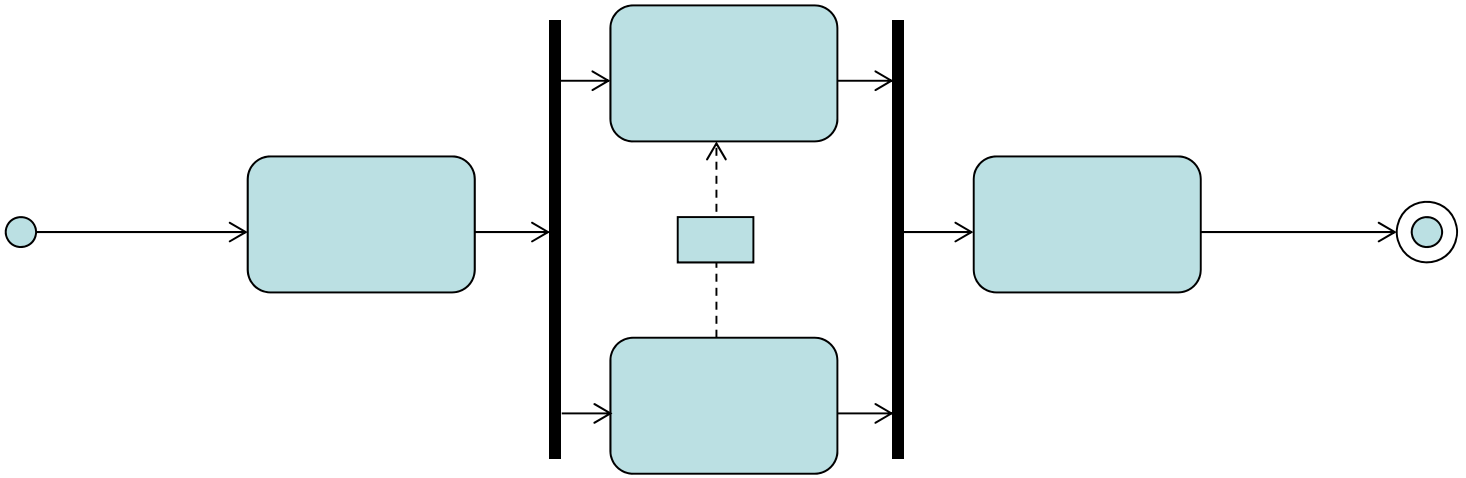


Diagrama de atividades

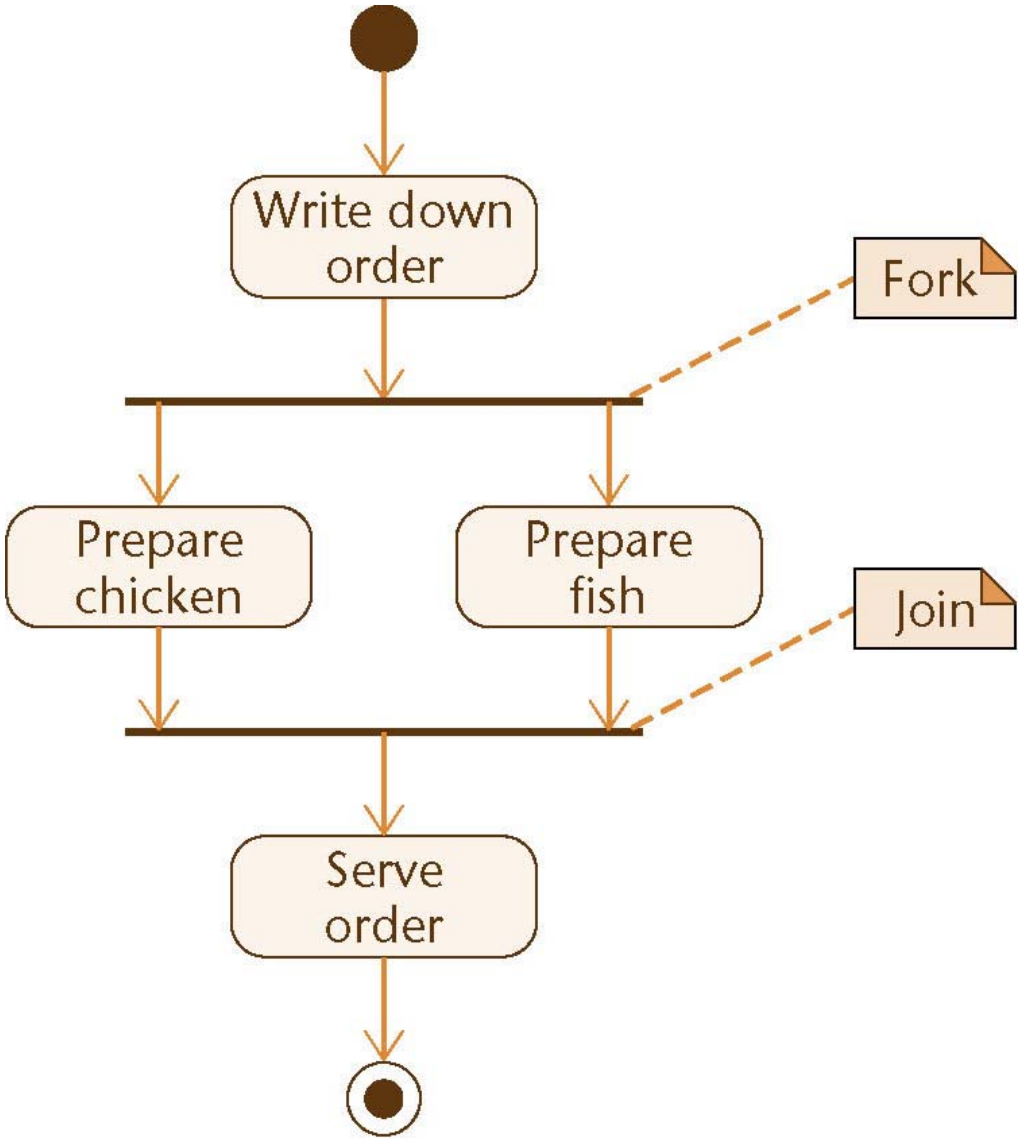
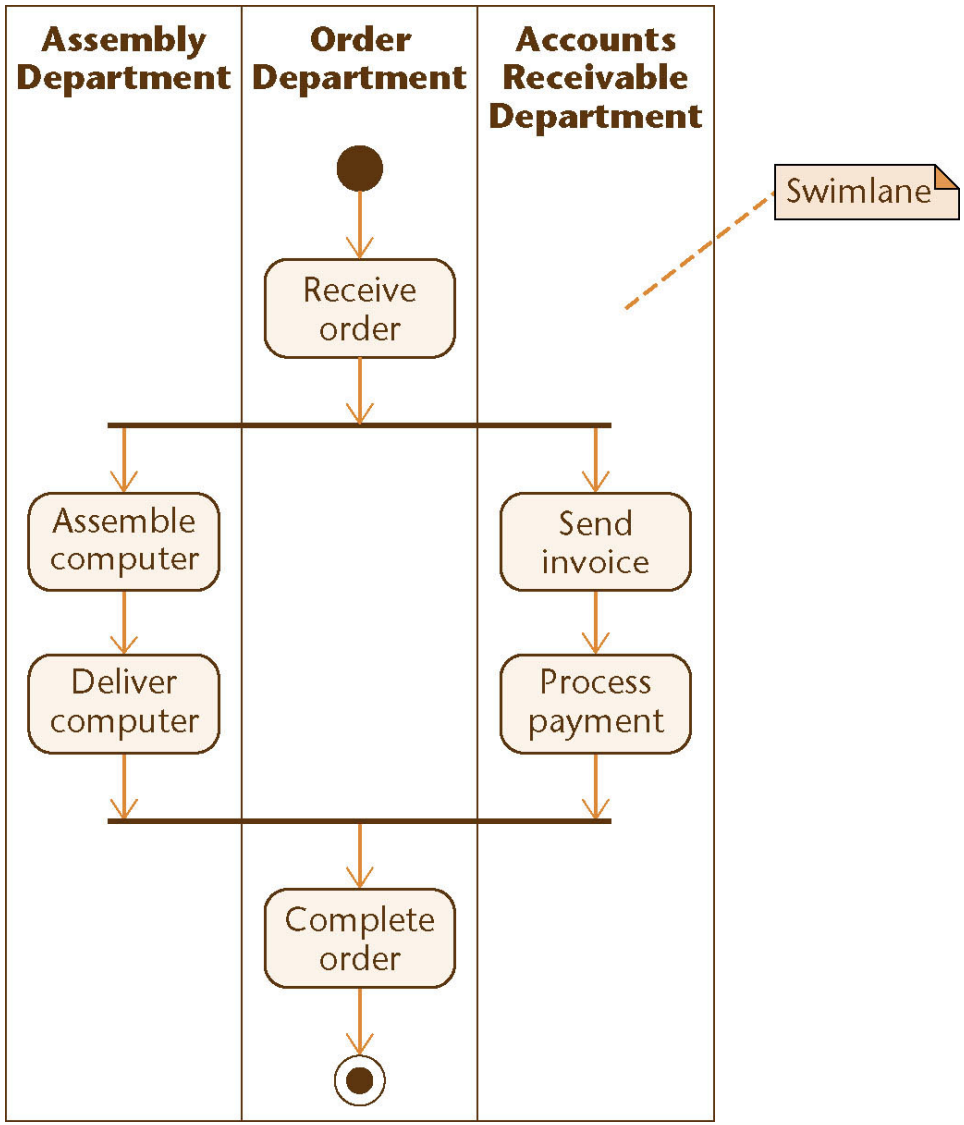


Diagrama de atividades com raias



Pacotes

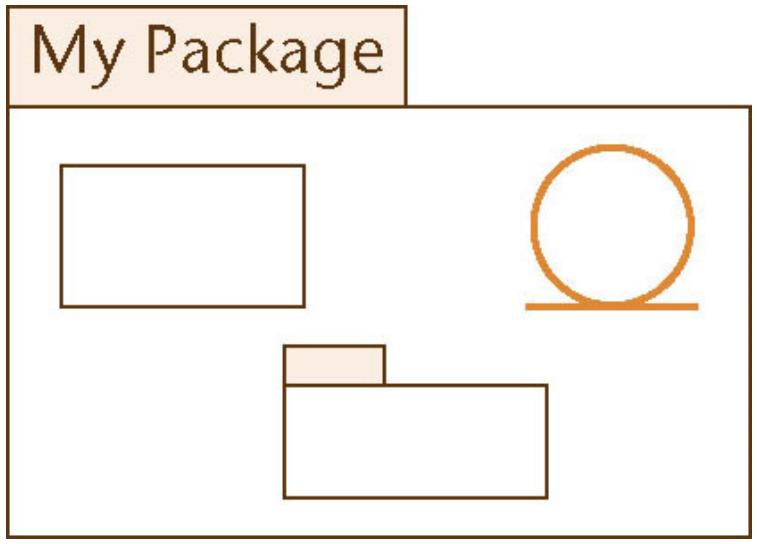


Diagrama de componentes

- Componentes: componentes de software e suas inter-dependências

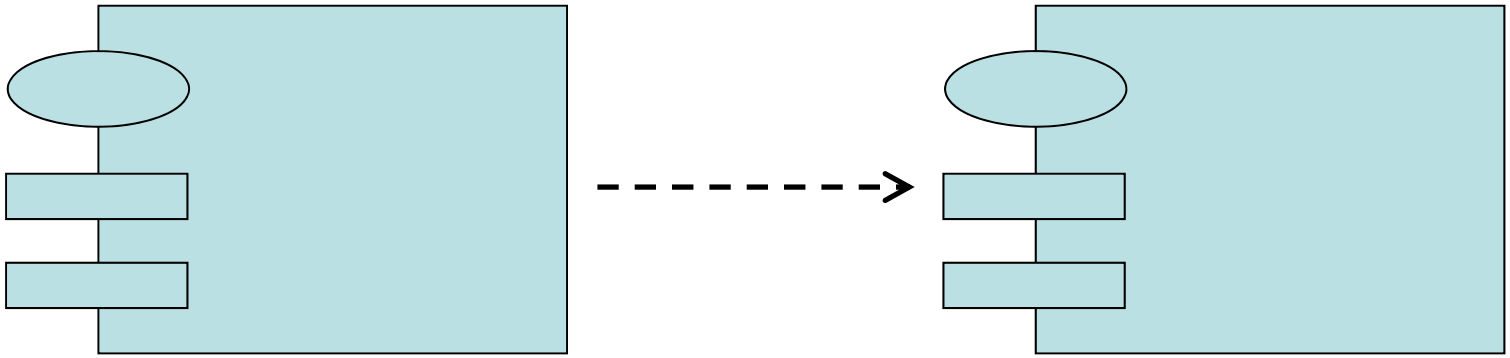
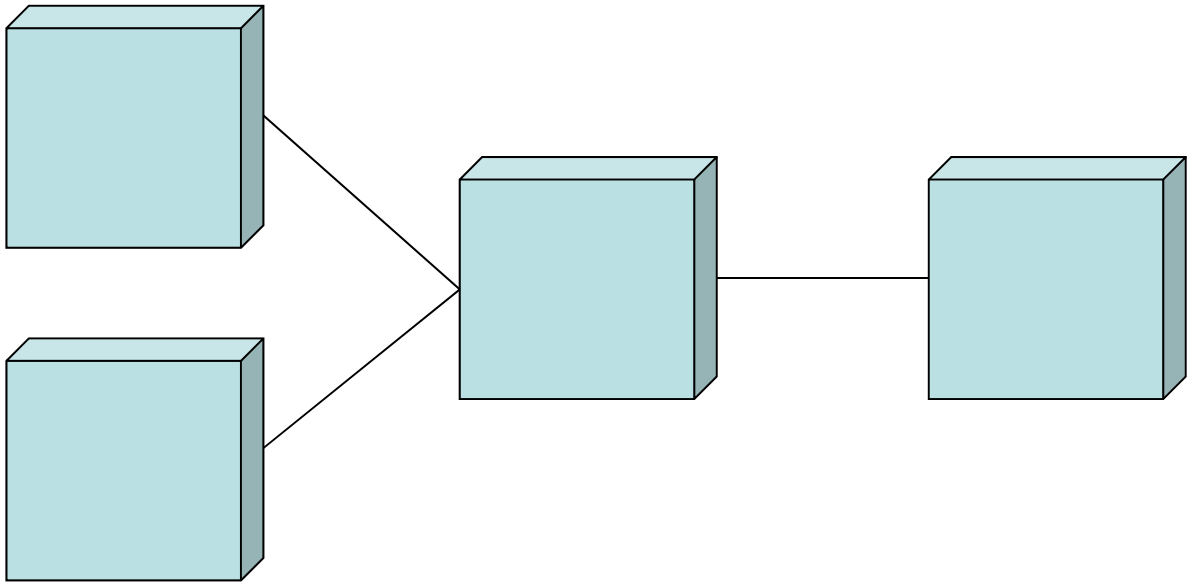


Diagrama de implantação

- componentes de software, localização física e processos físicos de comunicação



Referências desta aula

- Livro Schach
- Slides Prof. Flavio Correa