Revisão para a Prova 2

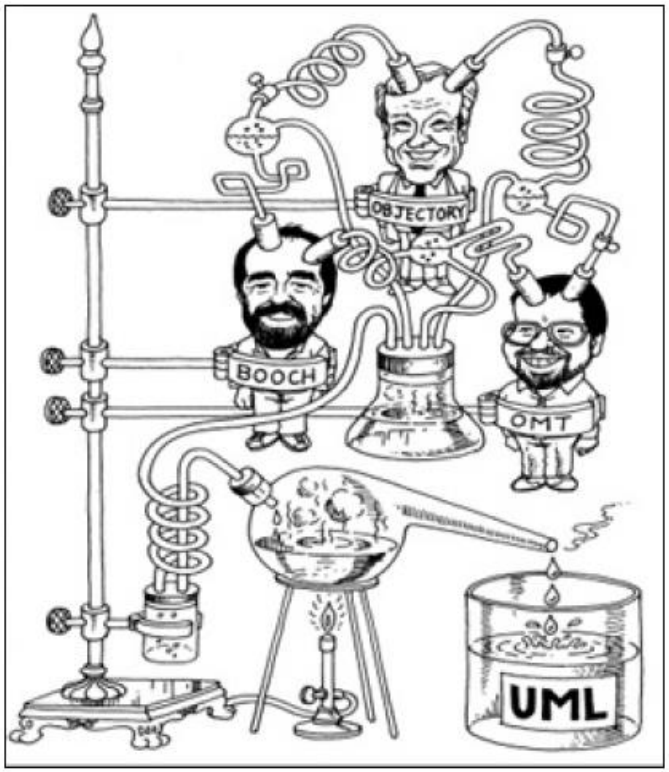
Eduardo Figueiredo

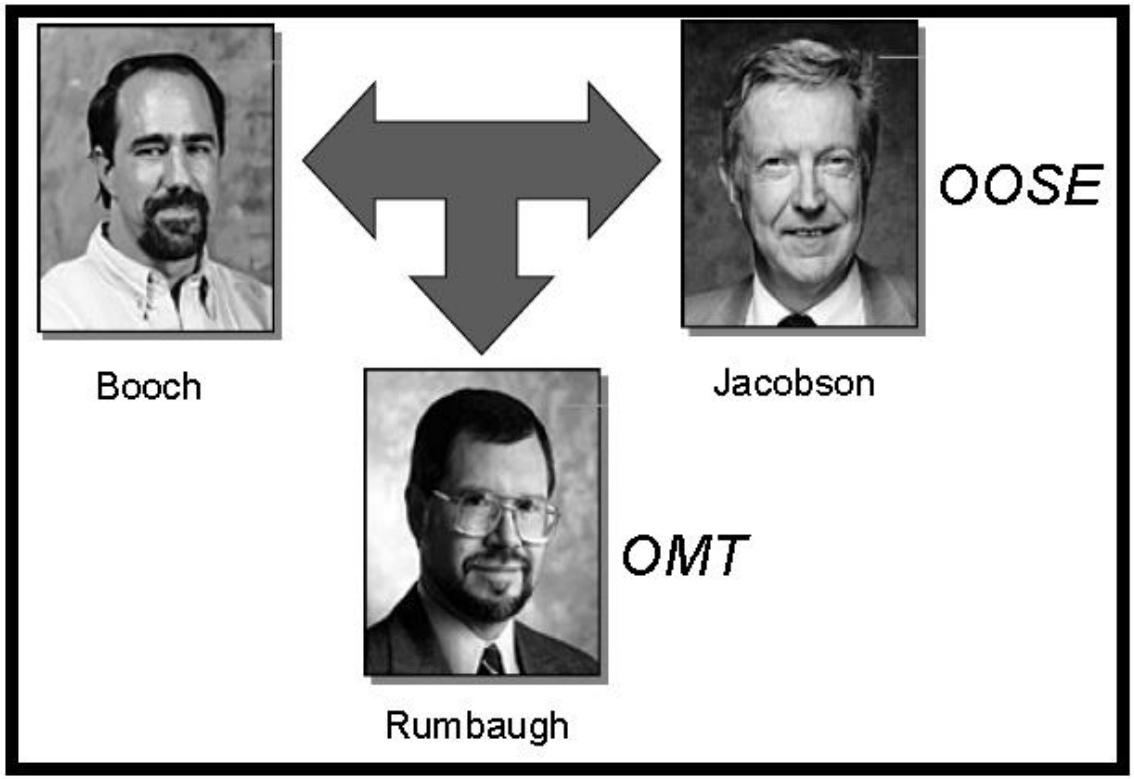
http://www.dcc.ufmg.br/~figueiredo

dcc603@dcc.ufmg.br

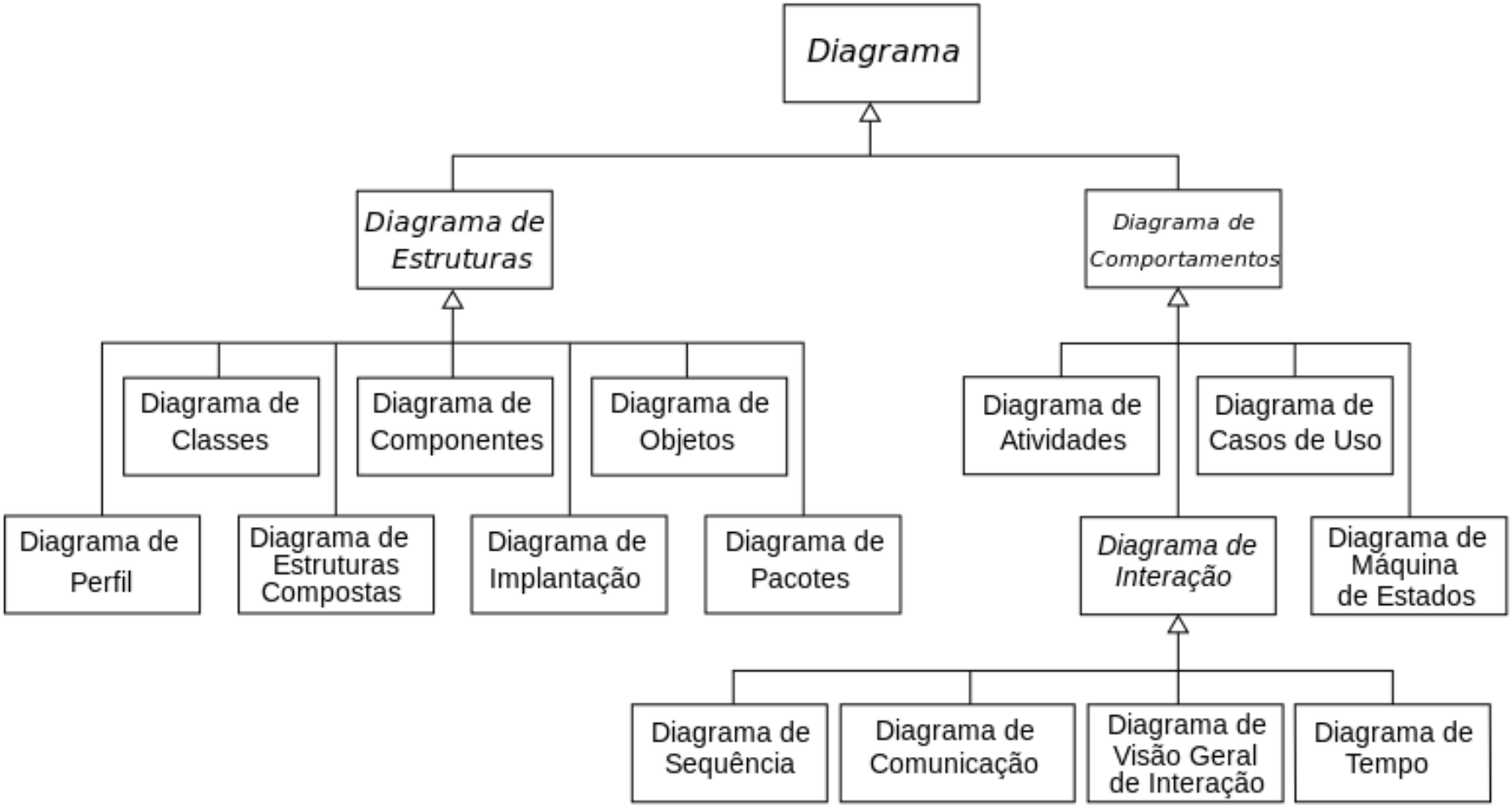
09 Setembro 2020

Aula 08: UML





Classificação dos Diagramas



Diagramas Estruturais

- Diagrama de Perfil
- **Diagrama de Classes**
- Diagrama de Estruturas Compostas
- **Diagrama de Objetos**
- **Diagrama de Componentes**
- **Diagrama de Implantação**
- Diagrama de Pacotes

Diagrama de Classes

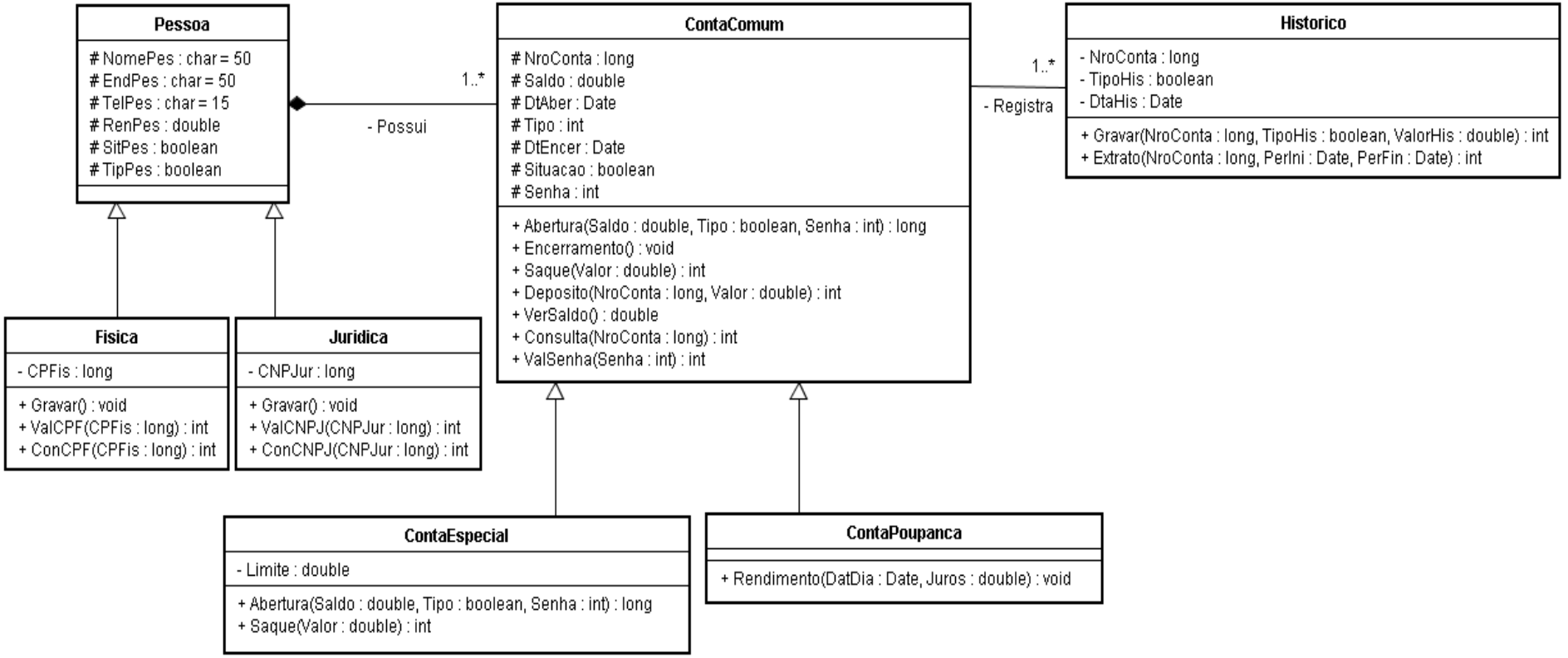


Diagrama de Objetos

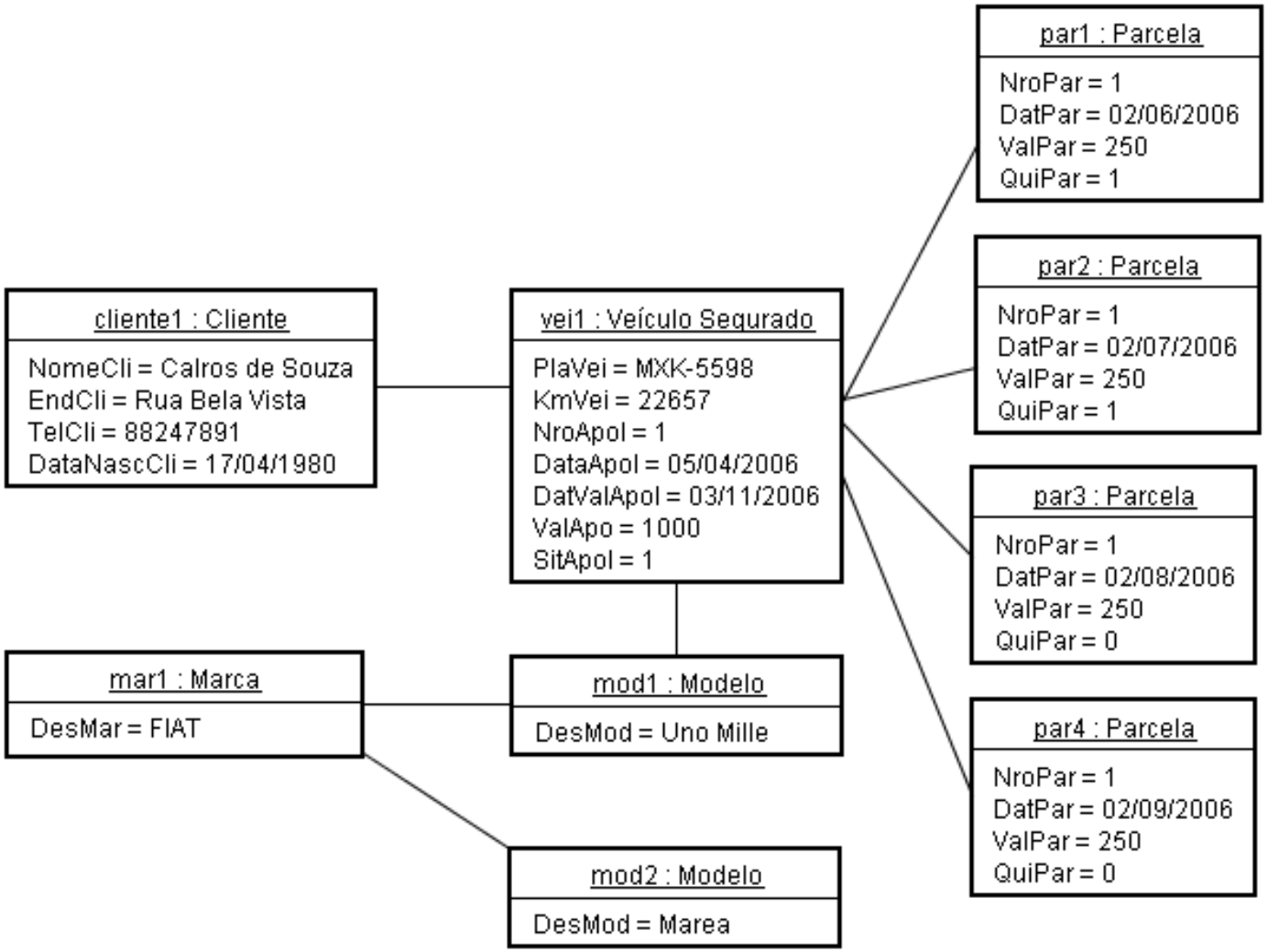


Diagrama de Componentes

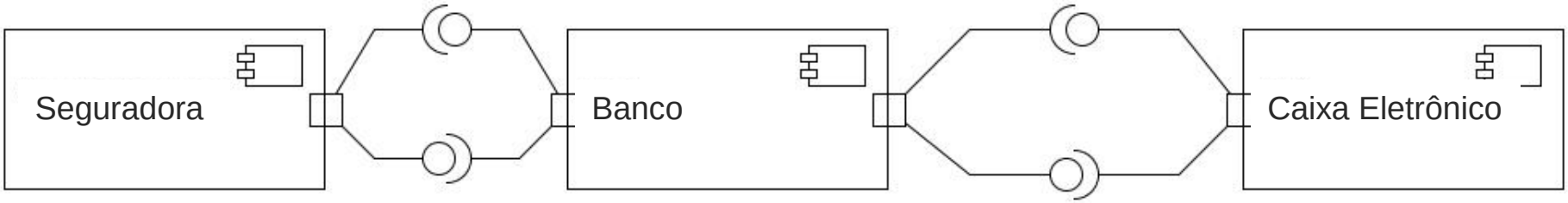
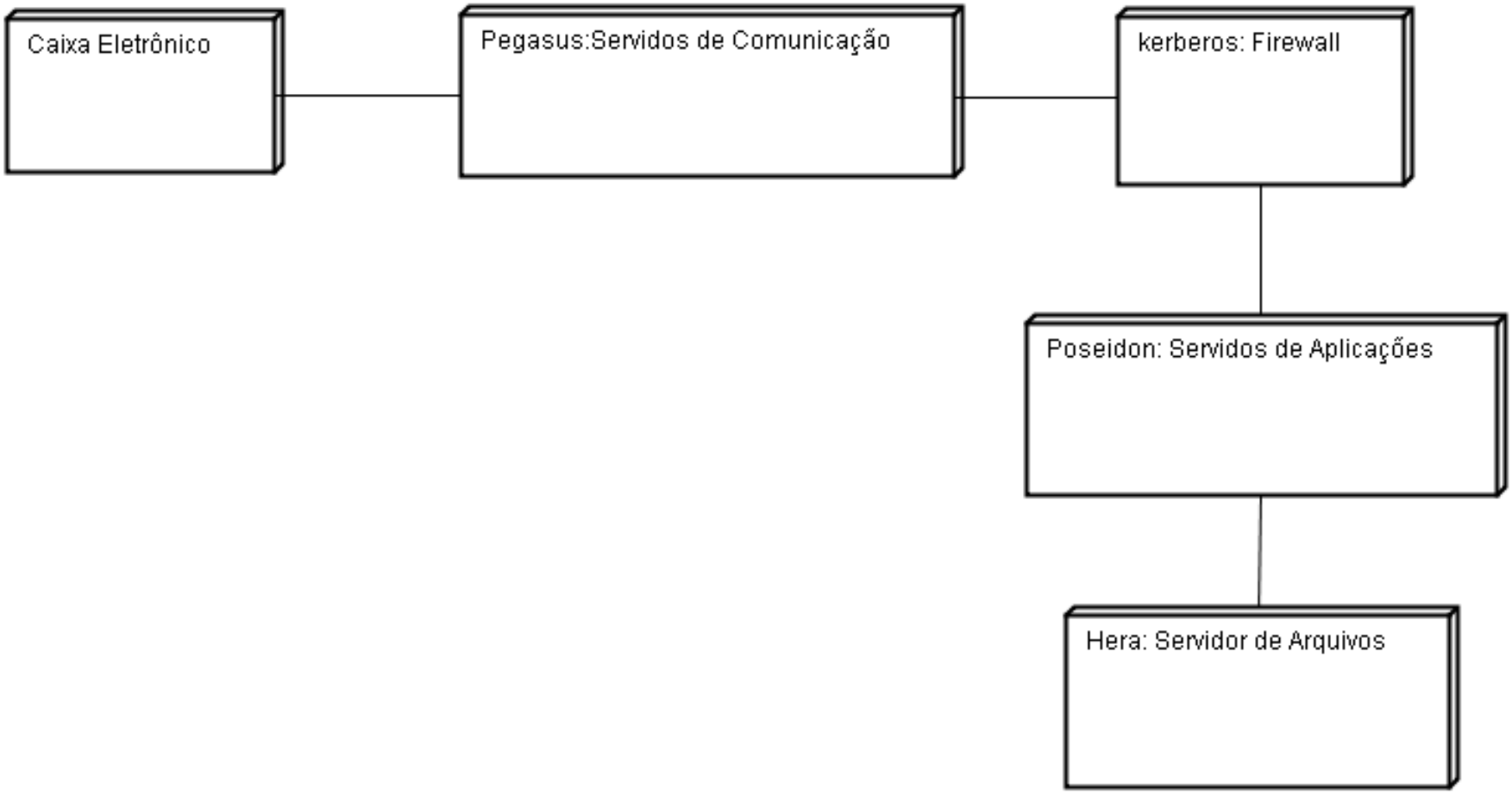


Diagrama de Implantação



Diagramas Comportamentais

- **Diagrama de Casos de Uso**
- **Diagrama de Sequência**
- **Diagrama de Comunicação**
- **Diagrama de Estados**
- **Diagrama de Atividades**
- Diagrama de Visão Geral de Interação
- Diagrama de Tempo

Diagrama de Caso de Uso

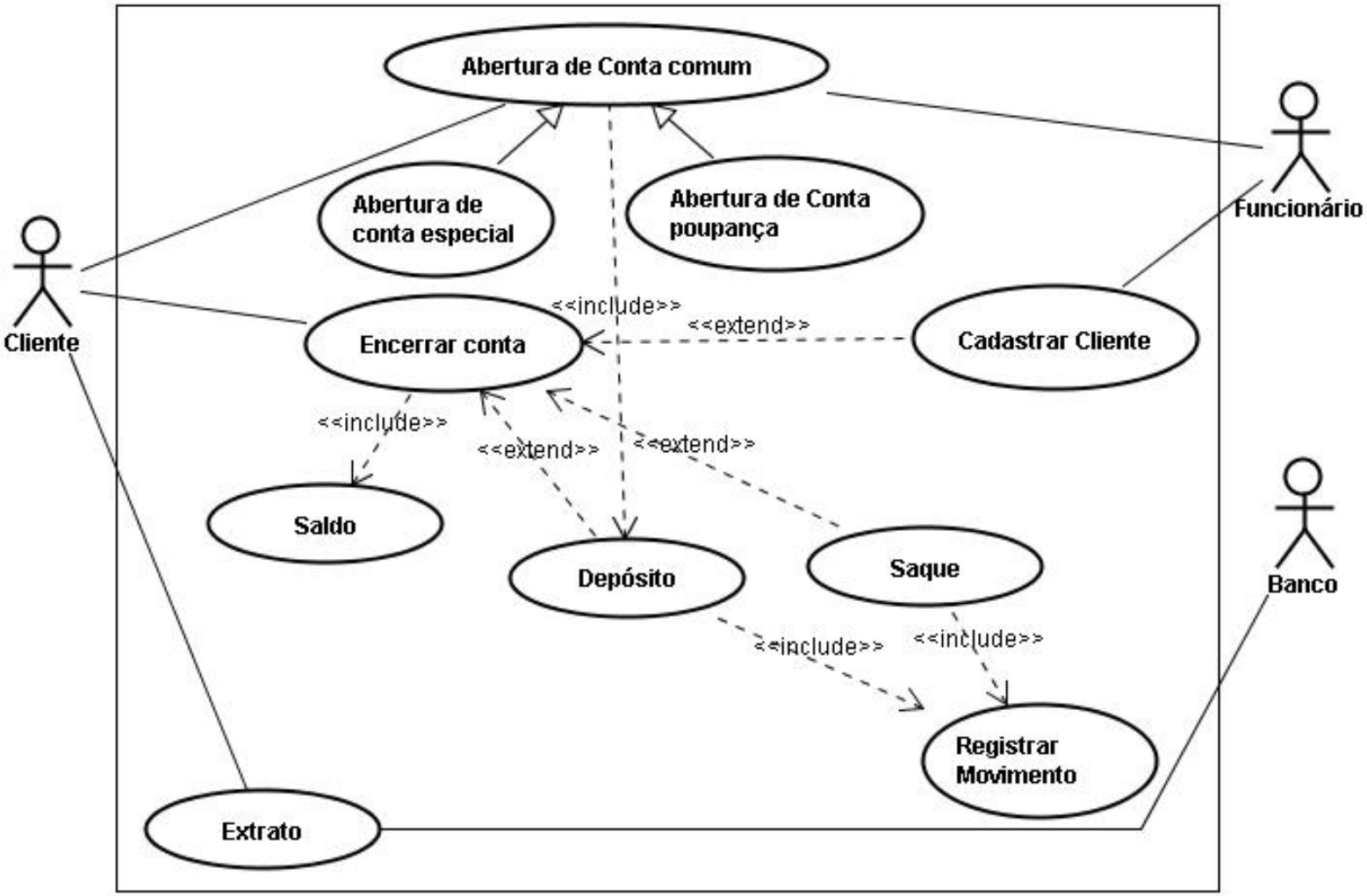


Diagrama de Sequência

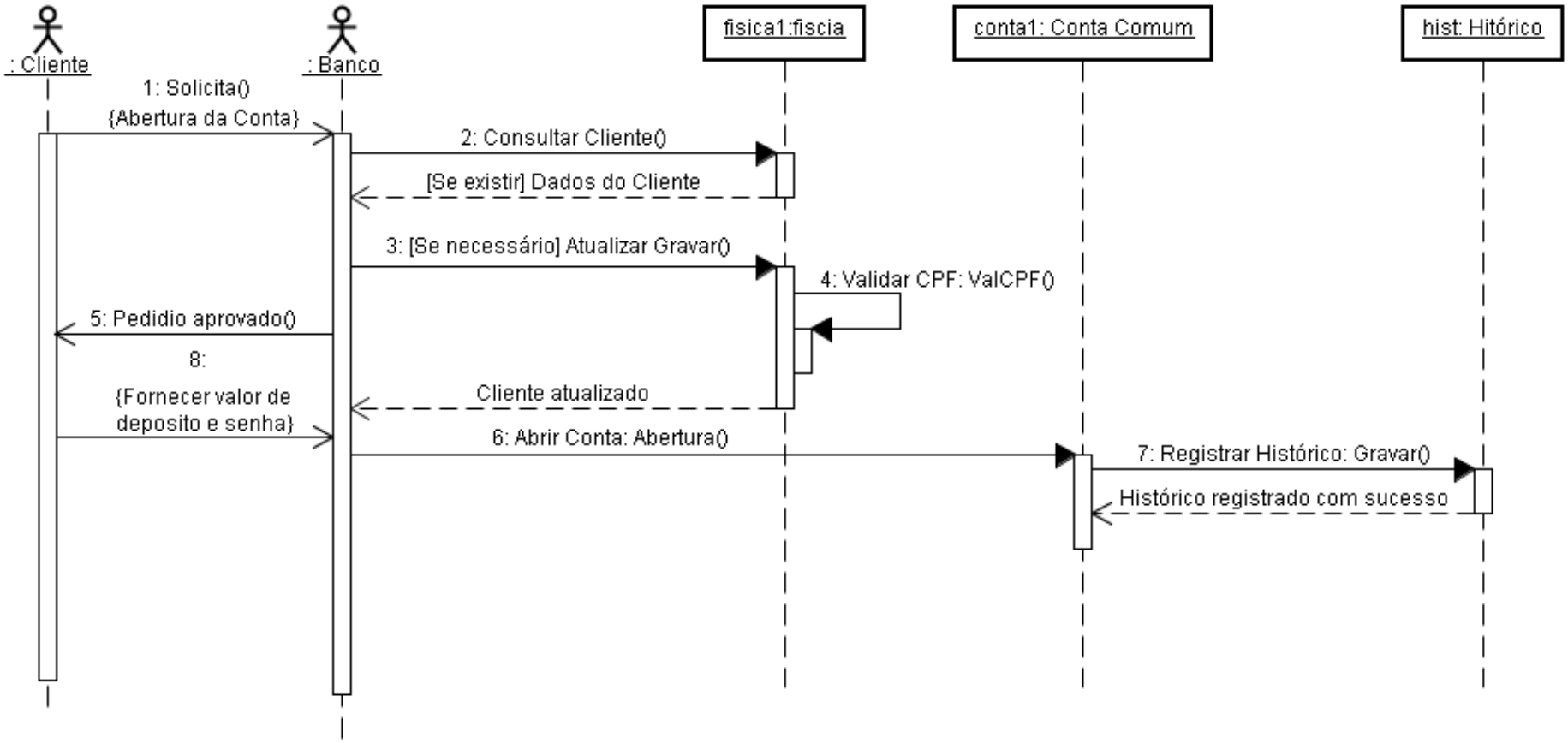


Diagrama de Comunicação

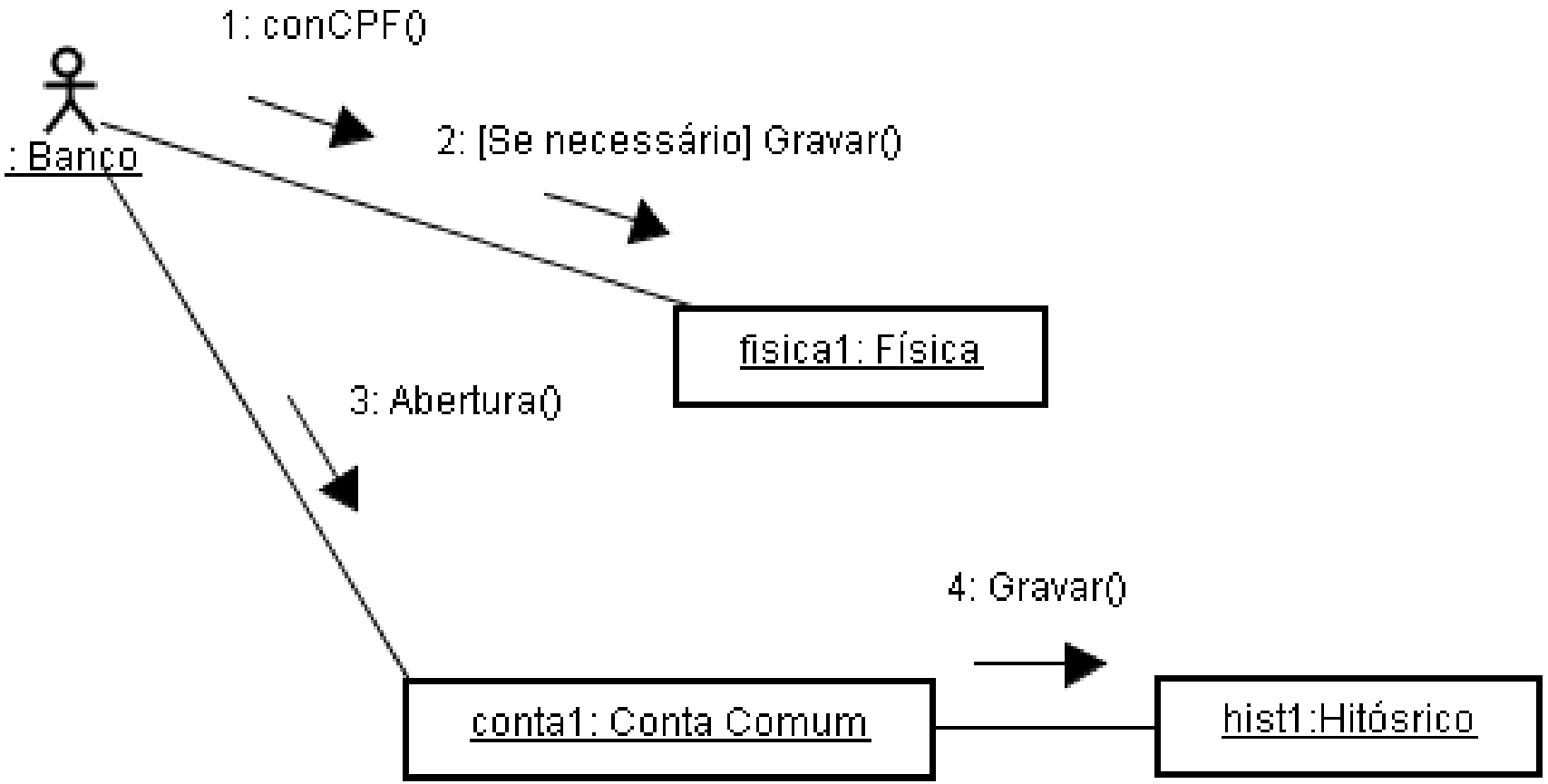
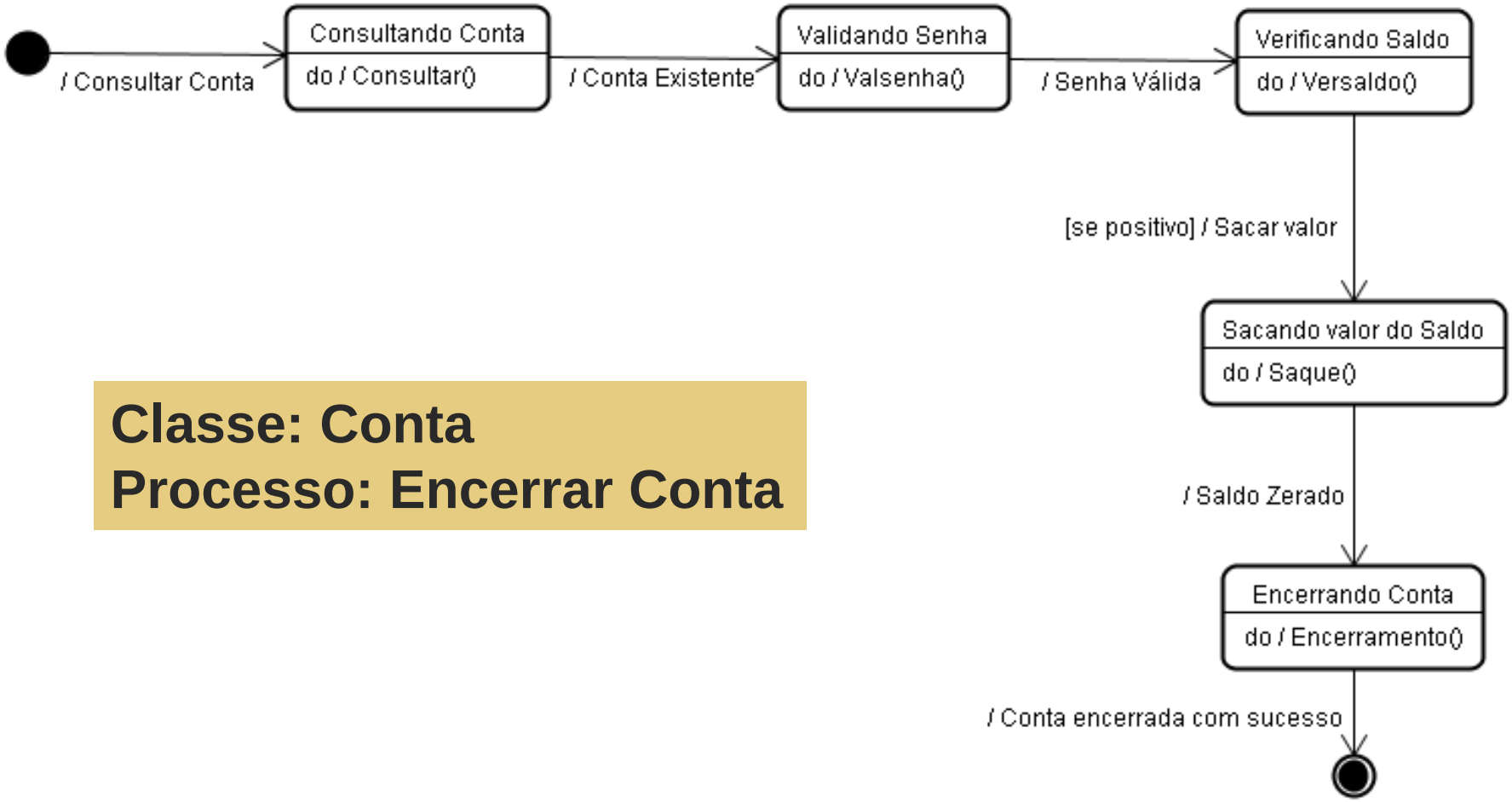


Diagrama de Estados



Classe: Conta
Processo: Encerrar Conta

Diagrama de Atividades

Processo: Validar Conta

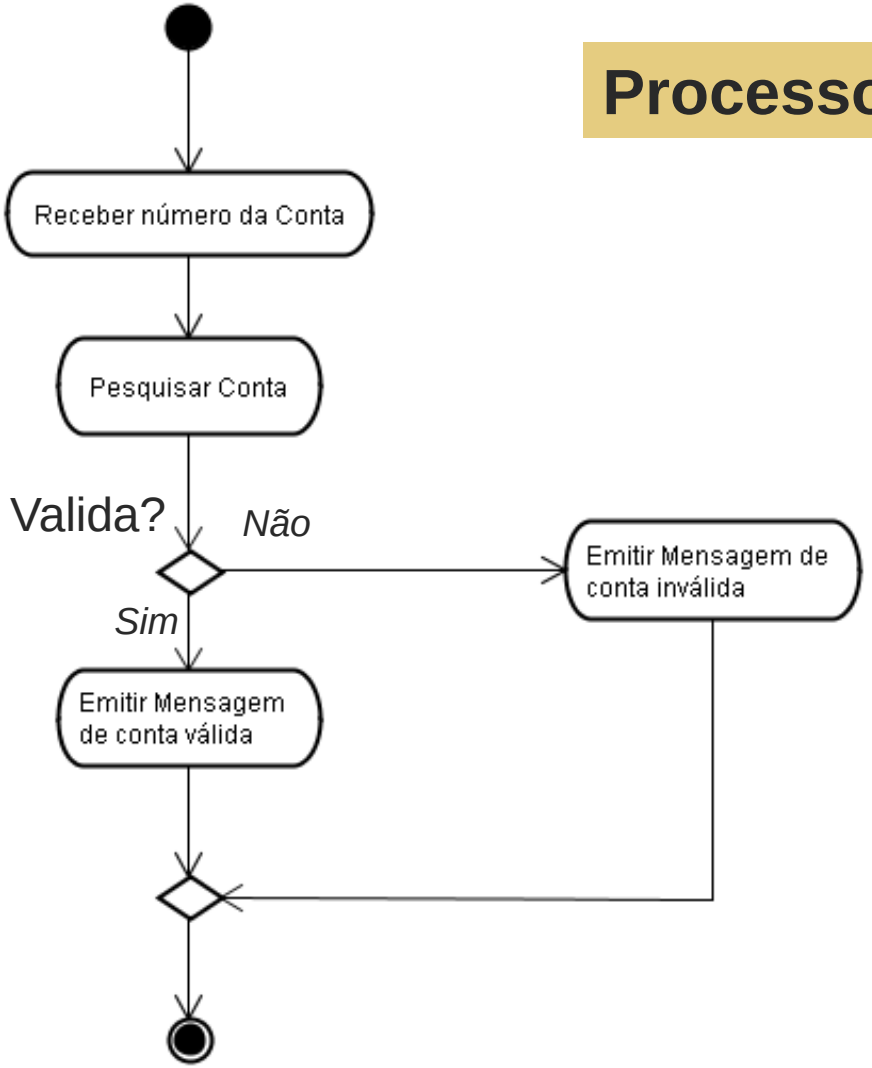


Diagrama de Casos de Uso

- Atores
  - Quem executa a funcionalidade
- Casos de Uso
  - Qual é a funcionalidade

Relacionamentos

- Principais tipos de relacionamentos
  - Associação
  - Inclusão
  - Extensão
  - Generalização
- Representam as interações
  - Atores e Casos de Uso
  - Dois ou mais Casos de Uso
  - Dois ou mais Atores

Cenários

- Descreve uma situação de uso do sistema
- Inclui informações como
  - Nome do Cenário
  - Ator
  - Pré-condição
  - Fluxo normal
  - Fluxos alternativos
  - Pós-condição

Bibliografia

- BOOCH, G., RUMBAUGH, J., JACOBSON, I. **UML, Guia do Usuário**. 2ª Ed., Editora Campus, 2005.
  - Prefácio
  - Capítulos 1 e 2
  - Capítulos 17 e 18
- M. Fowler. **UML Essencial**, 2a Edição. Bookmann, 2000.
  - Capítulo 3

Aula 09 - Exercício

1. Criar (um ou mais) Diagrama(s) de Casos de Uso
   - Usar o ArgoUML para criar os diagramas
   - https://argouml.en.softonic.com/
2. Documentar cada Caso de Uso utilizando a técnica de cenários
   - Usar o Word ou Google Docs (ou outro programa) para descrever os cenários

Aula 10 - Projeto Arquitetural

- O projeto é uma atividade criativa
  - Cada arquiteto tem sua própria maneira de projetar o software
- Projeto arquitetural faz a ligação entre
  - Requisitos (domínio do problema)
  - Projeto detalhado (domínio da solução)

Quando usar o diagrama?

- Pontos positivos

  - Facilita comunicação com *stakeholders*
  - Úteis para planejamento de projeto
  - Facilita o desenvolvimento em paralelo
- Pontos negativos

  - Não mostra a natureza dos relacionamentos entre componentes
  - Não indica as propriedades internas dos subsistemas

Padrões Arquiteturais

- Padrões arquiteturais expressam formas de organizar a estrutura fundamental do sistema
  - Permitem a construção de uma arquitetura aderente a certas propriedades
- O conhecimento de padrões arquiteturais ajuda na definição da arquitetura do sistema

Padrões Arquiteturais

- Da desordem a estrutura
  - **Layered Architecture (Arquitetura em Camadas)**
  - **Blackboard (Arquitetura de Repositório)**
  - **Pipes and Filters (Dutos e Filtros)**
- Sistemas distribuídos
  - **Client-Server (Cliente-Servidor)**
  - Broker
- Sistemas interativos
  - **Model-View-Controller (MVC)**
  - Presentation-Abstraction-Control
- Sistemas adaptáveis
  - Microkernel
  - Reflection

Discutidos no livro do Sommerville

Da Desordem a Estrutura

- Layered Architecture
  - Arquitetura em Camadas
- Blackboard
  - Arquitetura de Repositório
- Pipes and Filters
  - Dutos e Filtros

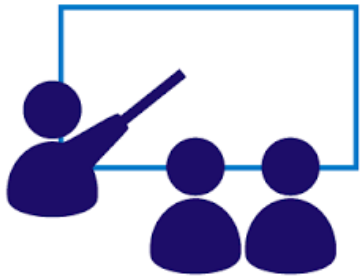
Arquitetura em Camadas

- Organiza o sistema em um conjunto de camadas
  - Cada camada oferece um conjunto de serviços

- Uma camada somente
  - Solicita serviços da camada inferior
  - Fornece serviços para a camada superior

Arquitetura de Repositório

- Também conhecido como *Blackboard*
- Os subsistemas manipulam a mesma base de dados
  - Um (ou mais) subsistema gera os dados
  - Vários subsistemas leem os dados
- Adotado principalmente quando dados são compartilhados em grandes quantidades



Dutos e Filtros

- Padrão de organização da dinâmica de um sistema
- Dois papéis principais
  - **Dutos**: componentes que conduzem ou distribuem os dados
  - **Filtros**: componentes que transformam os dados
- Usado principalmente em aplicações de processamento de dados

Bibliografia

- Ian Sommerville. **Engenharia de Software**, 9a. Edição. 2011.
  - Capítulo 6
- F. Buschmann et al. **Pattern-Oriented Software Architecture: A System of Patterns**. John Wiley & Sons, 1996.
  - Cap. 2 Architectural Patterns

Aula 11 - Prova 1

- O conteúdo da Prova 1 não será cobrado diretamente na Prova 2
  - Porém, pode ser cobrado indiretamente pois o conteúdo é sempre acumulativo

Aula 12 - Apresentações TP

- Os temas das apresentações podem ser usados para contexto de algumas questões.

Aula 13 - Desenvolvimento OO

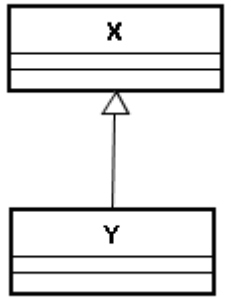
- Análise orientada a objetos
  - Cria um modelo de objetos para o domínio da aplicação (domínio do problema)
- Modelagem orientada a objetos
  - Cria um modelo de objetos para implementar requisitos (domínio da solução)
- Programação orientada a objetos
  - Implementa o projeto orientado a objetos usando uma linguagem de programação

Atividades de Modelagem OO

1. Definir o contexto do sistema
2. Projetar a arquitetura
3. Identificar os objetos principais
4. Desenvolver os modelos de projeto
5. Especificar interfaces entre objetos

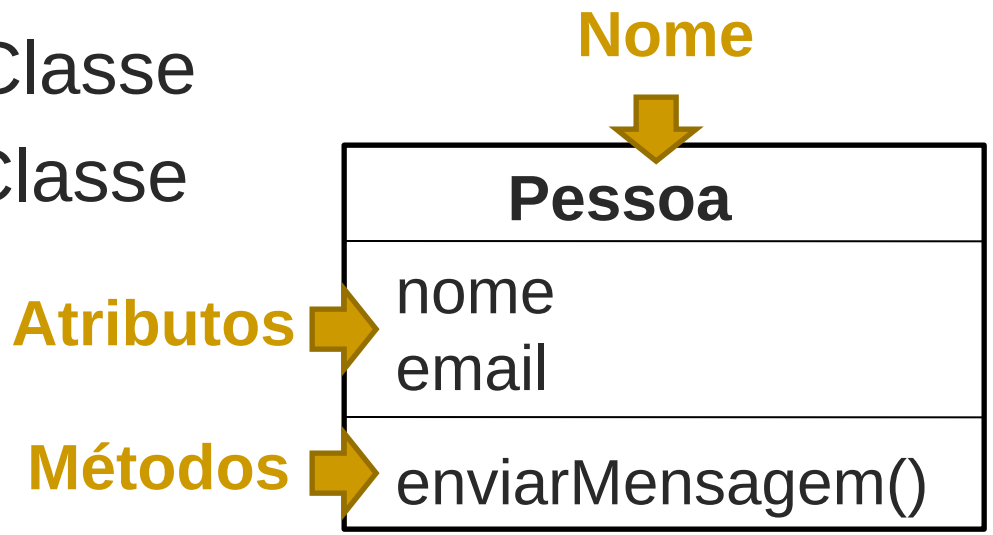
Diagrama de Classes

- Serve de apoio para a maioria dos outros diagramas
- Define a estrutura das classes do sistema
  - Apresenta uma visão estática de como as classes estão organizadas
- Estabelece como as classes se relacionam



Representação de uma Classe

- Uma classe é representada por um retângulo com três divisões:
  - Nome da Classe
  - Atributos da Classe
  - Métodos da Classe



Tipos de Visibilidade

- Pública (+)
  - O atributo ou método pode ser utilizado por qualquer classe
- Protegida (#)
  - Somente a classe, sub-classes e classes amigas (ex. pacote) têm acesso
- Privada (-)
  - Somente a própria classe terá acesso

Relacionamentos

- Classes possuem relacionamentos entre elas (para comunicação)
  - Compartilham informações
  - Colaboram umas com as outras
- Principais tipos de relacionamentos
  - Associação
  - Agregação / Composição
  - Herança
  - Dependência

Multiplicidade

0..1	No máximo um. Indica que os objetos da classe associada não precisam obrigatoriamente estar relacionados.
1..1	Um e somente um. Indica que apenas um objeto da classe se relaciona com os objetos da outra classe.
0..*	Muitos. Indica que podem haver muitos objetos da classe envolvidos no relacionamento
1..*	Um ou muitos. Indica que há pelo menos um objeto envolvido no relacionamento.
3..5	Valores específicos.

Q9 - Métodos Herdados

Superclasse

```
public abstract class Airplane {
  protected String model = "";
  public void fly() {
    System.out.println(model + " preparing to fly.");
    run();
    takeOff();
  }
  private void run() {
    System.out.println(model + " running.");
  }
  abstract void takeOff();
}
```

```
public class Airbus320 extends Airplane {
  public Airbus320() {
    this.model = "Airbus A320";
  }
  void takeOff() {
    System.out.println(model + " taking off.");
  }
}
```

Subclasse

Q9 - Métodos Herdados

```
public class Airbus320 extends Airplane {
  public Airbus320() {
    this.model = "Airbus A320";
  }
  void takeOff() {
    System.out.println(model + " taking off.");
  }
}
```

Subclasse

```
public class TestAirbus320 {
  public static void main(String[] args) {
    Airbus320 a320 = new Airbus320();
    a320.fly();
  }
}
```

- Saída (comportamento de Airbus320
  - Airbus A320 preparing to fly.
  - Airbus A320 running.
  - Airbus A320 taking off.

Bibliografia

- Ian Sommerville. **Engenharia de Software**, 9a. Edição. 2011. (Capítulo 7)
- Deitel, H. M.; Deitel P. J. **Java: Como Programar**, 8a. Edição. Pearson, 2010.
  - Capítulos 1 e 3
- G. Booch, J. Rumbaugh, I. Jacobson. **UML, Guia do Usuário**. 2ª Ed., Editora Campus, 2005. (Capítulos 4, 8, 9 e 10)
- M. Fowler. **UML Essencial**, 2a Edição. Bookmann, 2000. (Capítulos 4 e 6)

Aula 14 - Exercícios

1. Criar um (ou mais) diagrama(s) de classes
   - Minha sugestão é usar o ArgoUML

- Os diagramas devem conter o máximo de detalhes possíveis para facilitar a implementação
  - Classes e interfaces, métodos e atributos, visibilidade, relacionamentos, multiplicidade, notas, etc.

Aula 15 - Outros Diagramas

- Diagrama de Sequência
- Diagrama de Comunicação
- Diagrama de Atividades

Diagrama de Sequência

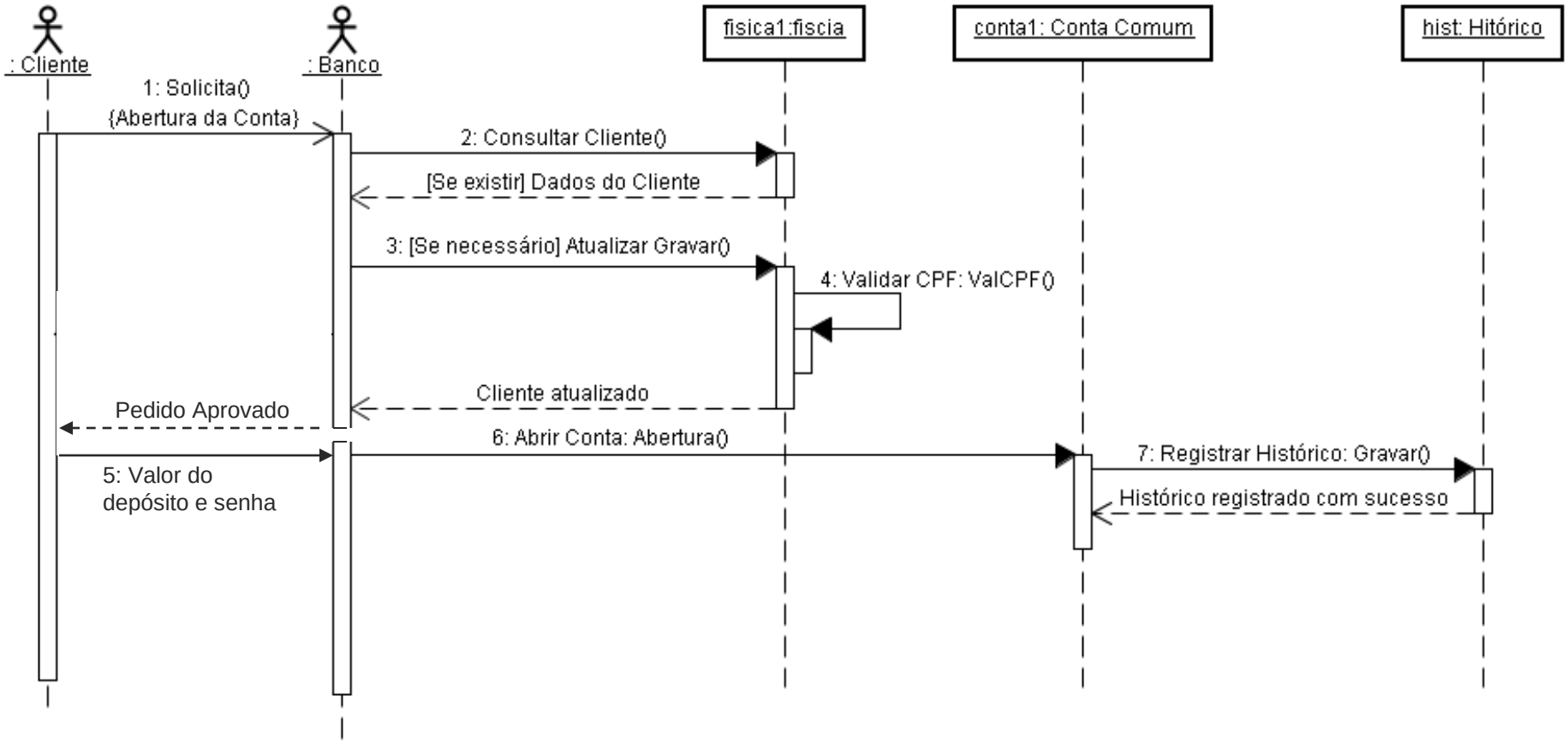
- Preocupa-se com a ordem temporal em que as mensagens são trocadas
- Pode ser usado para detalhar um Caso de Uso
- Identifica
  - O evento gerador da funcionalidade modelada (ator responsável pelo evento)
  - Os objetos envolvidos na ação

Mensagens do Diagrama

- Representam a comunicação entre objetos e/ou atores do Diagrama de Sequência

- Exemplos de mensagens
  - Chamadas de um método de um objeto por outro objeto
  - Comunicação entre dois atores

Diagrama de Sequência



Detalhe de um Caso de Uso

- Caso de Uso é um processo disparado pelo usuário
- O Diagrama de Sequência pode detalhar um caso de uso e mostrar
  - a ordem em que os eventos acontecem
  - as mensagens que são enviadas
  - os métodos que são chamados
  - como os objetos interagem entre si

Diagrama de Comunicação

- Também chamado de Diagrama de Colaboração
- Define a estrutura de como os objetos estão vinculados
  - Semelhante ao Diagrama de Classes
- Indica quais mensagens são trocadas entre objetos
  - Semelhante ao Diagrama de Sequência

Diagrama de Comunicação

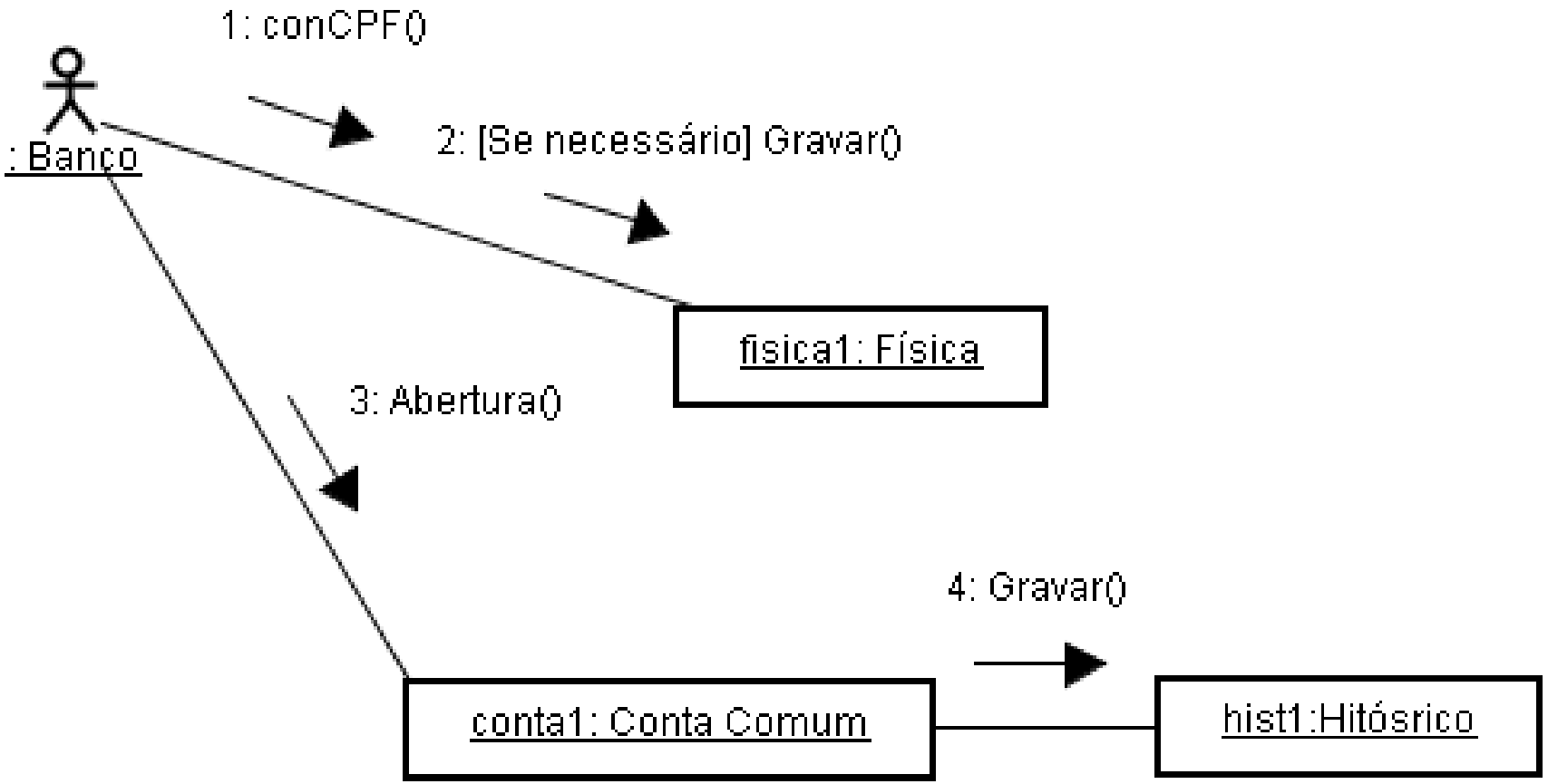
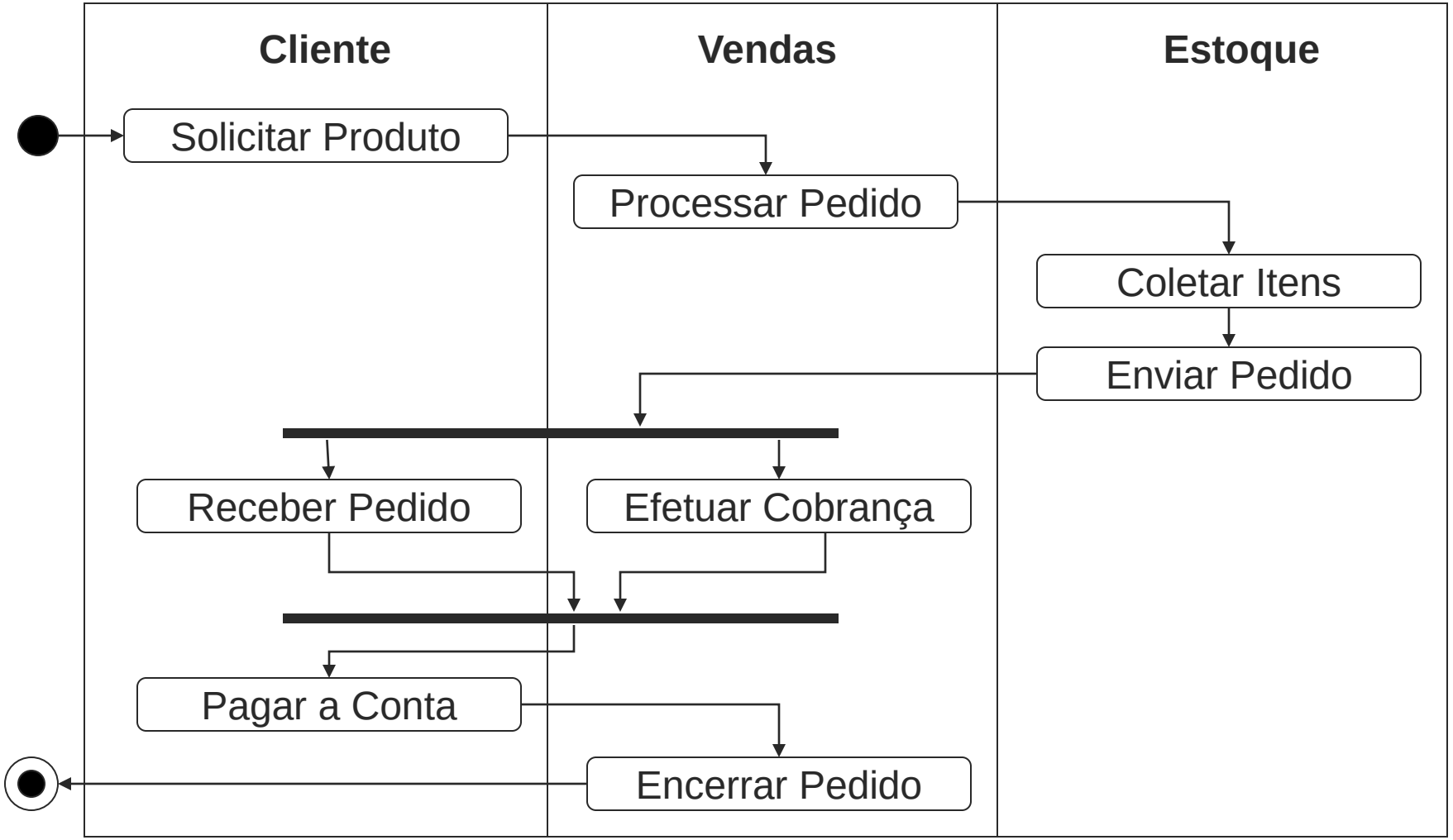


Diagrama de Atividades

- Mostram as atividades que compõem um processo do sistema e o fluxo de controle
  - Decompõe um processo em suas atividades
- Modelam a execução de atividades sequenciais ou concorrentes

Diagrama de Atividades



Bibliografia

- G. Booch, J. Rumbaugh, I. Jacobson. **UML, Guia do Usuário**. 2ª Ed., Editora Campus, 2005.
  - Capítulos 16, 19 e 20
- M. Fowler. **UML Essencial**, 2a Edição. Bookmann, 2000.
  - Capítulos 5 e 9