

# UML – Linguagem de Modelagem Unificada

Prof. Antonio Guardado

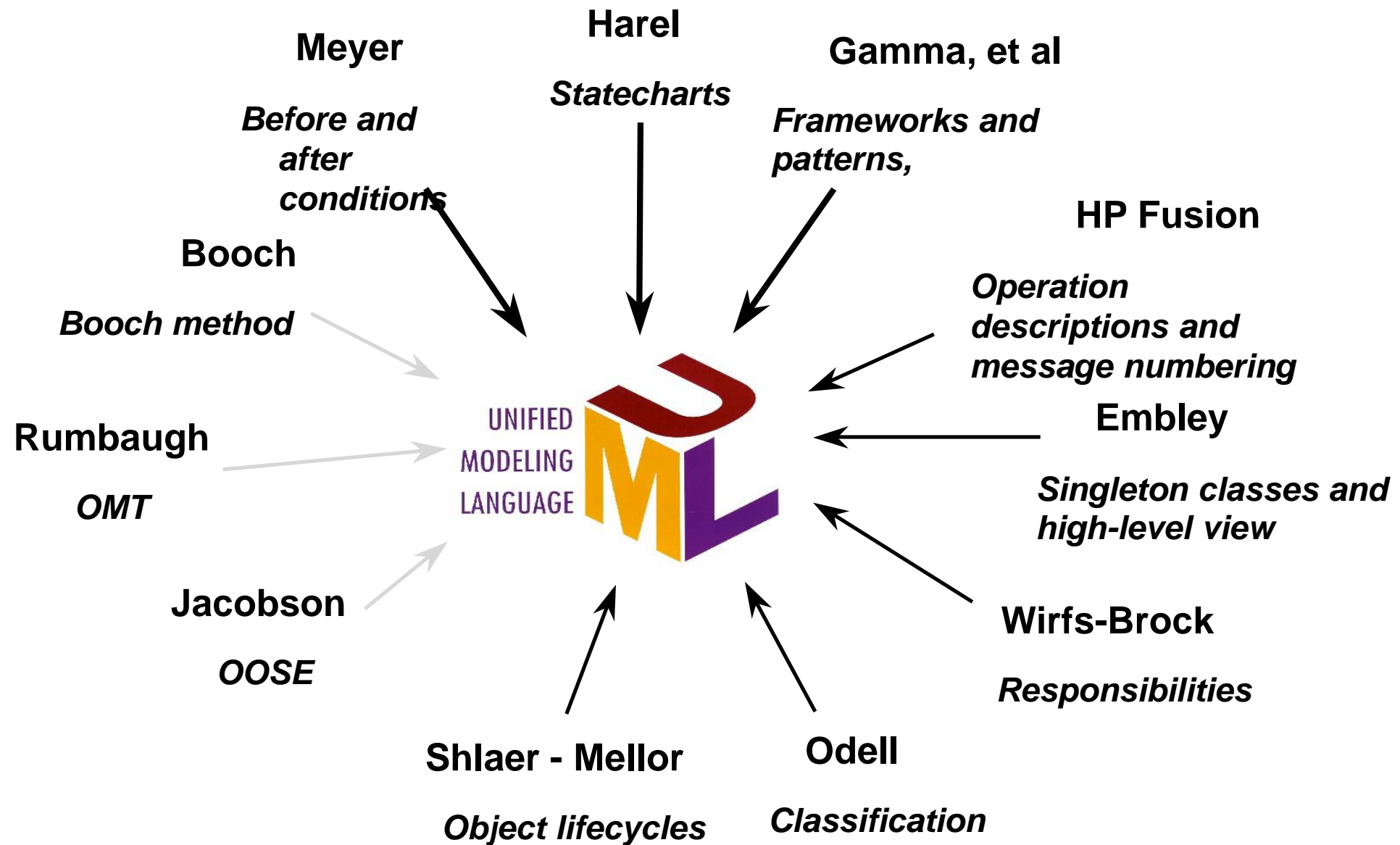
(baseado no livro UML 2 – Uma abordagem prática –  
Gileanes Guedes)

# 1- Linguagem de Modelagem Unificada

- A UML é uma linguagem para
  - visualização
  - especificação
  - construção
  - documentação
- de artefatos de um sistema com uma componente intensiva de software (*software intensive system*)
- UML não é uma metodologia
  - não diz quem deve fazer o quê, quando e como
  - UML pode ser usado segundo diferentes metodologias, tais como RUP (*Rational Unified Process*), FDD (*Feature Driven Development*), etc.
- UML não é uma linguagem de programação



# 1.1 - Unificação de métodos e formalismos



## 1.2 - Vantagens da UML

---

- É um padrão aberto
  - versão 1.1 aprovada pelo OMG (Object Management Group) em Novembro de 1997
- Suporta todo o ciclo de vida do software
  - modelagem do negócio (processos e objetos do negócio)
  - modelagem de requisitos alocados ao software
  - modelagem da solução de software
- Suporta diversas áreas de aplicação
- É baseada na experiência e necessidades da comunidade de usuários
- É suportada por muitas ferramentas

## 1.3 - Modelos, Diagramas, Visões e Elementos (1)

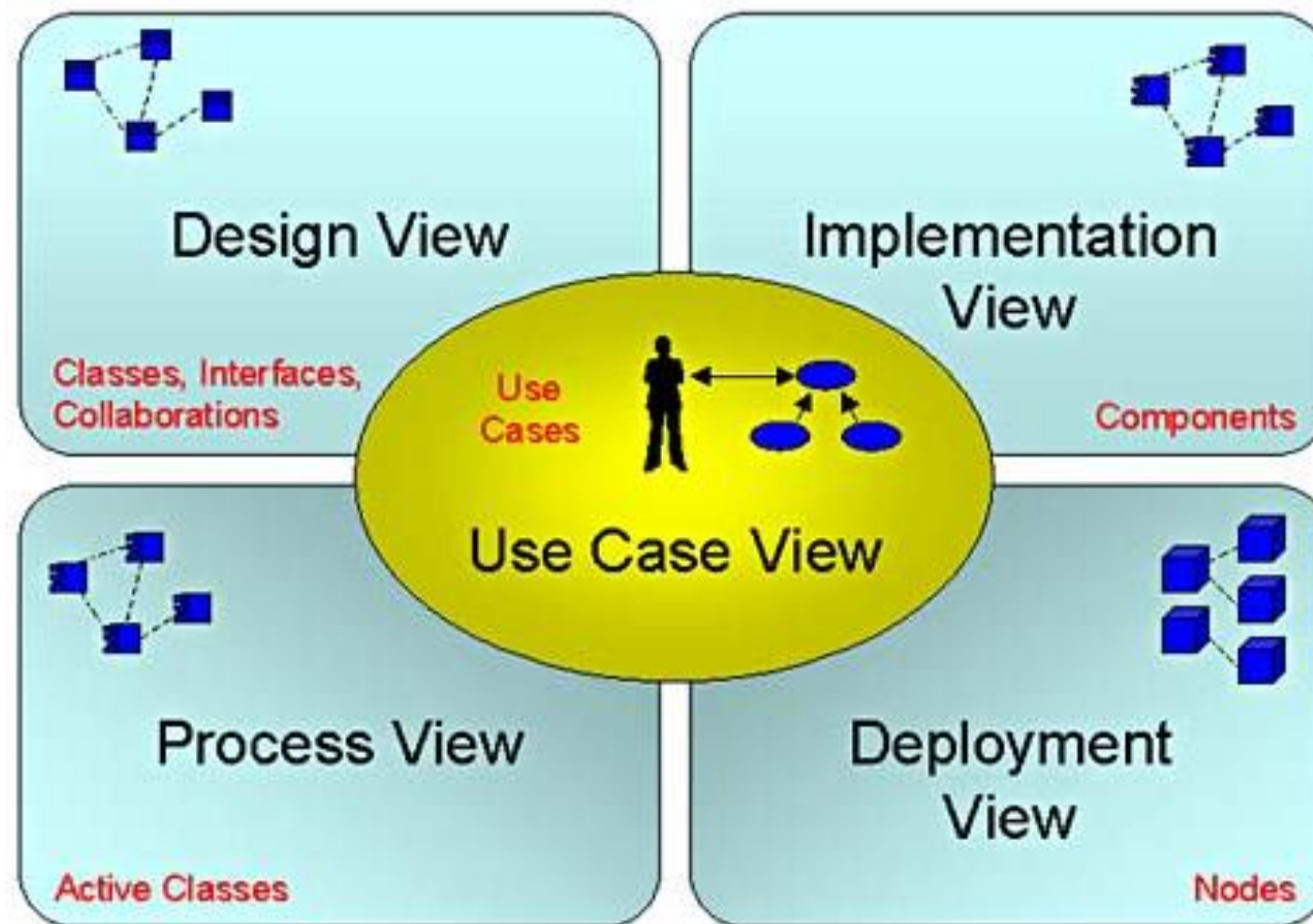
- Um modelo é uma representação em **pequena escala**, numa perspectiva particular, de um sistema existente ou a criar
- Ao longo do ciclo de vida de um sistema são construídos vários modelos, sucessivamente refinados e enriquecidos
- Um modelo é constituído por um conjunto de **diagramas** (desenhos) consistentes entre si, acompanhados de descrições textuais dos **elementos** que aparecem nos vários diagramas

## 1.3 - Modelos, Diagramas, Visões e Elementos (2)

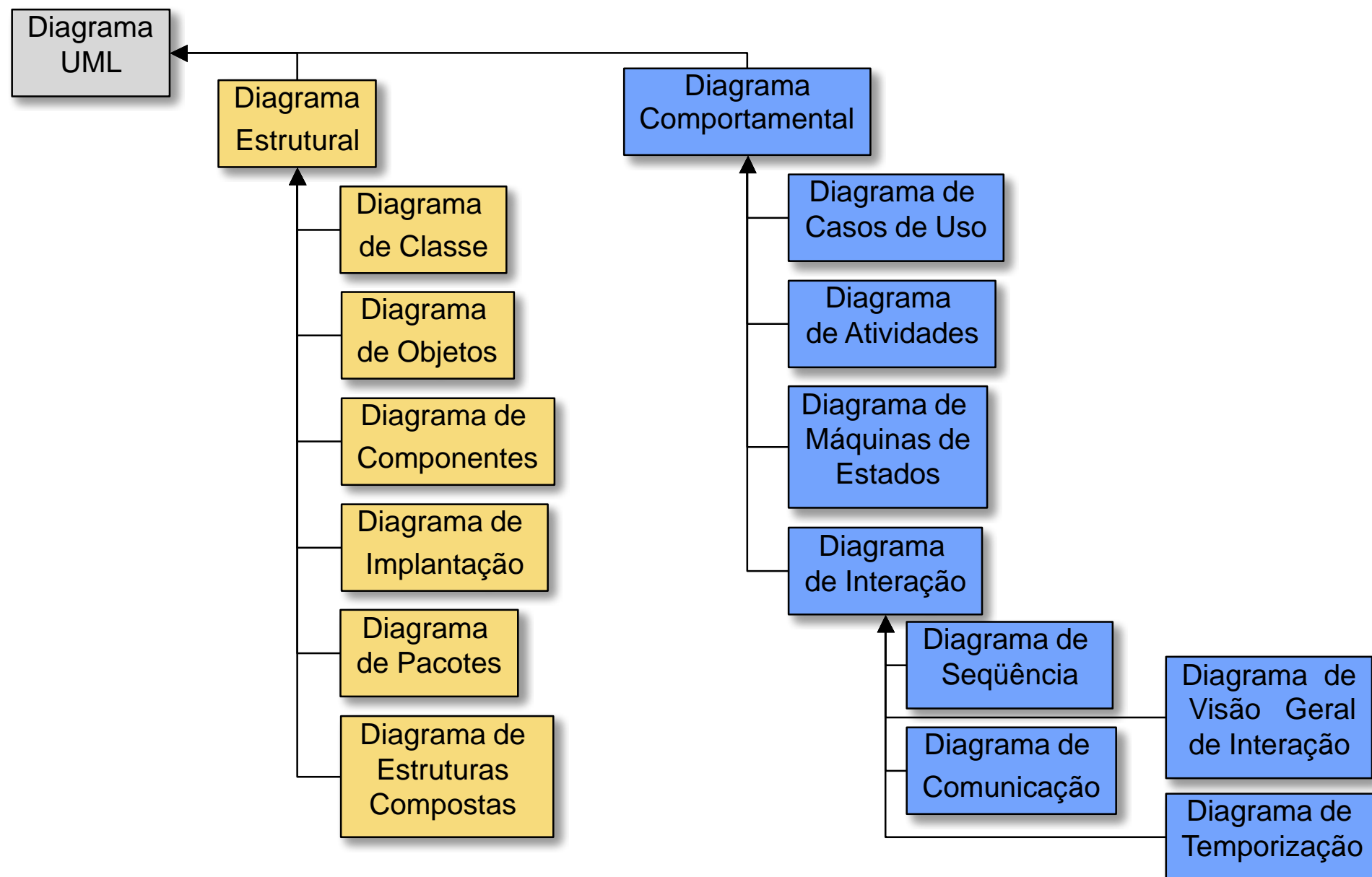
- Um diagrama é uma visão sobre um modelo
  - De acordo com o interesse de uma das partes envolvidas (stakeholder)
  - Proporciona uma representação parcial do sistema
  - Deve ser semanticamente consistente com outras visões
- Na UML 1.5, há 9 diagramas padrões e 3 de organização
  - Diagramas de visão estática: casos de uso (use case), classes, objetos, componentes, implantação (deployment)
  - Diagramas de visão dinâmica: sequência, colaboração, estados (statechart), actividades
  - Diagramas de organização: pacotes, subsistemas e modelos
- O mesmo elemento (exemplo: classe) pode aparecer em vários diagramas de um modelo



## 1.4 - Diferentes Visões



# 1.5 - Diagramas na UML 2.X





## 2- Diagrama de Casos de Uso

O diagrama de casos de uso é o diagrama mais geral e informal da UML, utilizado normalmente nas fases de levantamento e análise de requisitos do sistema, embora venha a ser consultado durante todo o processo de modelagem e possa servir de base para outros diagramas. Apresenta uma linguagem simples e de fácil compreensão para que os usuários possam ter uma ideia geral de como o sistema irá se comportar. Procura identificar os atores (usuários, outros sistemas ou até mesmo algum hardware especial) que utilizarão de alguma forma o software, bem como os serviços, ou seja, as funcionalidades que o sistema disponibilizará aos atores, conhecidas nesse diagrama como casos de uso. A figura 1 é um exemplo desse diagrama.

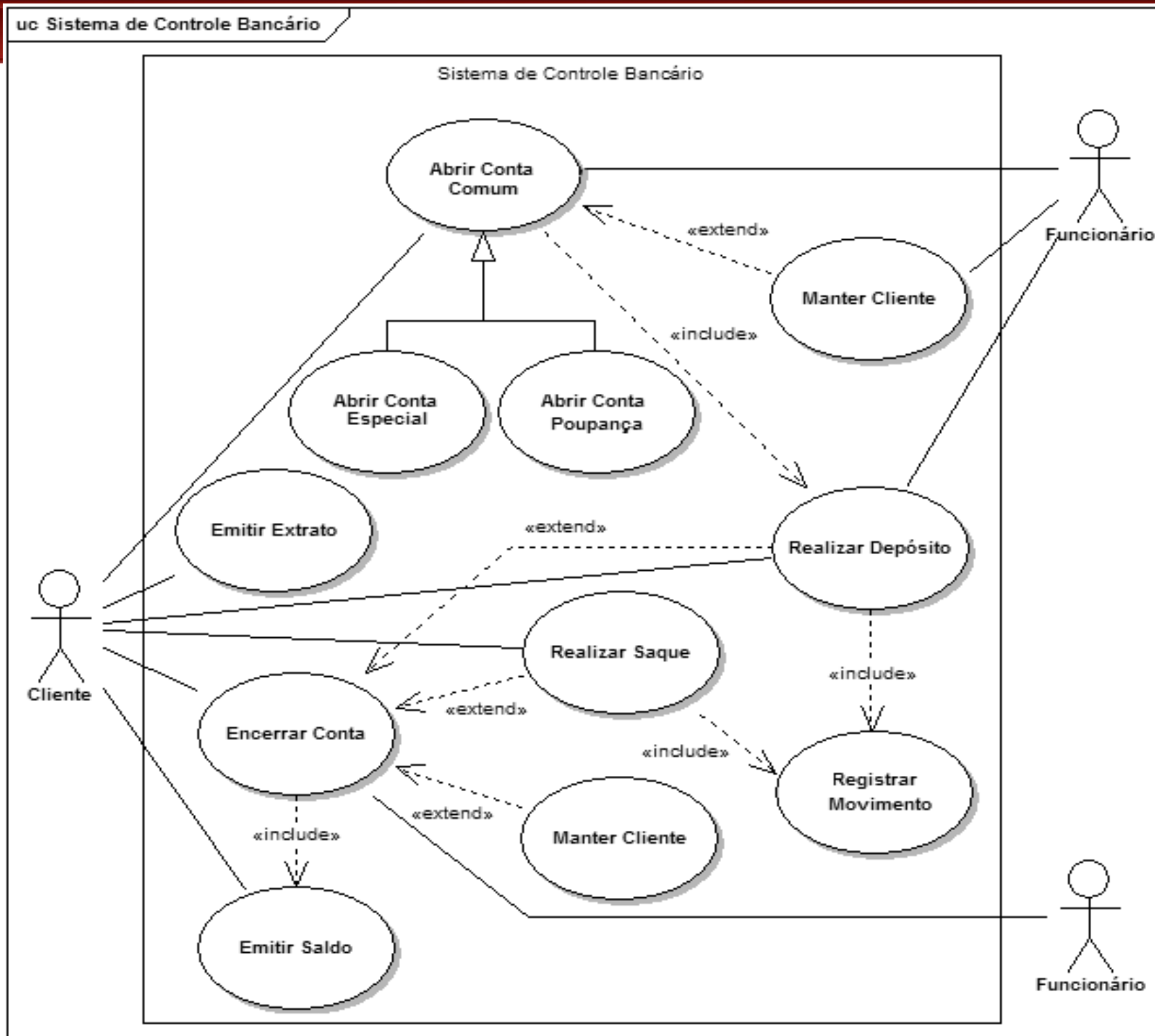


Figura 1– Exemplo de Diagrama de Casos de Uso.

### 3- Diagrama de Classes

O diagrama de classes é provavelmente o mais utilizado e é um dos mais importantes da UML. Serve de apoio para a maioria dos demais diagramas. Como o próprio nome diz, define a estrutura das classes utilizadas pelo sistema, determinando os atributos e métodos que cada classe tem, além de estabelecer como as classes se relacionam e trocam informações entre si. A figura 2 apresenta um exemplo desse diagrama.

class Modelo de Domínio - Sistema de Controle Bancário

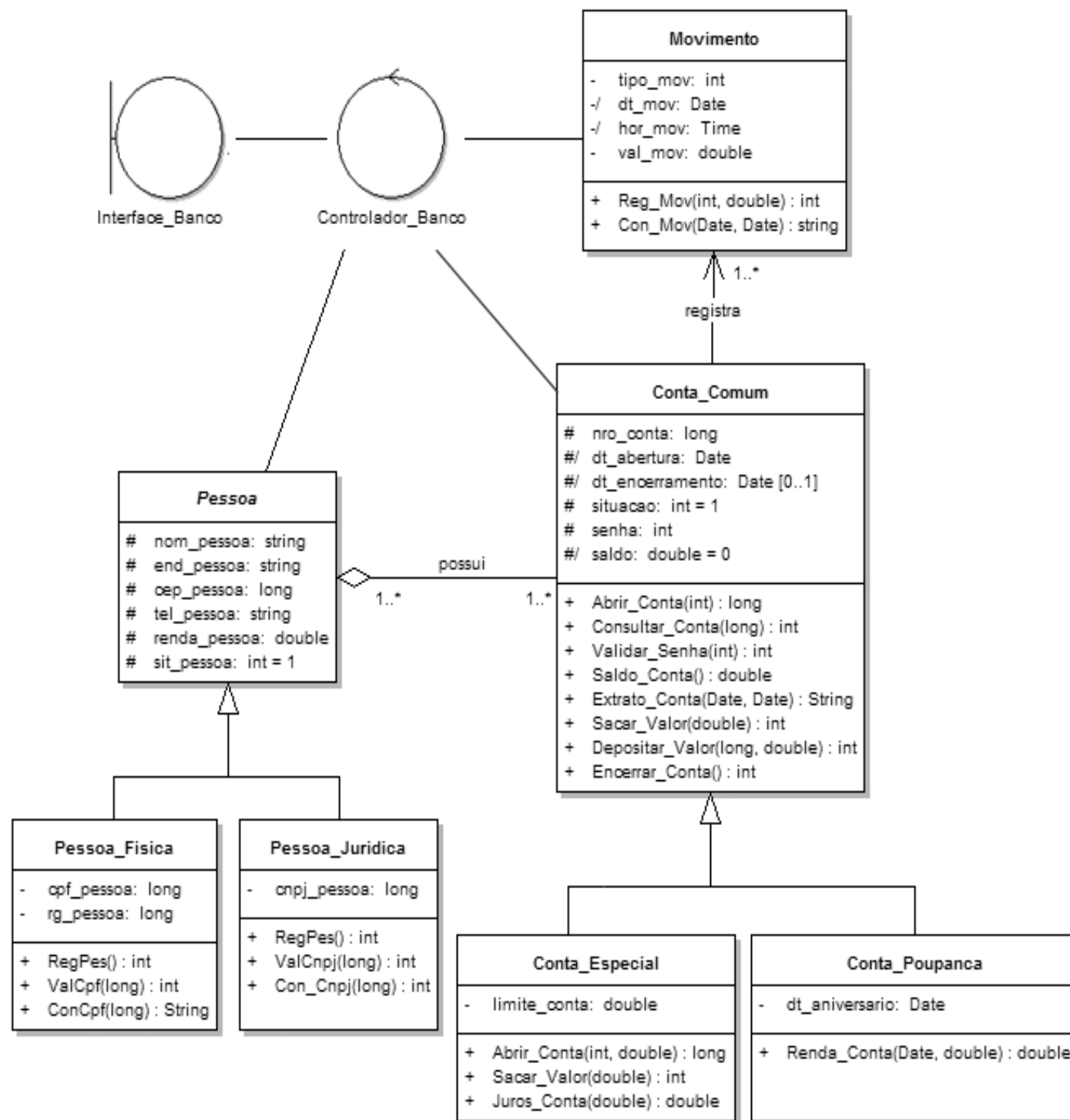
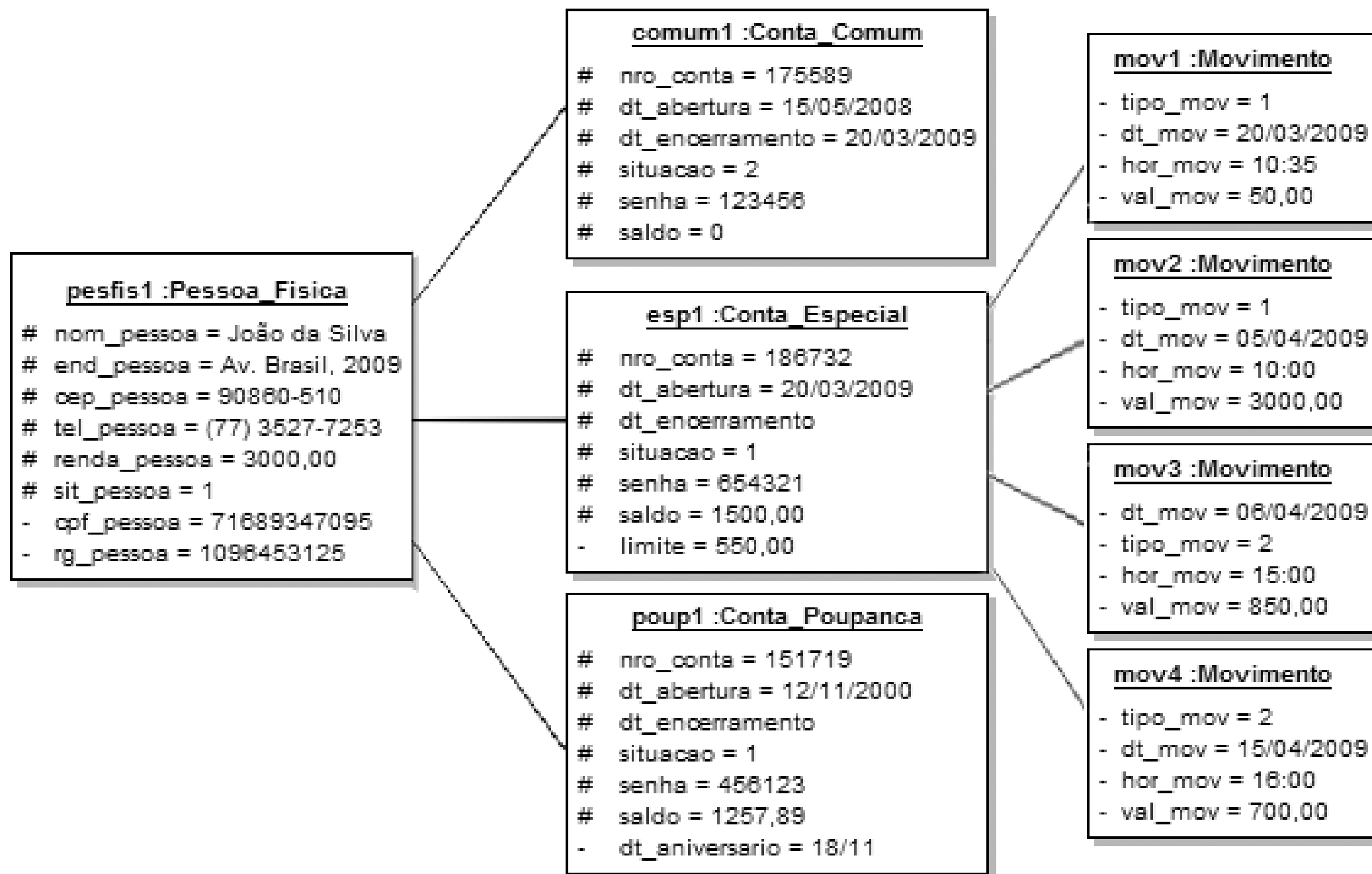


Figura 2— Exemplo de Diagrama de Classes.

## 4 - Diagrama de Objetos

O diagrama de objetos está amplamente associado ao diagrama de classes. Na verdade, o diagrama de objetos é praticamente um complemento do diagrama de classes e bastante dependente deste. O diagrama fornece uma visão dos valores armazenados pelos objetos de um diagrama de classes em um determinado momento da execução de um processo do software. A figura 3 apresenta um exemplo de diagrama de objetos.

## object Diagrama de Objetos - Sistema de Controle Bancário

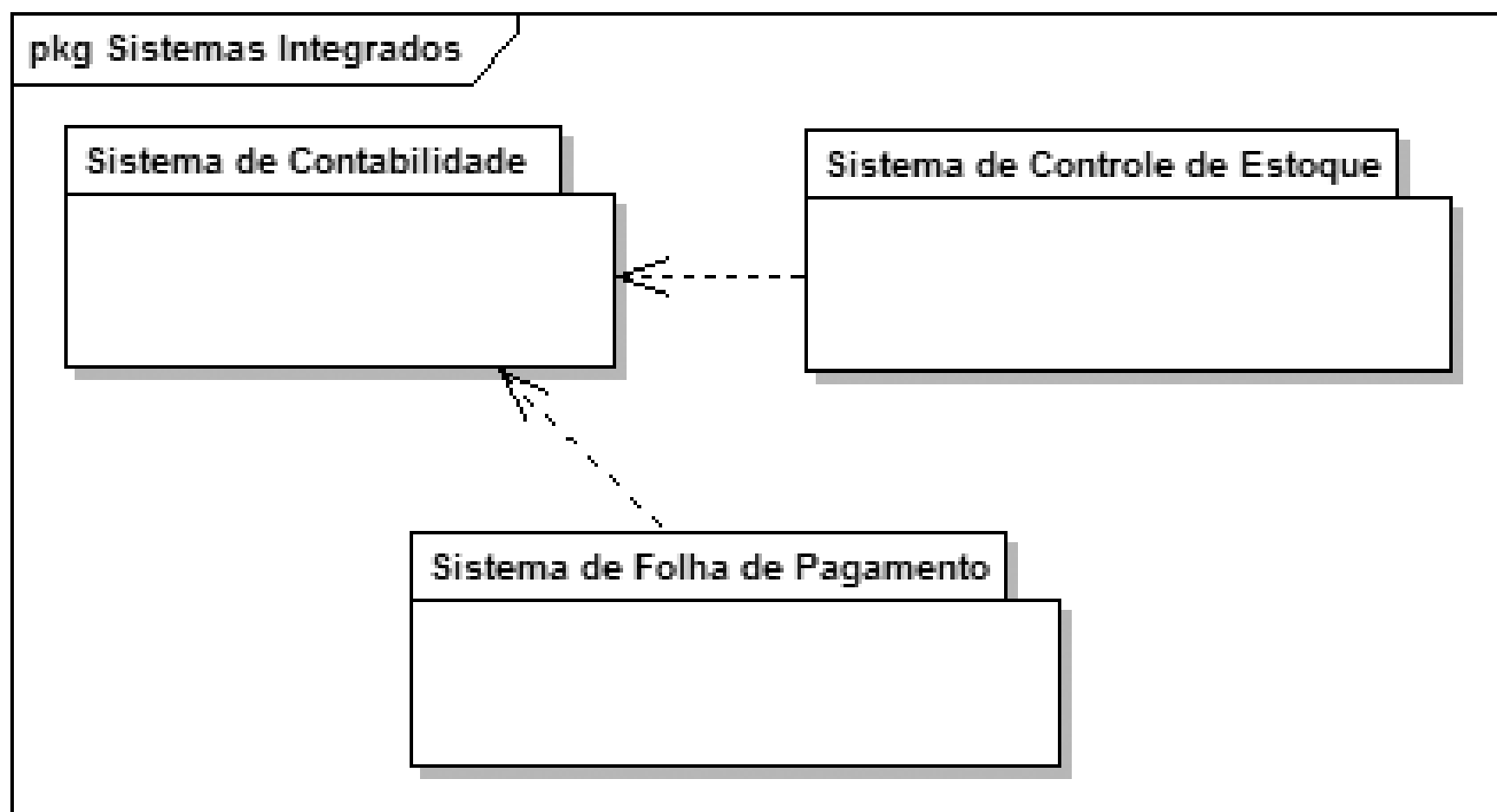




## 5- Diagrama de Pacotes

O diagrama de pacotes é um diagrama estrutural que tem por objetivo representar os subsistemas ou submódulos englobados por um sistema de forma a determinar as partes que o compõem. Pode ser utilizado de maneira independente ou associado com outros diagramas. Esse diagrama pode ser utilizado também para auxiliar a demonstrar a arquitetura de uma linguagem, como ocorre com a própria UML ou ainda para definir as camadas de um software ou de um processo de desenvolvimento. A figura 4 apresenta um exemplo do mesmo.

## 5- Diagrama de Pacotes (2)



*Figura 4 – Exemplo de Diagrama de Pacotes.*

## 6 - Diagrama de Sequência

O diagrama de sequência é um diagrama comportamental que preocupa-se com a ordem temporal em que as mensagens são trocadas entre os objetos envolvidos em um determinado processo. Em geral, baseia-se em um caso de uso definido pelo diagrama de mesmo nome e apoia-se no diagrama de classes para determinar os objetos das classes envolvidas em um processo. Um diagrama de sequência costuma identificar o evento gerador do processo modelado, bem como o ator responsável por esse evento, e determina como o processo deve se desenrolar e ser concluído por meio da chamada de métodos disparados por mensagens enviadas entre os objetos. A figura 5 apresenta um exemplo desse diagrama.

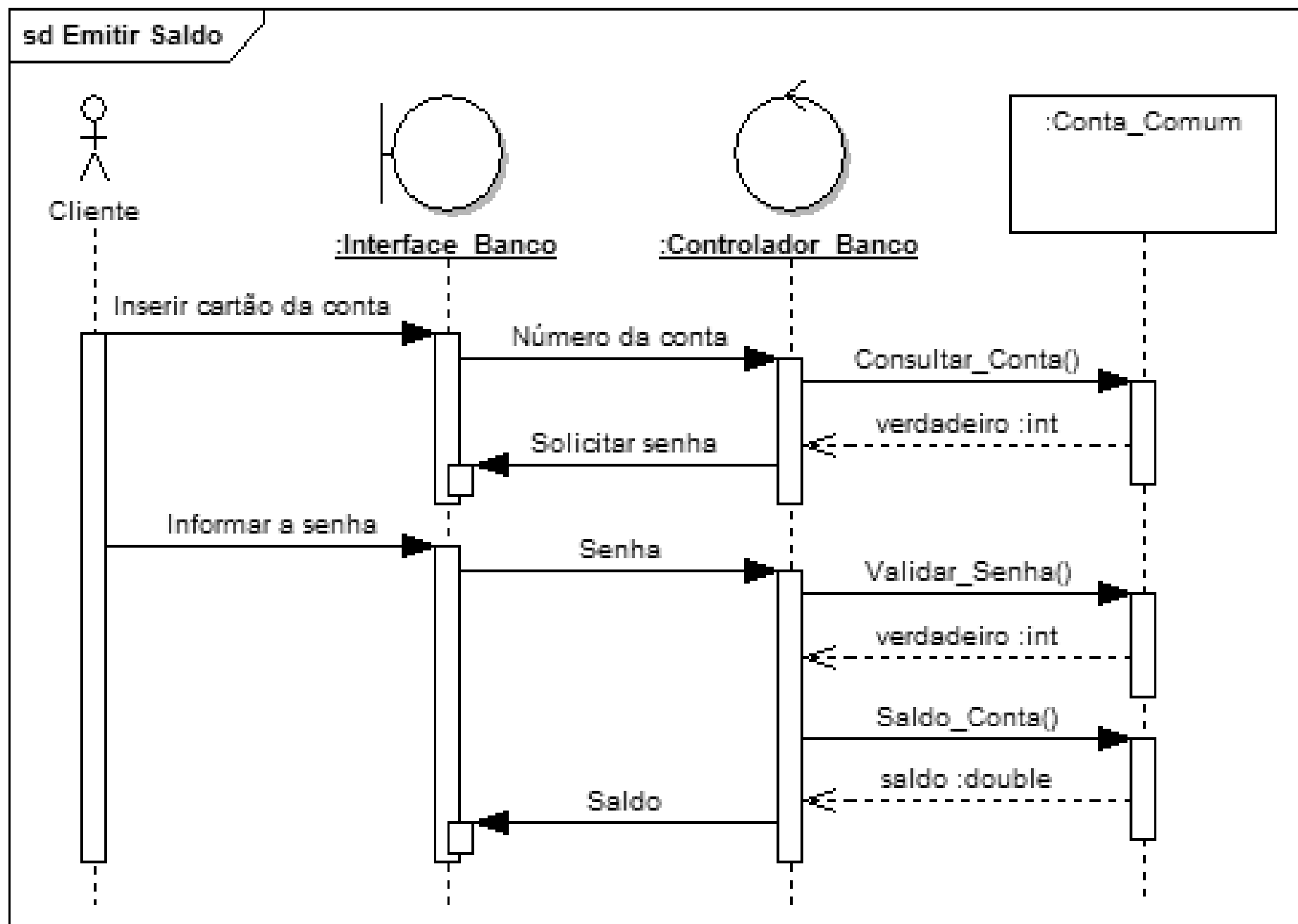


Figura 5 – Exemplo de Diagrama de Sequência.

## 7 - Diagrama de Comunicação

O diagrama de comunicação era conhecido como de colaboração até a versão 1.5 da UML, tendo seu nome modificado para diagrama de comunicação a partir da versão 2.0. Está amplamente associado ao diagrama de sequência: na verdade, um complementa o outro. As informações mostradas no diagrama de comunicação com frequência são praticamente as mesmas apresentadas no de sequência, porém com um enfoque distinto, visto que esse diagrama não se preocupa com a temporalidade do processo, concentrando-se em como os elementos do diagrama estão vinculados e quais mensagens trocam entre si durante o processo. A figura 6 apresenta um exemplo de diagrama de comunicação.

## 7 - Diagrama de Comunicação (2)

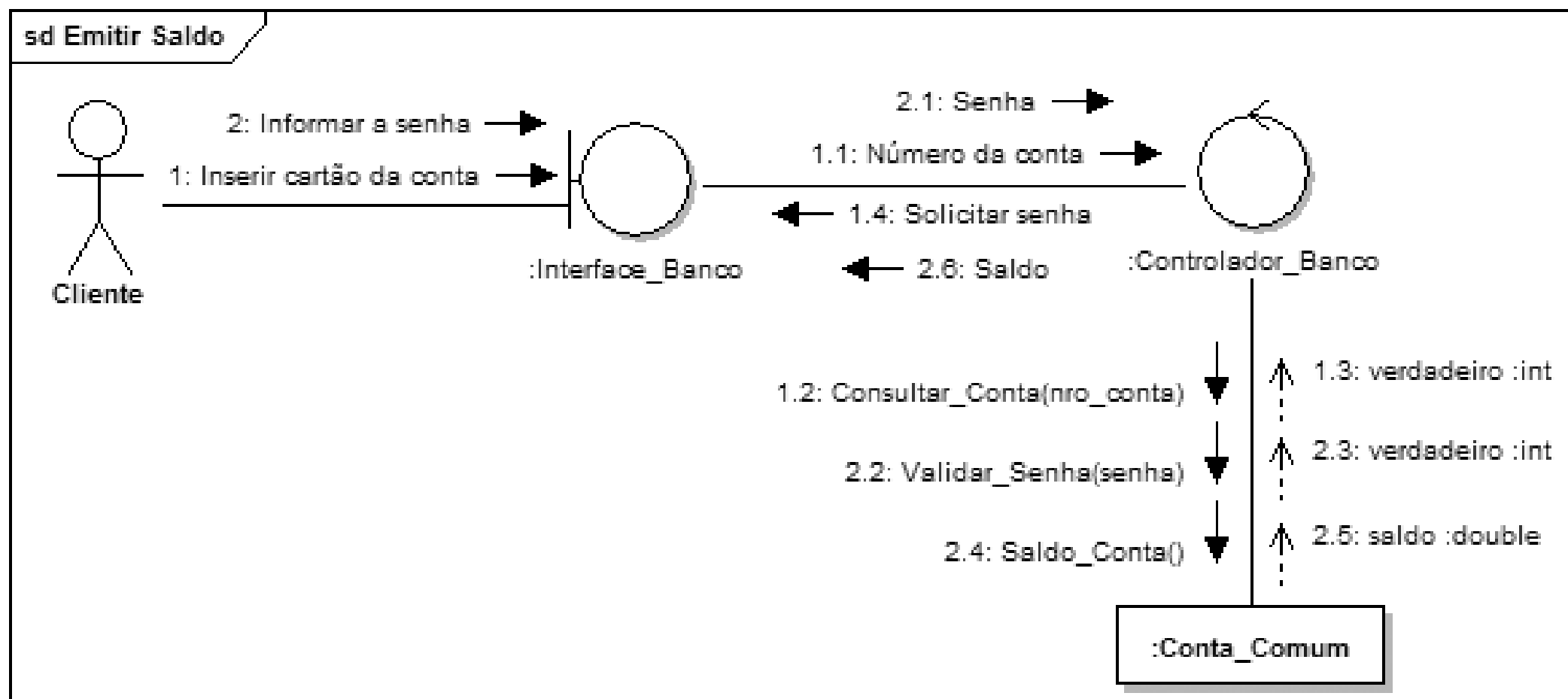


Figura 6 – Exemplo de Diagrama de Comunicação.



## 8- Diagrama de Máquina de Estados

O diagrama de máquina de estados demonstra o comportamento de um elemento por meio de um conjunto finito de transições de estado, ou seja, uma máquina de estados. Além de poder ser utilizado para expressar o comportamento de uma parte do sistema, quando é chamado de máquina de estado comportamental, também pode ser usado para expressar o protocolo de uso de parte de um sistema, quando identifica uma máquina de estado de protocolo.

Como o diagrama de sequência, o de máquina de estados pode basear-se em um caso de uso, mas também pode ser utilizado para acompanhar os estados de outros elementos, como, por exemplo, uma instância de uma classe. A figura 7 apresenta um exemplo de diagrama de máquina de estados.

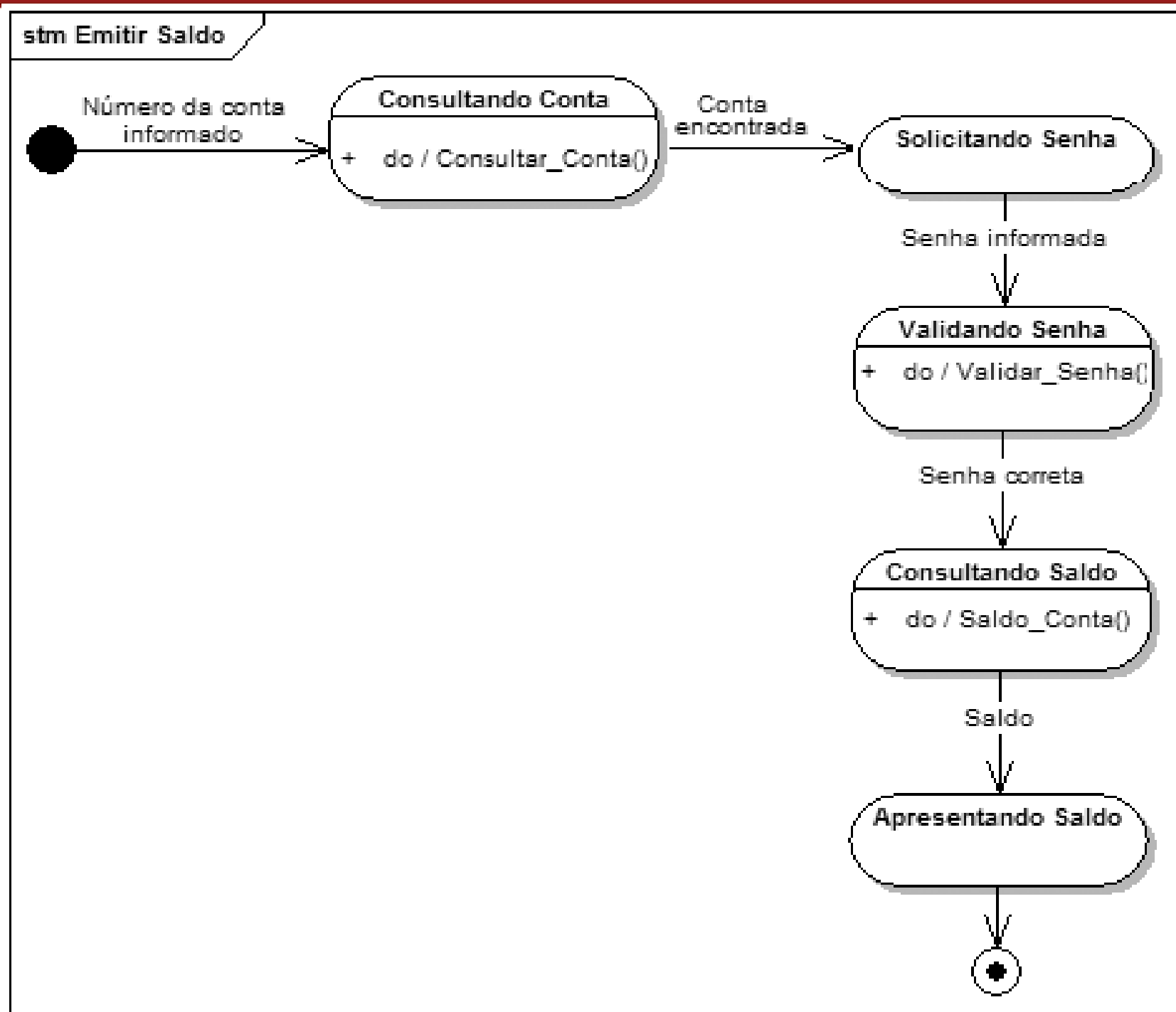
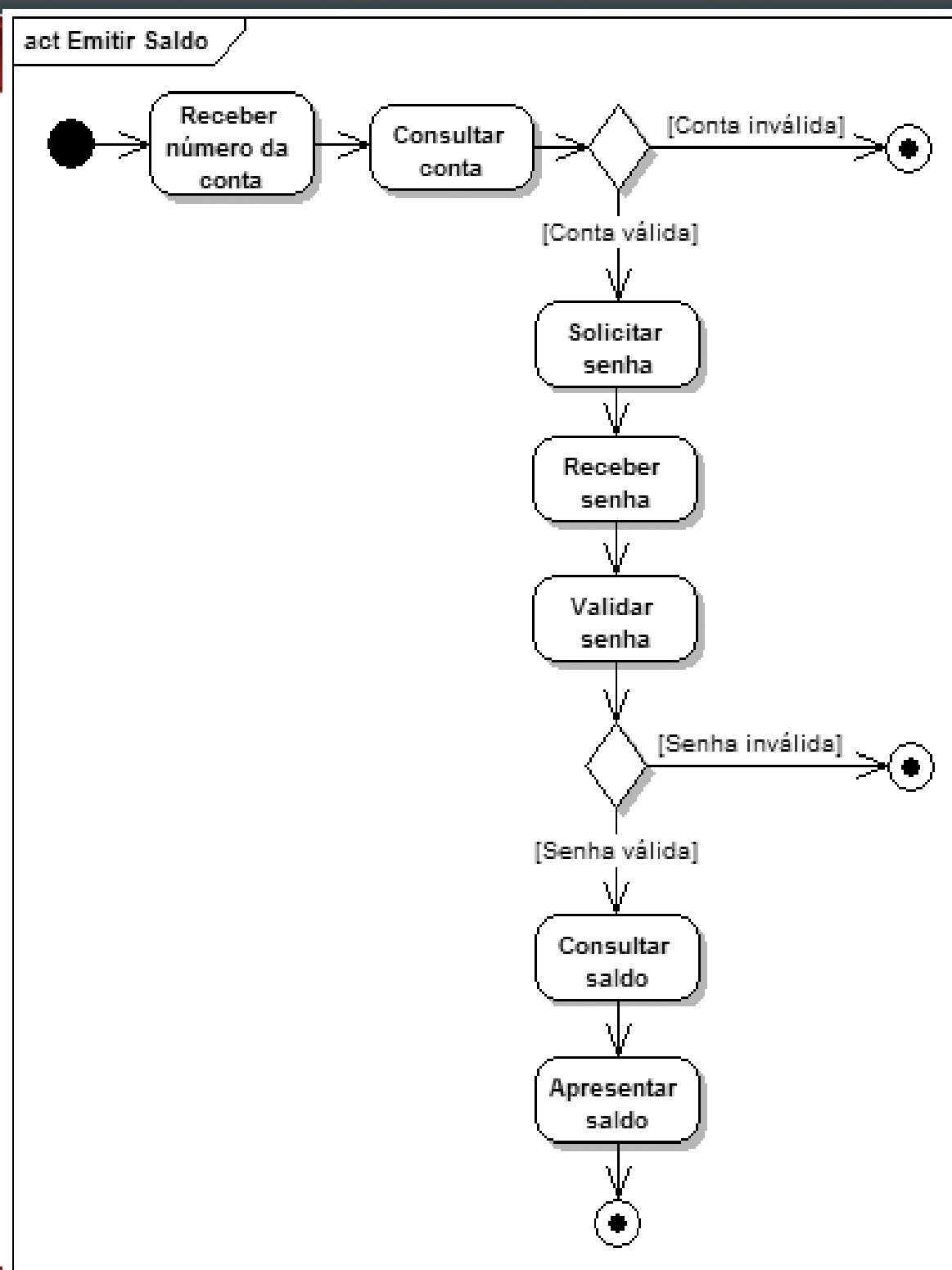


Figura 7 – Exemplo  
de Diagrama de  
Gráfico de Estados.

## 9- Diagrama de Atividade

O diagrama de atividade era considerado um caso especial do antigo diagrama de gráfico de estados, hoje conhecido como diagrama de máquina de estados, conforme descrito na seção anterior. A partir da UML 2.0, foi considerado independente do diagrama de máquina de estados. O diagrama de atividade preocupa-se em descrever os passos a serem percorridos para a conclusão de uma atividade específica, podendo esta ser representada por um método com certo grau de complexidade, um algoritmo, ou mesmo por um processo completo. O diagrama de atividade concentra-se na representação do fluxo de controle de uma atividade. A figura 8 apresenta um exemplo desse diagrama.



*Figura 8 – Exemplo de Diagrama de Atividade.*

## 10- Diagrama de Visão Geral de Interação

O diagrama de visão geral de interação é uma variação do diagrama de atividade que fornece uma visão geral dentro de um sistema ou processo de negócio. Esse diagrama passou a existir apenas a partir da UML 2. A figura 9 apresenta um exemplo do diagrama em questão.

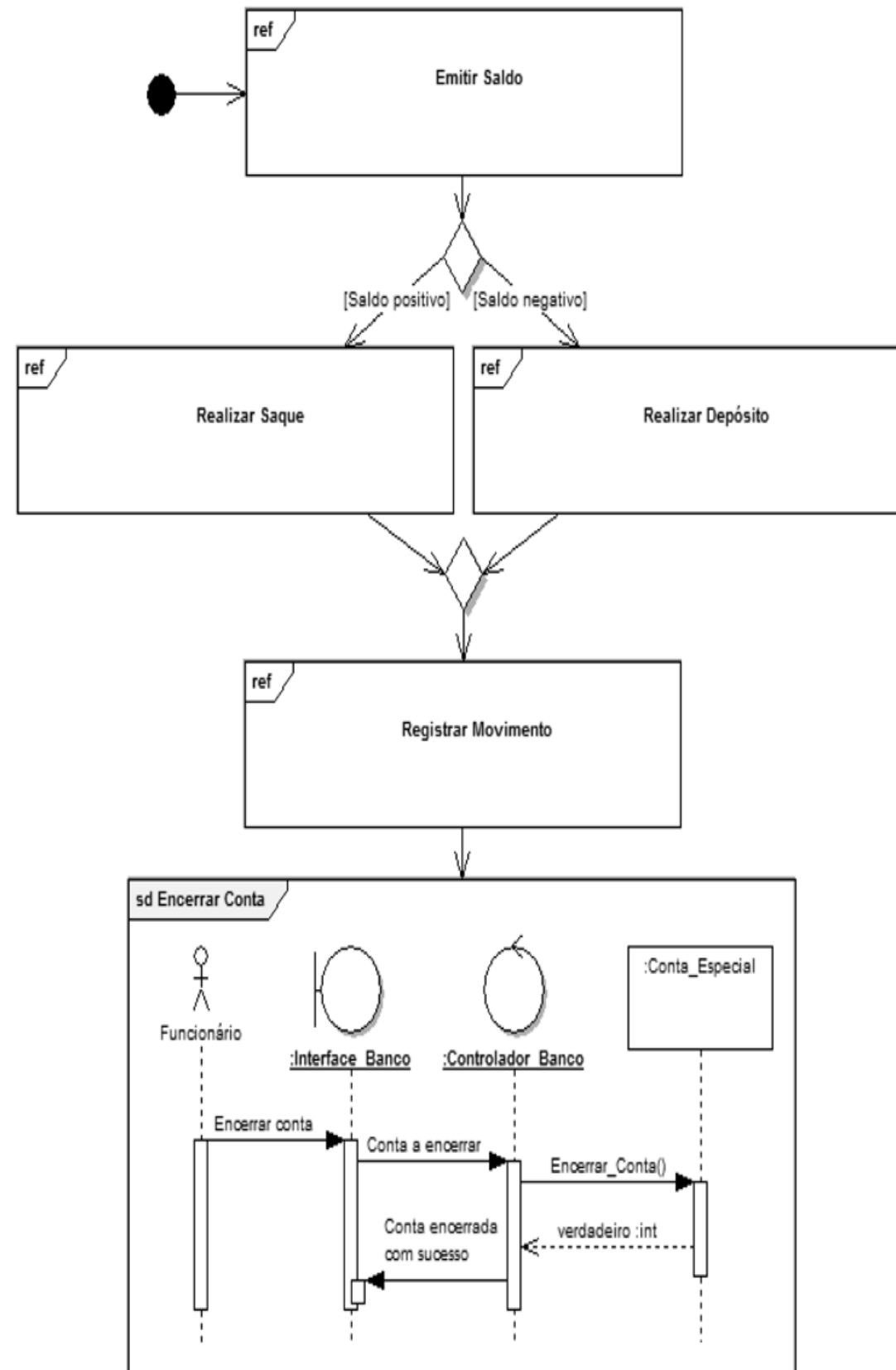


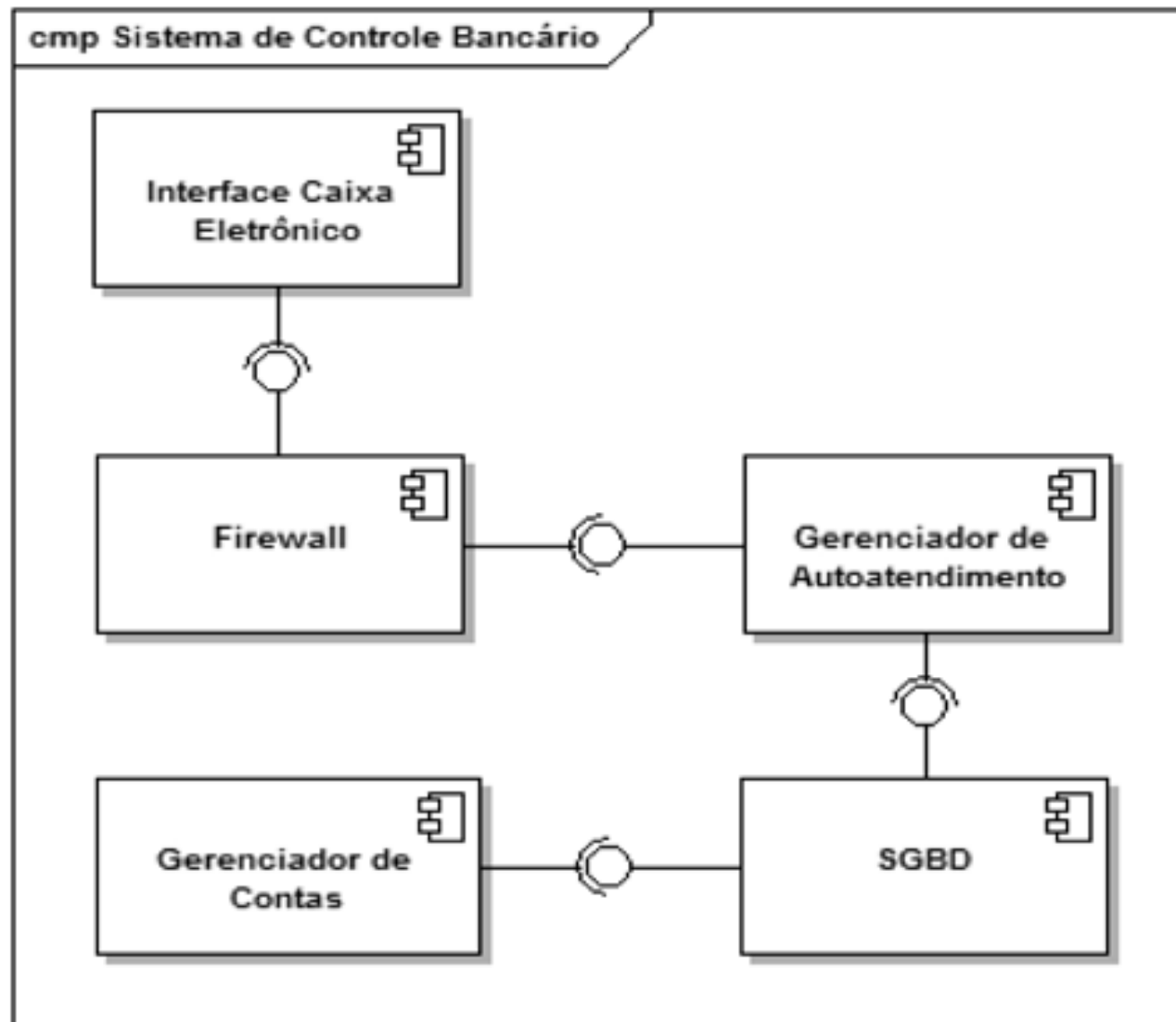
Figura 9 – Exemplo de Diagrama de Visão Geral de Interação.



## 11 - Diagrama de Componentes

O diagrama de componentes está amplamente associado à linguagem de programação que será utilizada para desenvolver o sistema modelado. Esse diagrama representa os componentes do sistema quando for implementado.

## 11 - Diagrama de Componentes (2)



*Figura 10 – Exemplo de Diagrama de Componentes.*

## 12 - Diagrama de Implantação

O diagrama de implantação determina as necessidades de hardware do sistema, as características físicas como servidores, estações, topologias e protocolos de comunicação, ou seja, todo o aparato físico sobre o qual o sistema deverá ser executado. A figura 11 apresenta um exemplo desse diagrama.

## 12 - Diagrama de Implantação (2)

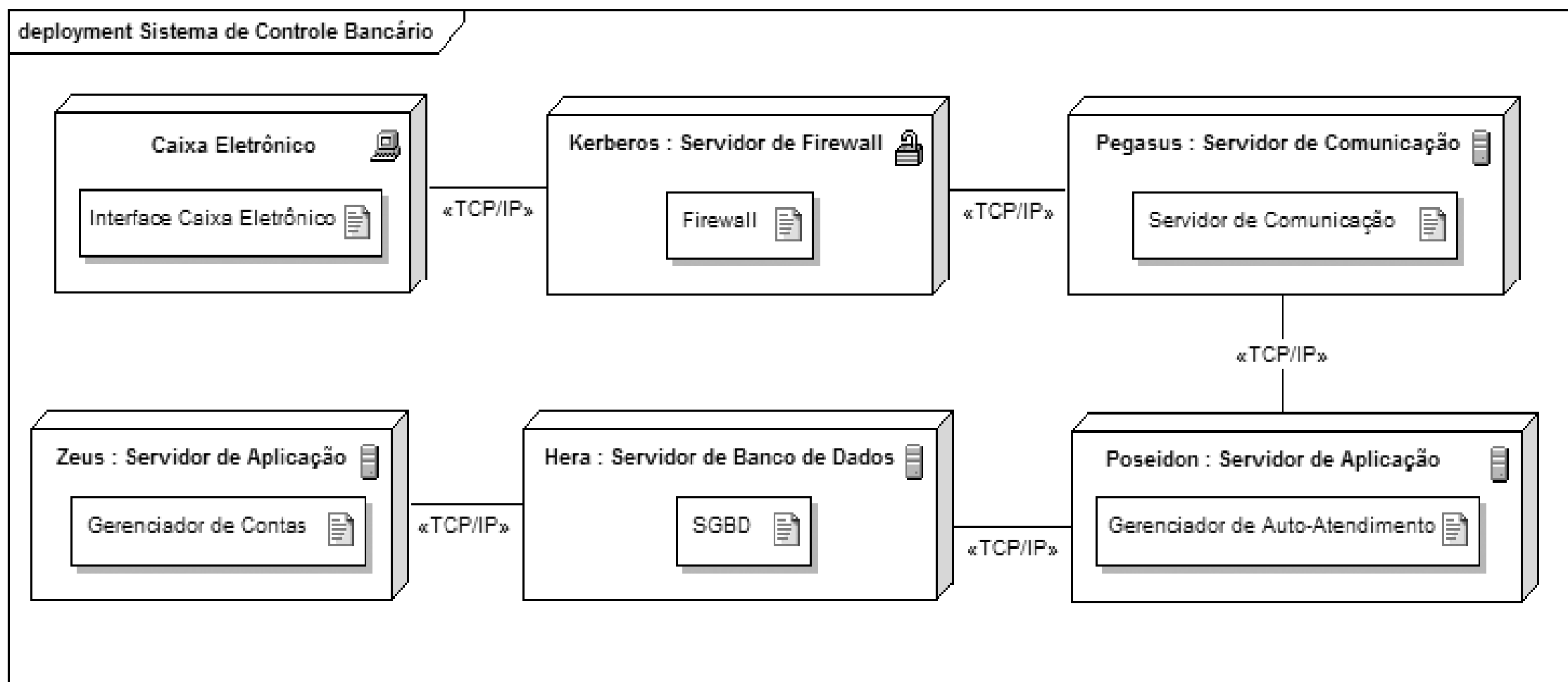
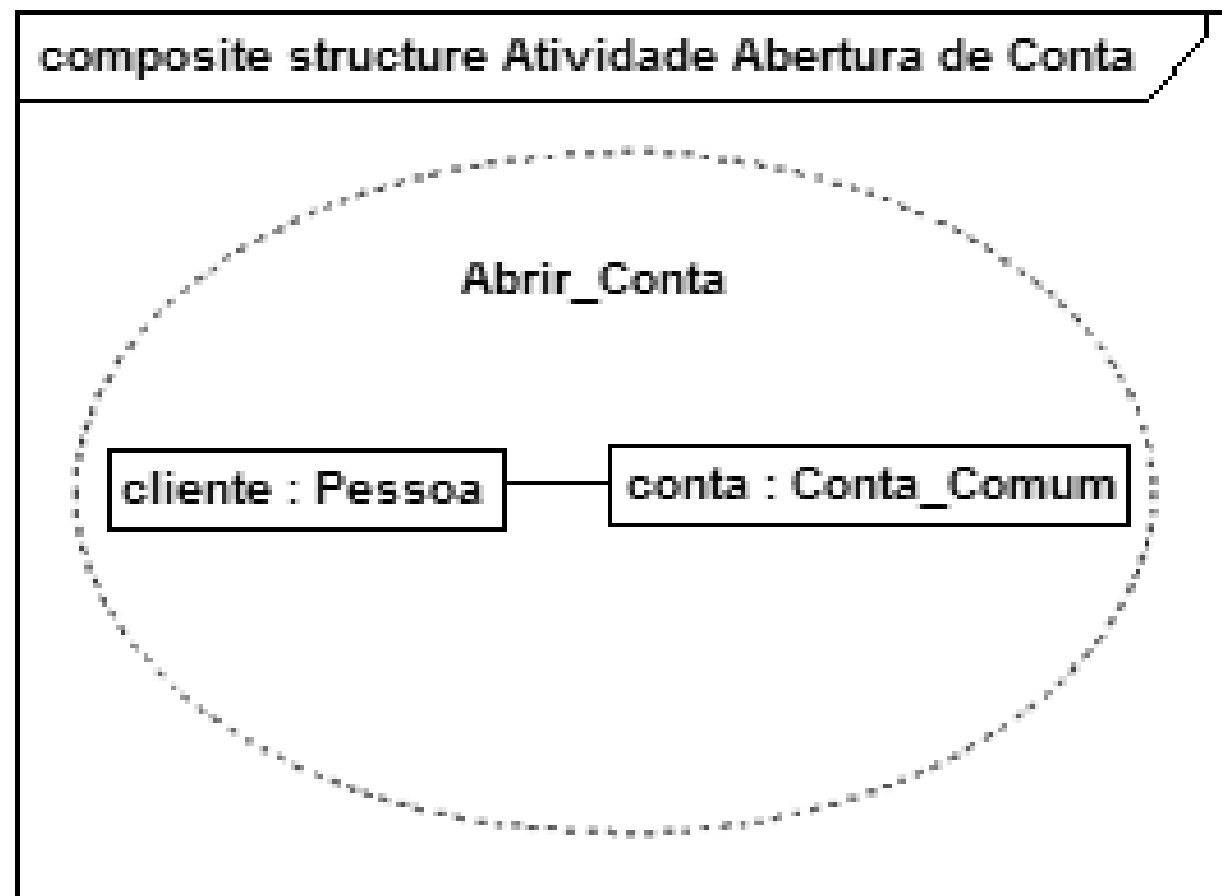


Figura 11 – Exemplo de Diagrama de Implantação.

## 13- Diagrama de Estrutura Composta

O diagrama de estrutura composta descreve a estrutura interna de um classificador, como uma classe ou componente, detalhando as partes internas que o compõem, como estas se comunicam e colaboram entre si. Também é utilizado para descrever uma colaboração em que um conjunto de instâncias cooperam entre si para realizar uma tarefa. A figura 12 mostra um exemplo de diagrama de estrutura composta.

## 13- Diagrama de Estrutura Composta (2)



*Figura 12 – Exemplo de Diagrama de Estrutura Composta.*



## 14- Diagrama de Tempo ou de Temporização

O diagrama de tempo descreve a mudança no estado ou condição de uma instância de uma classe ou seu papel durante um período.

Tipicamente utilizado para demonstrar a mudança no estado de um objeto no tempo em resposta a eventos externos. A figura 13 apresenta um exemplo desse diagrama.

## 14- Diagrama de Tempo ou de Temporização (2)

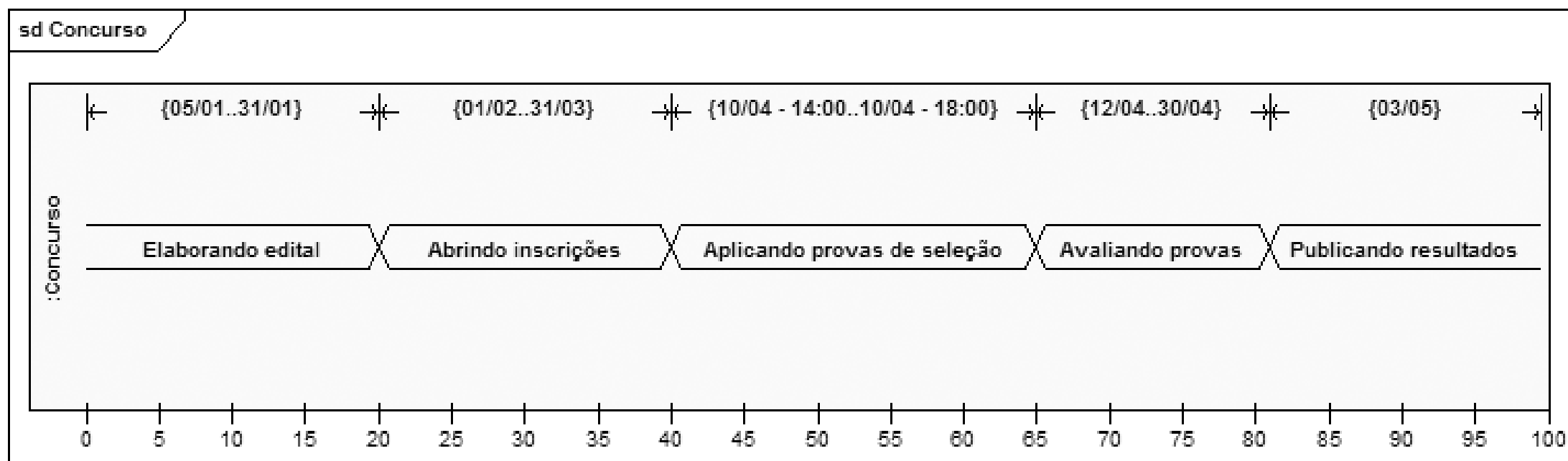


Figura 13 – Exemplo de Diagrama de Temporização.

## 15 – Visão Geral dos Diagramas da UML

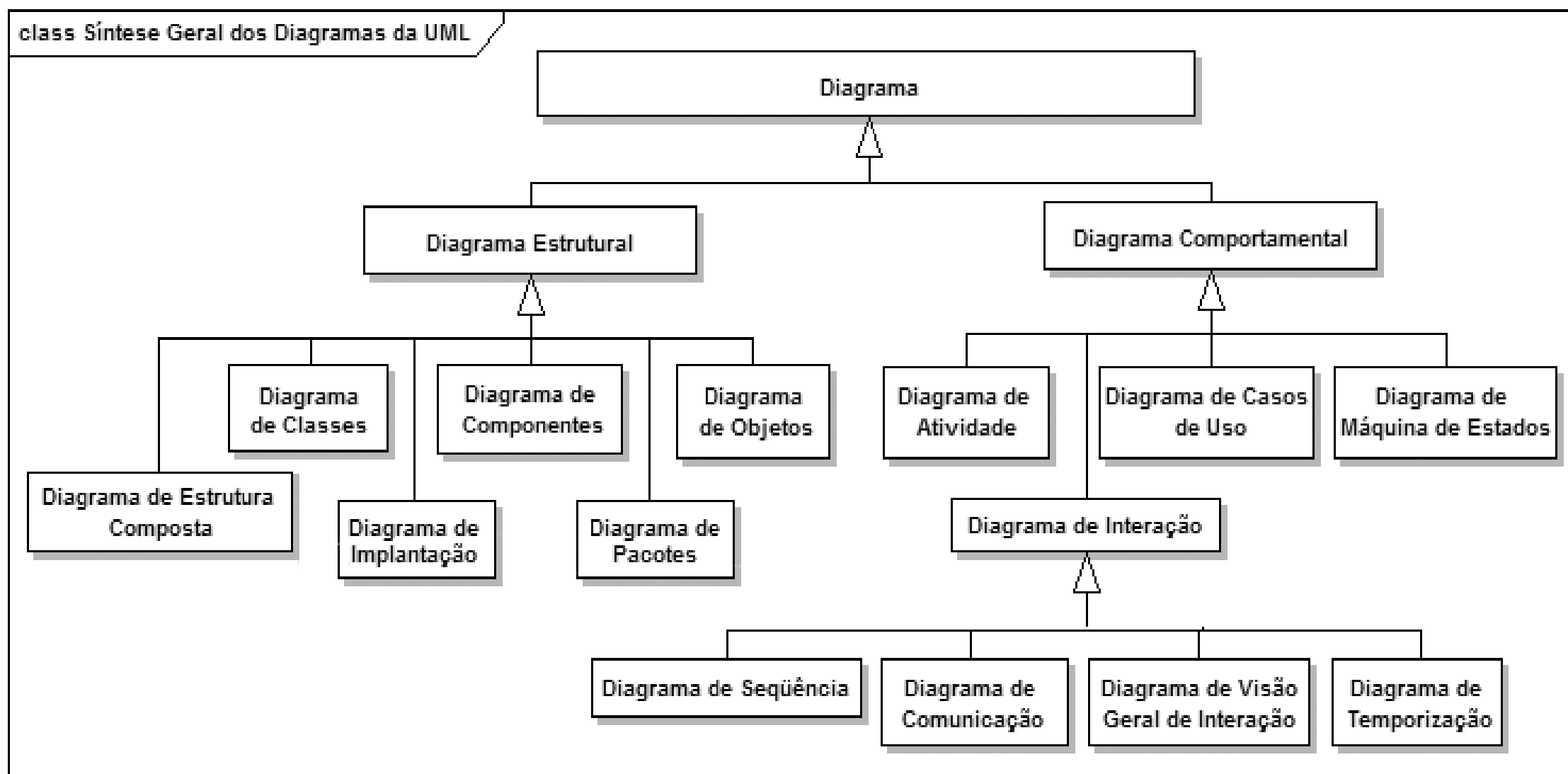


Figura 14 – Diagramas da UML.