

# Introdução à UML

Prof. Humberto Torres Marques Neto

Setembro / 2018

PUC Minas

# Referência Bibliográfica

**BOOCH, Grady, RUMBAUGH, James, JACOBSON, Ivair. UML: Guia do Usuário. 2 ed. Rio de Janeiro: Campus, 2005.**

**COCKBURN, Alistair. Escrevendo casos de uso eficazes: um guia prático para desenvolvedores de software. Porto Alegre: Bookman, 2005.**

**LARMAN, Craig. Utilizando UML e padrões: uma introdução à análise e ao projeto orientados a objetos. 3 ed. Porto Alegre: Bookman. 2007.**

**BOOCH, Grady. Object oriented analysis and design with applications. 2. ed. Addison Wesley Publishing, 1993. 586p.**

**COAD, Peter, YOURDON, Edward. Análise baseada em objetos. Rio de Janeiro: Campus, 1996. 225p.**

**BEZERRA, Eduardo. Princípios de análise e projeto de sistemas com UML. Rio de Janeiro: Campus, 2002.**

# Administração da Complexidade

*“When designing a complex software system, it is essential to decompose it into smaller and smaller parts, each of which we may then refine independently.” (BOOCH, 1994. p. 16)*

- Decomposição Algorítmica
  - **Projeto estruturado top-down**
  - **Ênfase na ordem que ocorre os eventos**
- Decomposição Orientada à Objetos
  - **Abstrações chave de um problema**
  - **Ênfase nos agentes que atuam ou sofrem algum impacto de uma ação relacionada ao sistema**

# Principais Métodos de Análise

- Decomposição Funcional
- Enfoque de Fluxo de Dados
- Modelagem de Informações
- Baseado em Objetos ou Orientado à Objetos

*Qual o melhor método?*

*É o melhor em qualquer caso?*

# Motivações e Benefícios da AOO

- Compreensão de domínios de problemas complexos
- Melhoria na interação entre o analista e o especialista de domínio de problema
- Aumento da consistência interna dos resultados da análise (estrutura e comportamento)
- Representação explícita dos pontos comuns
- Elaboração de especificações que suportam alterações

# Motivações e Benefícios da AOO

- Reutilização dos resultados de análise
- Apresentação de uma representação básica consistente (análise e desenho)
- Evolução das ferramentas CASE (*round trip engineering*)
- Desenvolvimento de sistemas multi-camadas

# O que é UML?

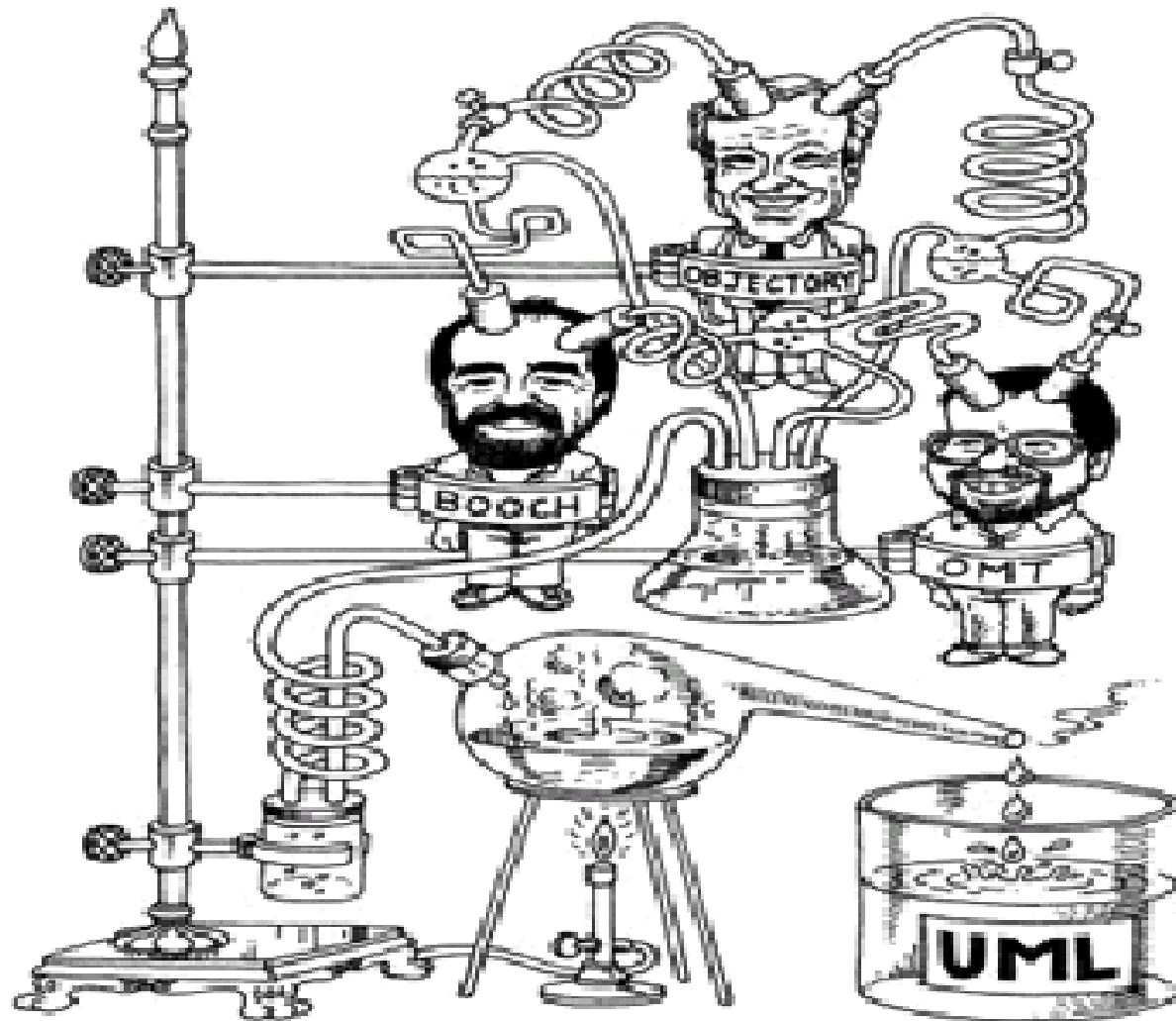
- UML significa “**Linguagem de Modelagem Unificada**”
- A UML combina o melhor de:
  - Conceitos de Modelagem de Dados (Diagramas de Entidade Relacionamento)
  - Modelagem de Negócios (work flow)
  - Modelagem de Objetos
  - Modelagem de Componentes

# O que é UML?

- A UML é a padronização da linguagem de desenvolvimento orientado a objetos para visualização, especificação, construção e documentação de todos os artefatos de um sistemas
- Pode ser usada com todos os tipos de processos, em todo o ciclo do desenvolvimento do software e através de diferentes tecnologias de implementação



Como surgiu a UML?



# Histórico da UML

- **1975 - 1980** - surgimento das linguagens de modelagem orientadas a objetos.
- **1989 - 1994** - número de métodos aumentou em cinco vezes
- Alguns métodos:
  - Booch (bom para as fases de projeto e análise)
  - OOSE (Object-Oriented Software Engineering) de Jacobson (bom suporte para os casos como uma forma de encontrar os requisitos, a análise e o projeto em alto nível)
  - OMT (Object Modeling Technique) de Rumbaugh (bom na análise e projeto de sistema com grandes volumes de dados)
  - Outros: Coad-Yourdon, Fusion, Mellor

# Histórico da UML

- Em **Out/1994**, Rumbaugh se une a Booch na Rational para unificar o OMT com o Modelo proposto por Booch
- Em **Out/1995**, quando Rumbaugh e Booch lançaram a primeira versão da UML – 0.8, Jacobson se reuniu ao grupo
- Em **1996** os três lançaram a versão 0.9 da UML
- Em **Jan/1997** a UML 1.0, construída com o apoio de um consórcio formado por DEC, HP, IBM, Unisys, Microsoft, Oracle, Texas Instruments, entre outros, foi submetida ao OMG (Object management group) e foi adotada em **Nov/1997**
- A versão atual da UML é a 2.0

# Utilização da UML

- Mostrar os limites de um sistema e suas principais funções utilizando casos de uso e seus atores
- Ilustrar as realizações dos casos de uso através de diagramas de interação
- Representar a estrutura estática do sistemas utilizando diagramas de classe

# Utilização da UML

- Modelar o comportamento de Objetos com diagramas de estado
- Exibir a arquitetura de implementação física com os diagramas de componente e implantação
- Pode ter suas funcionalidades estendidas com estereótipos

# Alguns Benefícios da UML

- Define um mapeamento da análise à implementação sem descartar o desenho
- Facilita os testes
- Define uma notação expressiva e consistente
- Facilita a comunicação entre as pessoas
- Ajuda a apontar inconsistências e omissões
- Suporta o processo de desenvolvimento para grandes e pequenos sistemas

# Características da UML

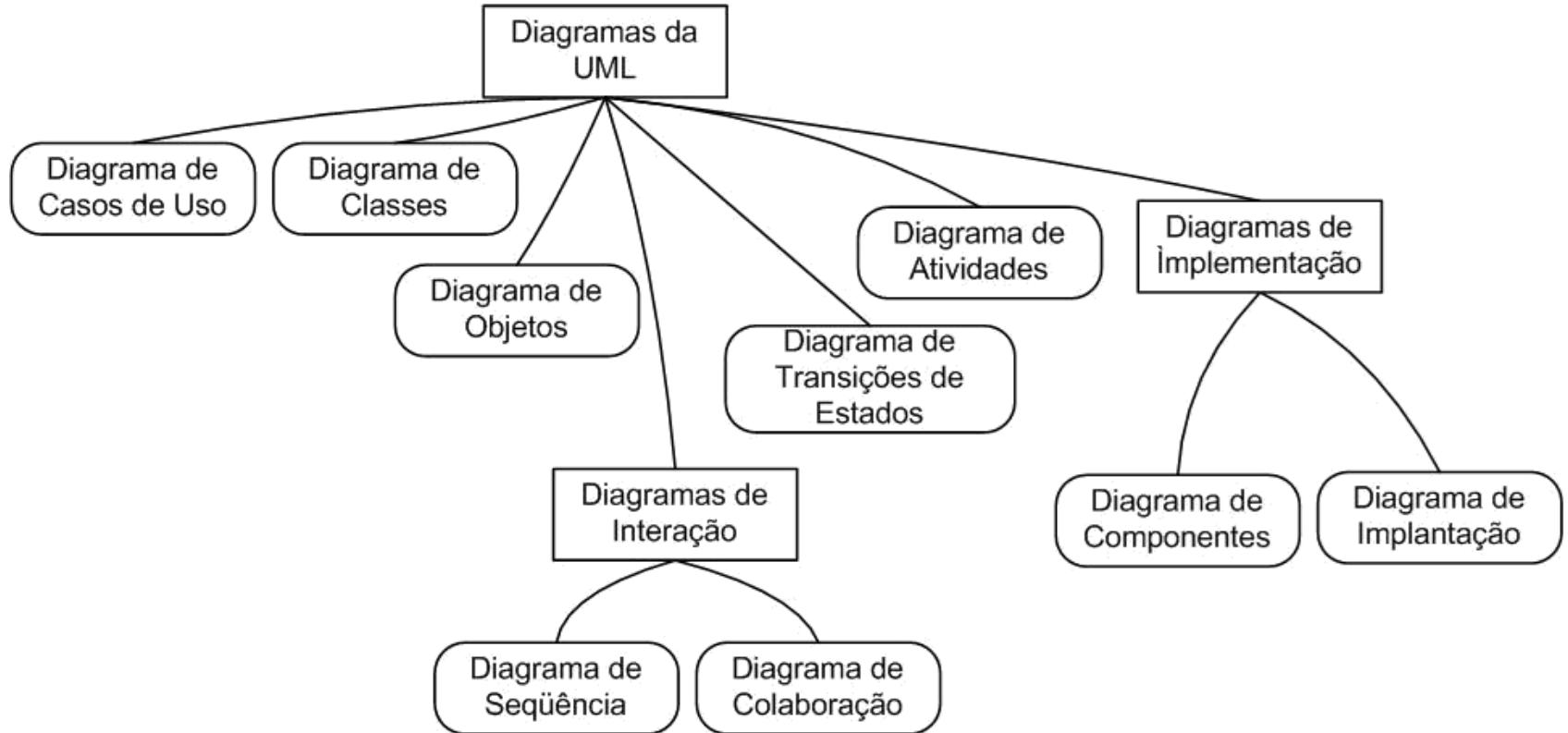
- UML é...
  - uma linguagem visual.
  - independente de linguagem de programação.
  - independente de processo de desenvolvimento.
- UML **não** é...
  - uma linguagem programação.
  - uma técnica de modelagem.

# Diagramas da UML

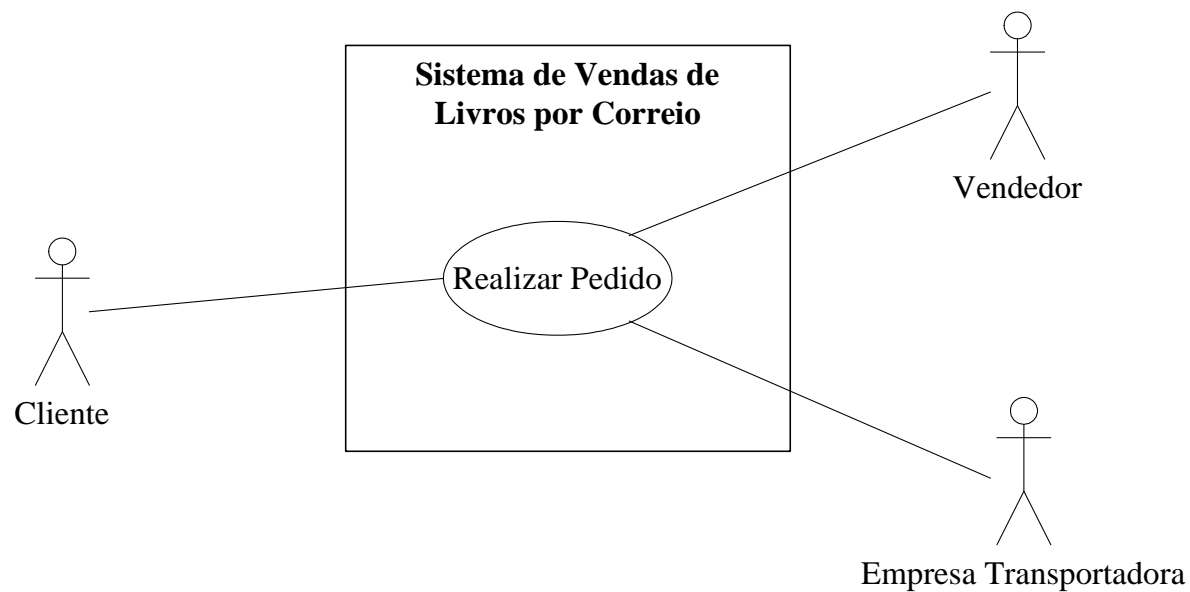
- Um processo de desenvolvimento que utilize a UML como linguagem de modelagem envolve a criação de diversos documentos.
  - Estes documentos podem ser textuais ou gráficos.
  - Estes documentos são denominados **artefatos de software**.
  - São os artefatos que compõem as visões do sistema.
- Os artefatos gráficos produzidos durante o desenvolvimento de um sistema de software são definidos através da utilização dos **diagramas da UML**.



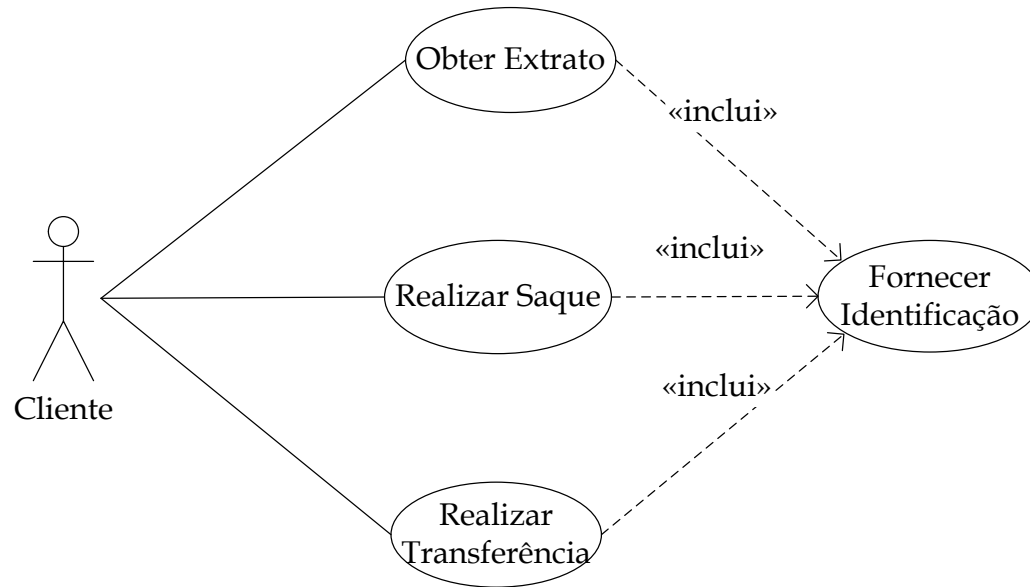
# Diagramas da UML



# Diagrama de Casos de Uso



# Diagrama de Casos de Uso



# Caso de uso

## Descrição Contínua

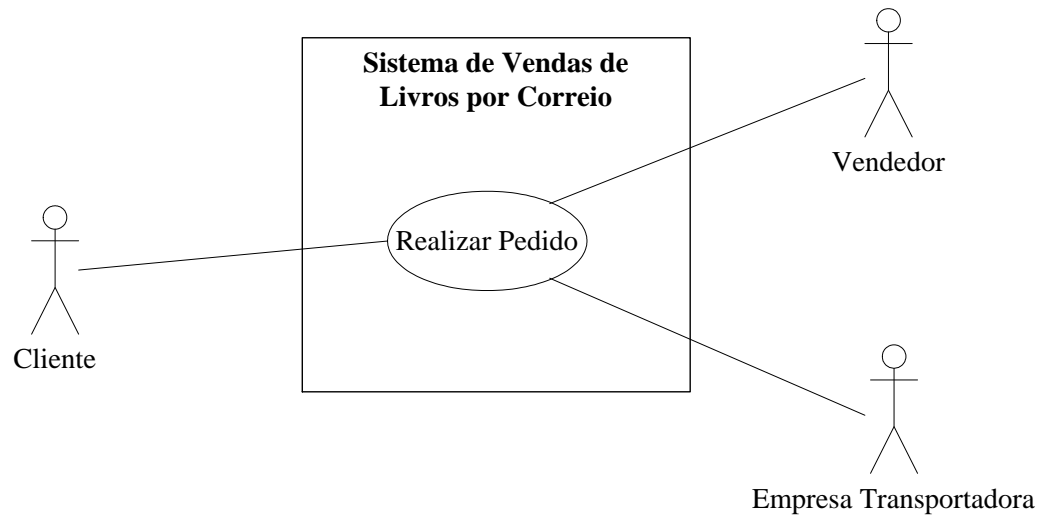
- O Cliente chega ao caixa eletrônico e insere seu cartão. O Sistema requisita a senha do Cliente. Após o Cliente fornecer sua senha e esta ser validada, o Sistema exibe as opções de operações possíveis. O Cliente opta por realizar um saque. Então o Sistema requisita o total a ser sacado. O Sistema fornece a quantia desejada e imprime o recibo para o Cliente.

# Caso de uso

## Descrição Numerada

1. Cliente insere seu cartão no caixa eletrônico.
2. Sistema apresenta solicitação de senha.
3. Cliente digita senha.
4. Sistema exibe menu de operações disponíveis.
5. Cliente indica que deseja realizar um saque.
6. Sistema requisita quantia a ser sacada.
7. Cliente retira a quantia e recibo.

# Diagrama de Casos de Uso

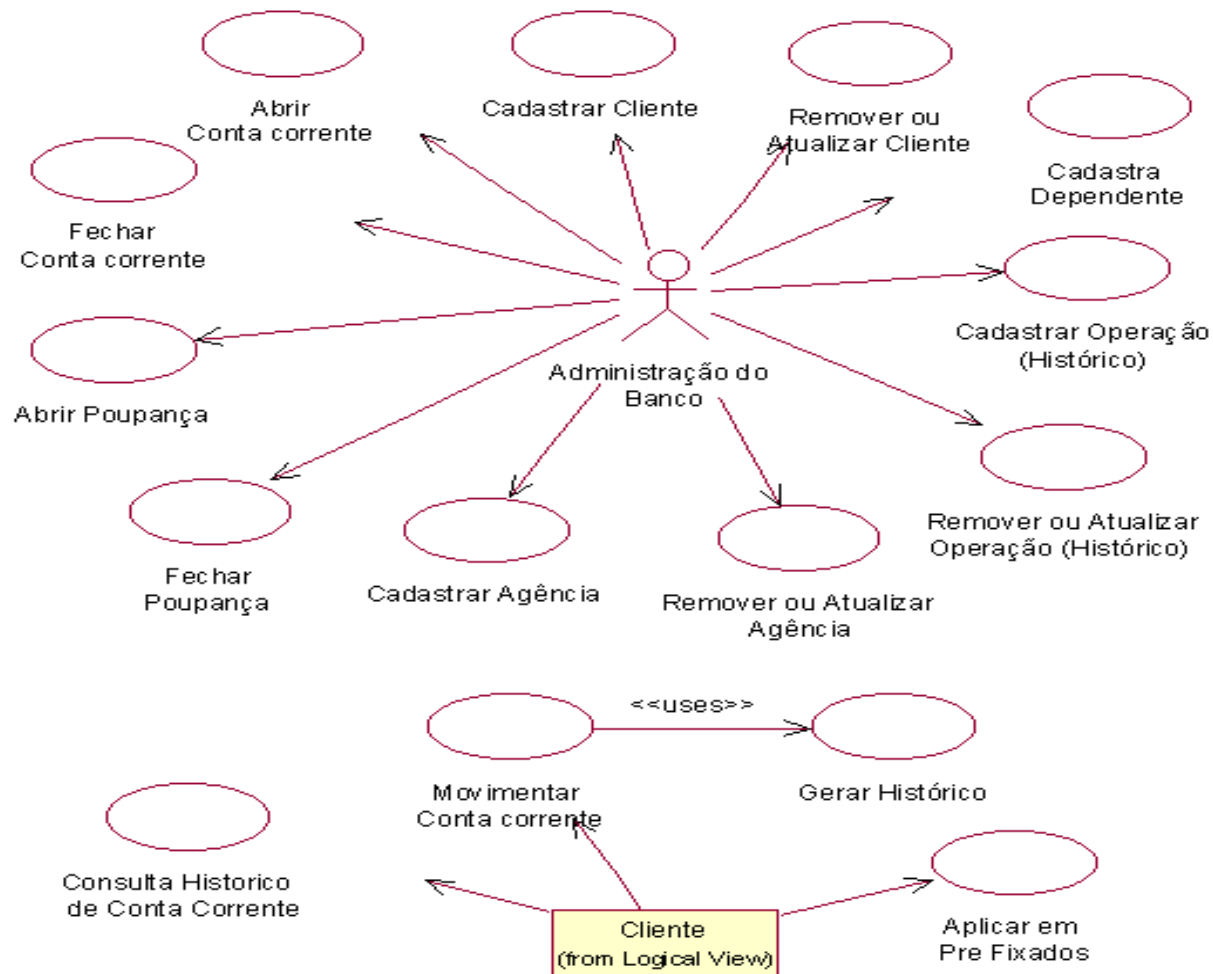


# Caso de uso

## Cenários

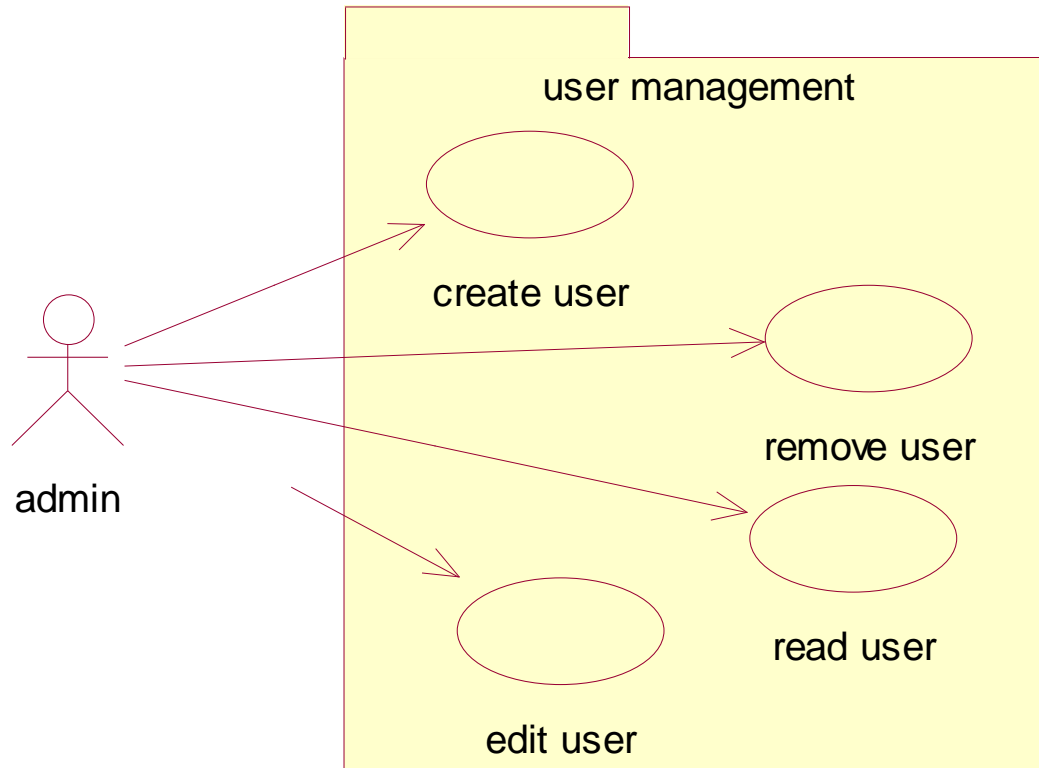
- Um Cliente telefona para a empresa.
- Um Vendedor atende ao telefone.
- Cliente declara seu desejo de fazer um pedido de compra.
- Vendedor pergunta a forma de pagamento.
- Cliente indica que vai pagar com cartão de crédito.
- Vendedor requisita o número do cartão, a data de expiração e o endereço de entrega.
- Vendedor pede as informações do primeiro item.
- Cliente fornece o primeiro item.
- Vendedor pede as informações do segundo item.
- Cliente fornece o segundo item
- Vendedor pede as informações do terceiro item
- Cliente e informa o terceiro item.
- Vendedor informa que o terceiro item está fora de estoque.
- Cliente pede para que O Vendedor feche o pedido somente com os dois primeiros itens.
- Vendedor fornece o valor total, a data de entrega e uma identificação do pedido.
- Cliente agradece e desliga o telefone.
- Vendedor contata a Transportadora para enviar o pedido de O Cliente.

# Diagrama de Casos de Uso

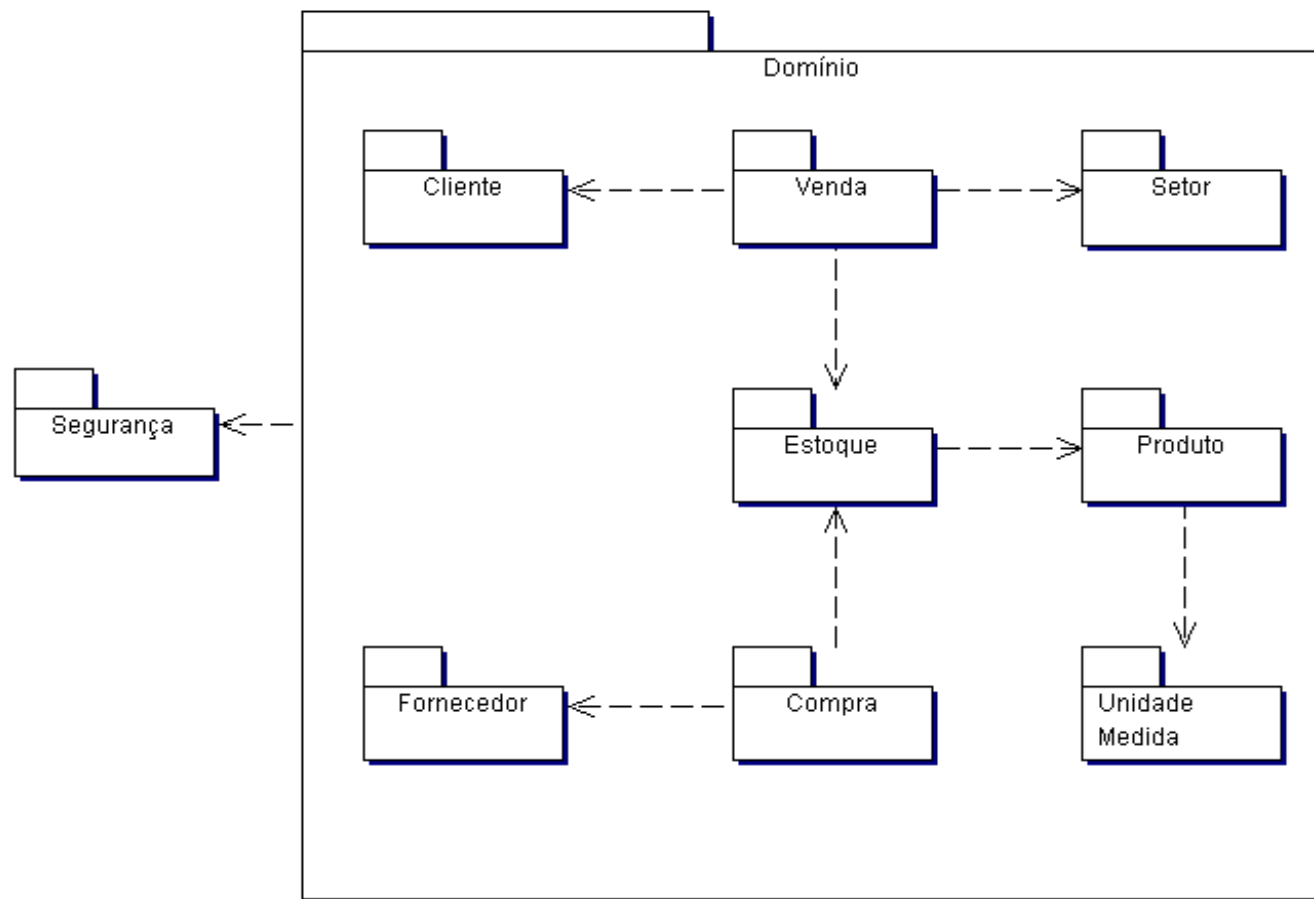




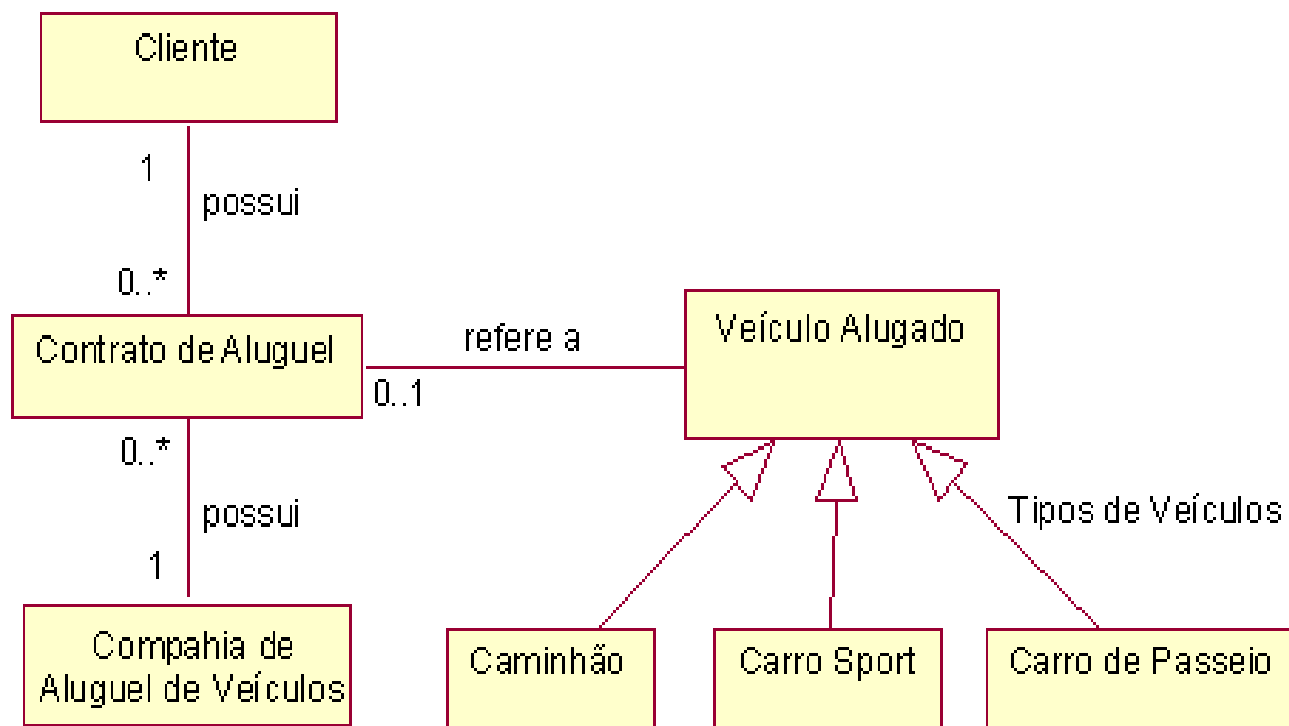
# Diagrama de Pacotes



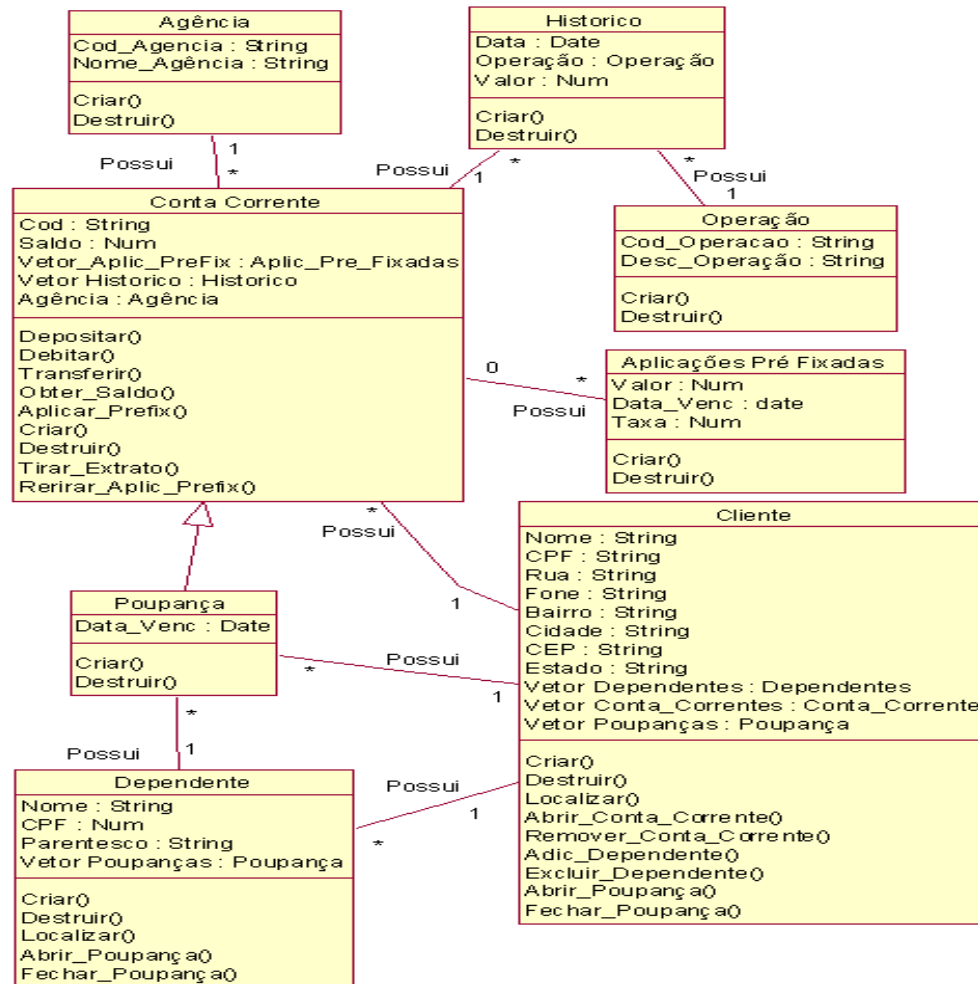
# Diagrama de Pacotes



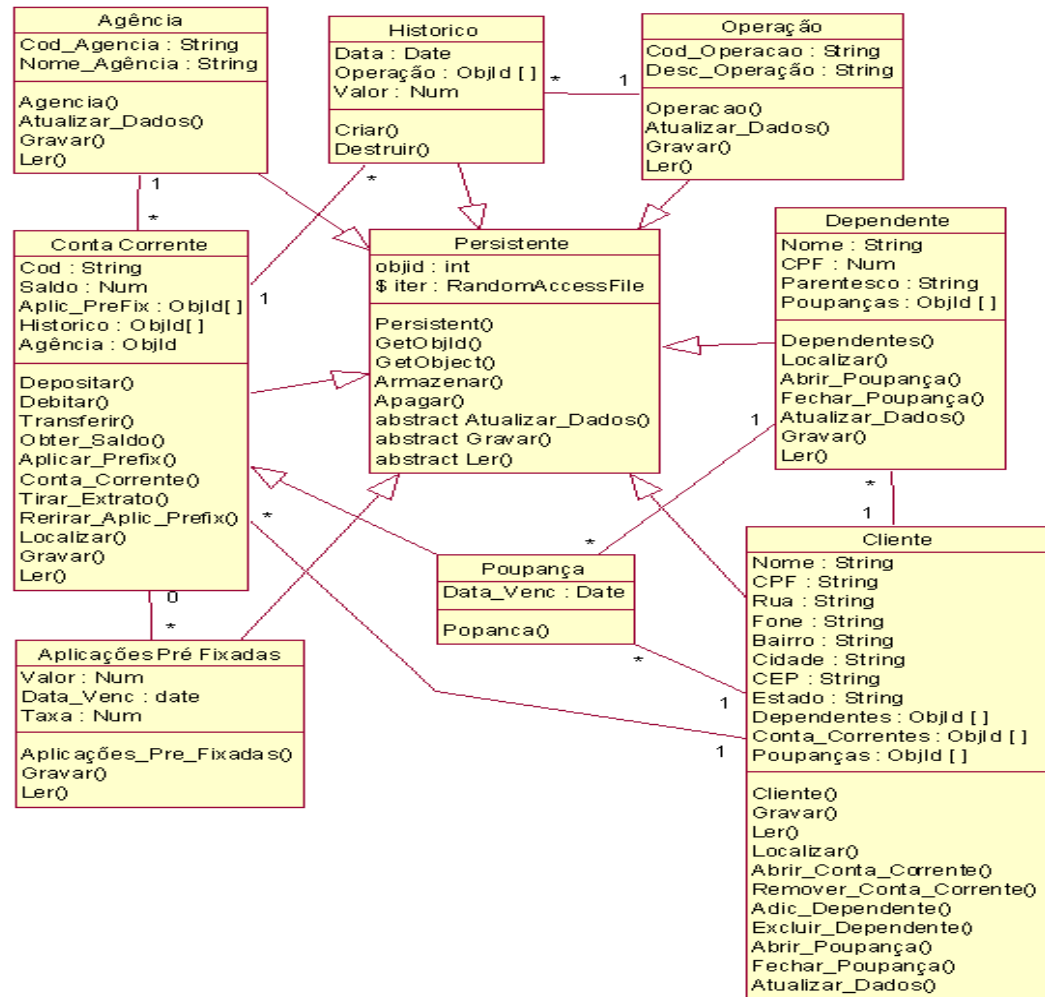
# Diagrama de Classes



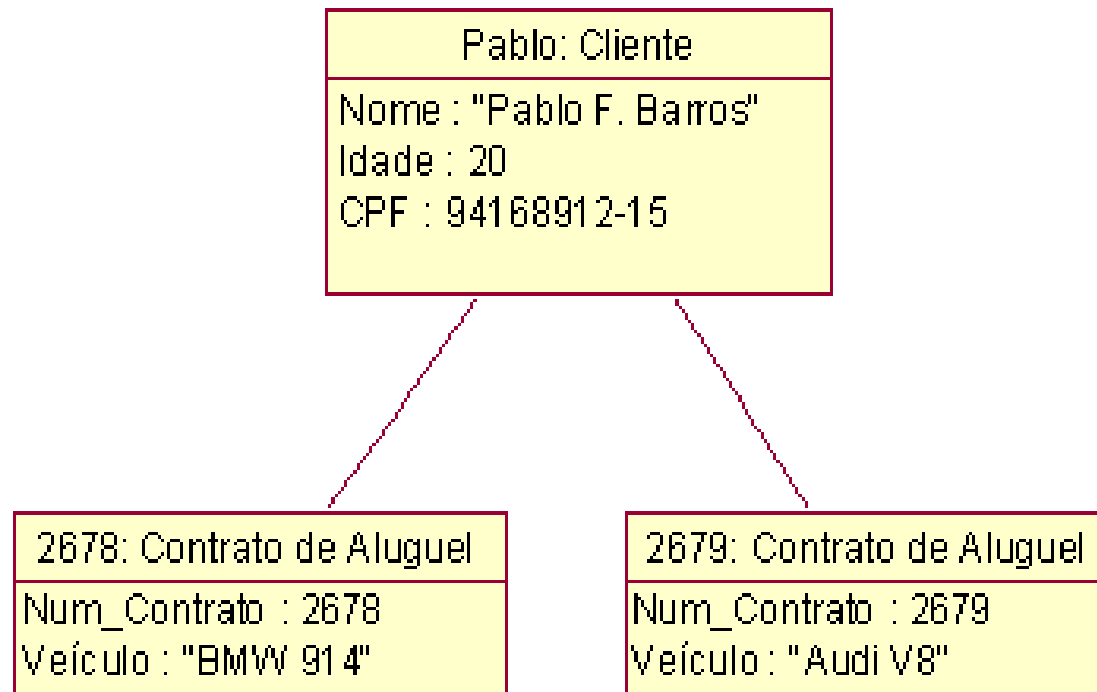
# Diagrama de Classes



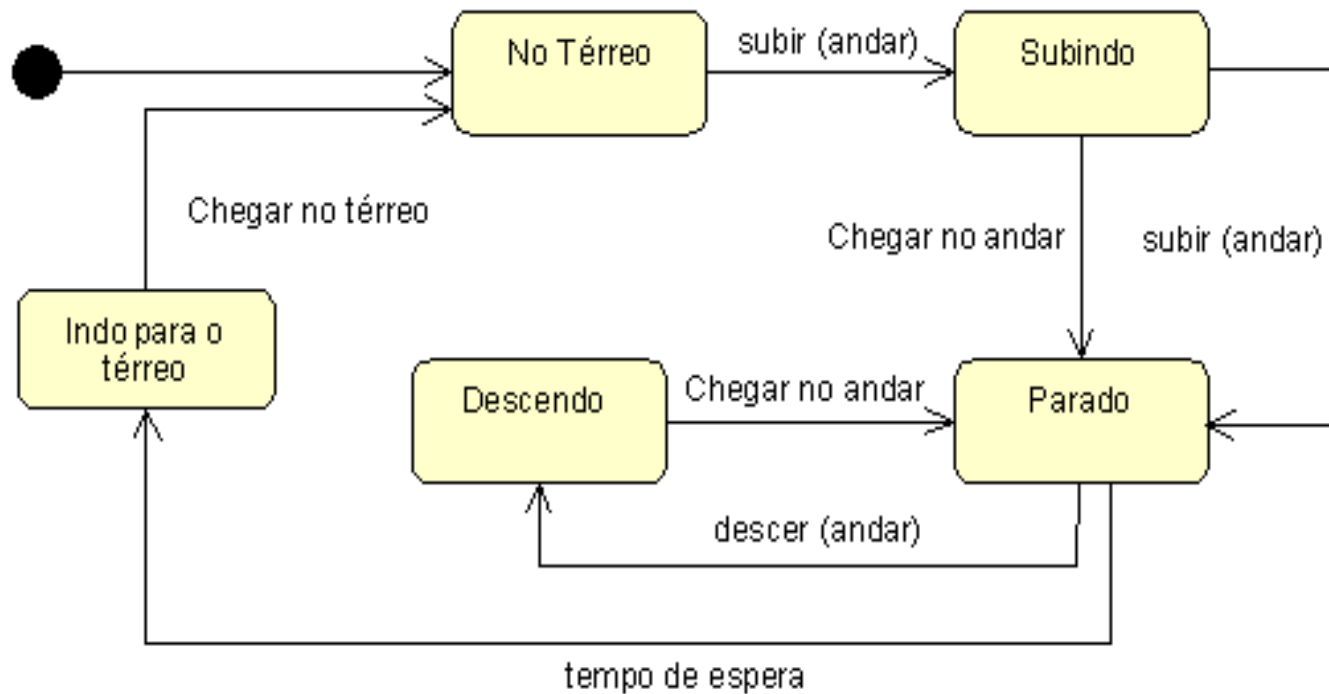
# Diagrama de Classes



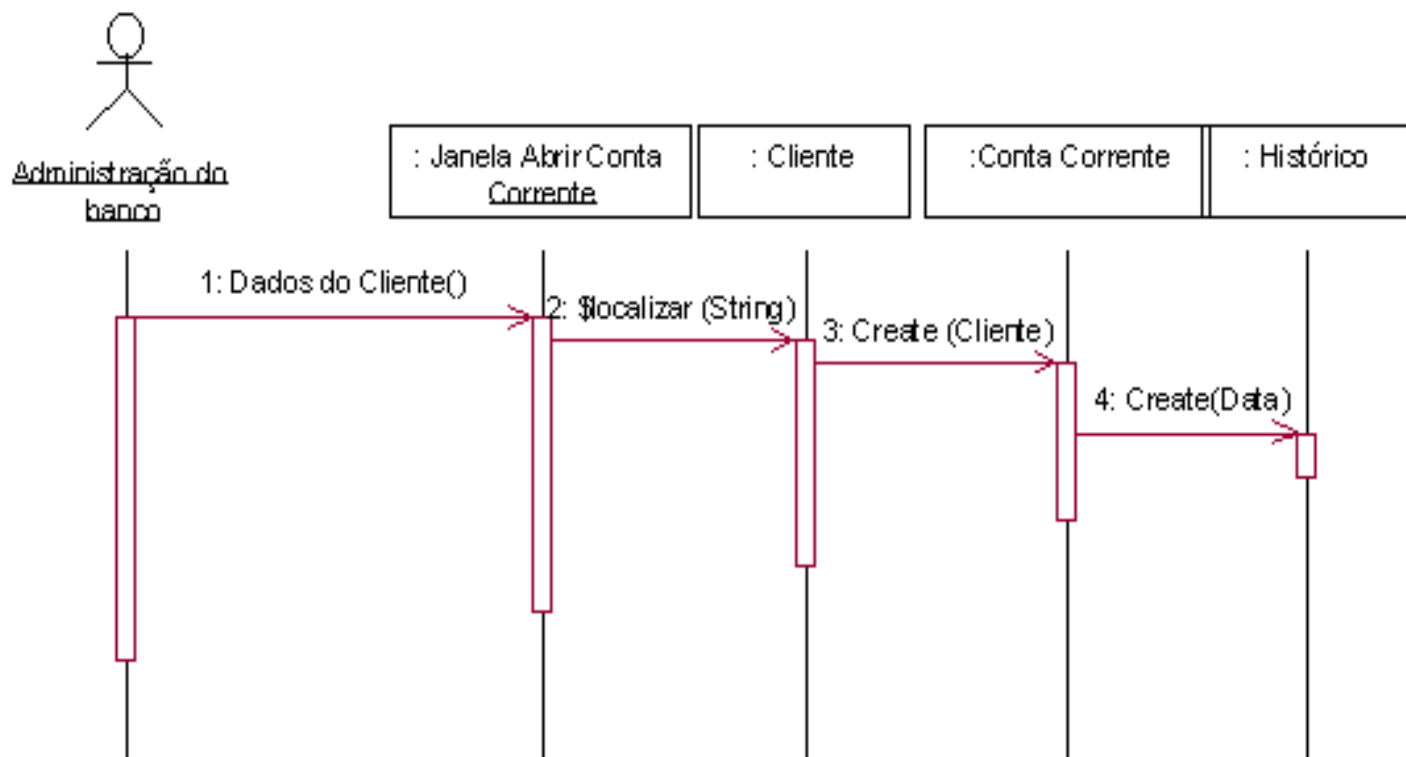
# Diagrama de Objetos



# Diagrama de Transição de Estados

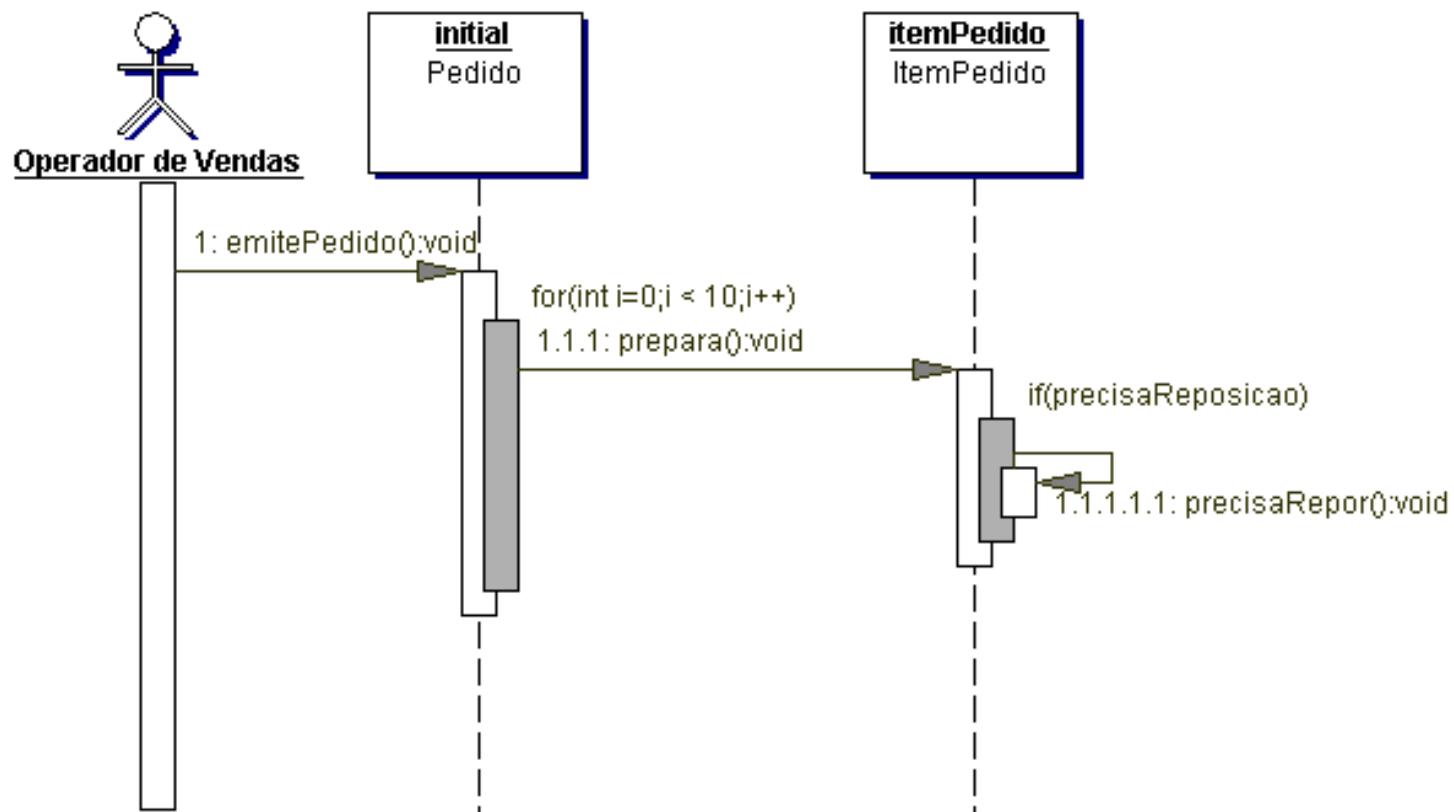


# Diagrama de Seqüência

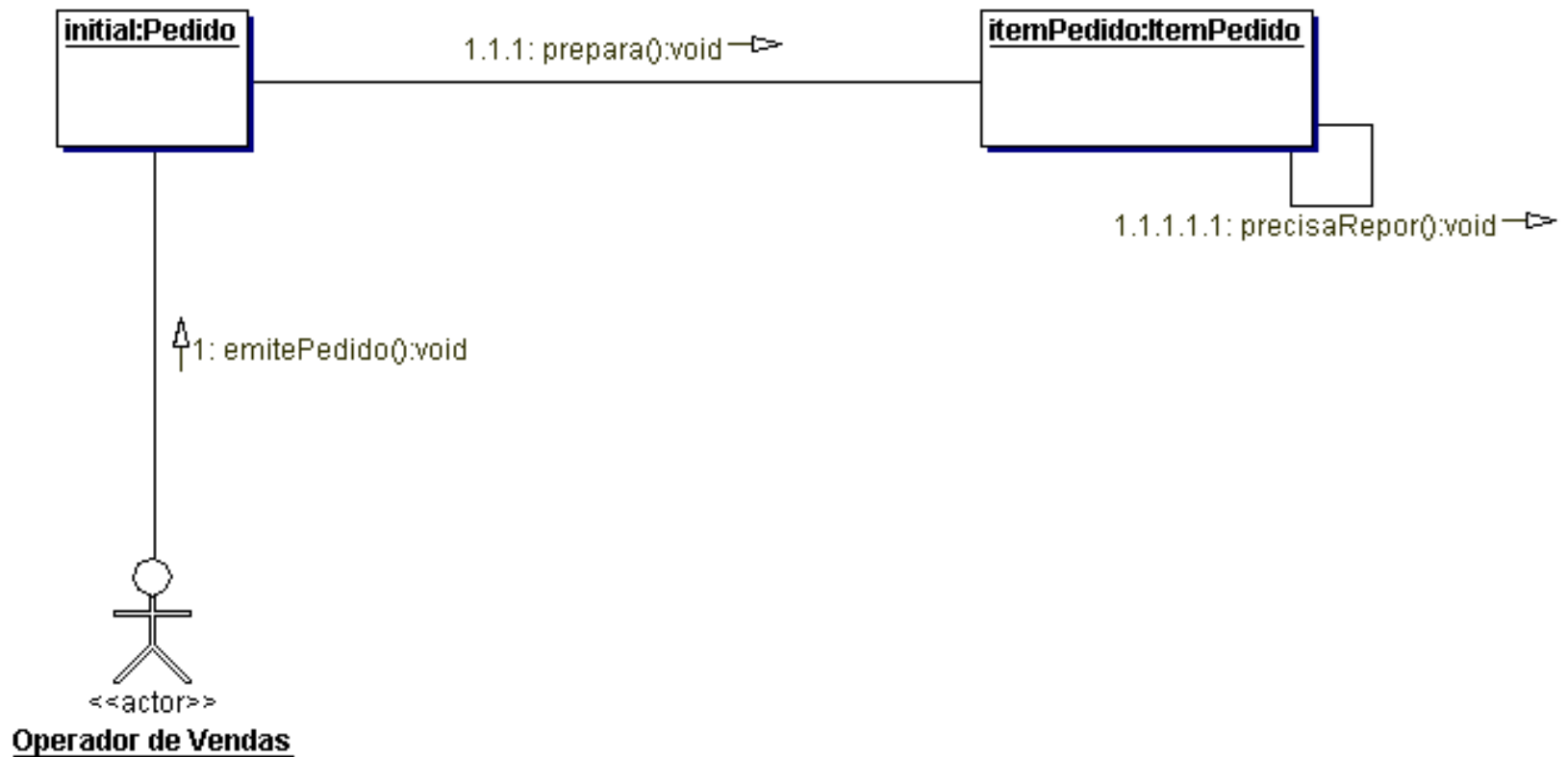




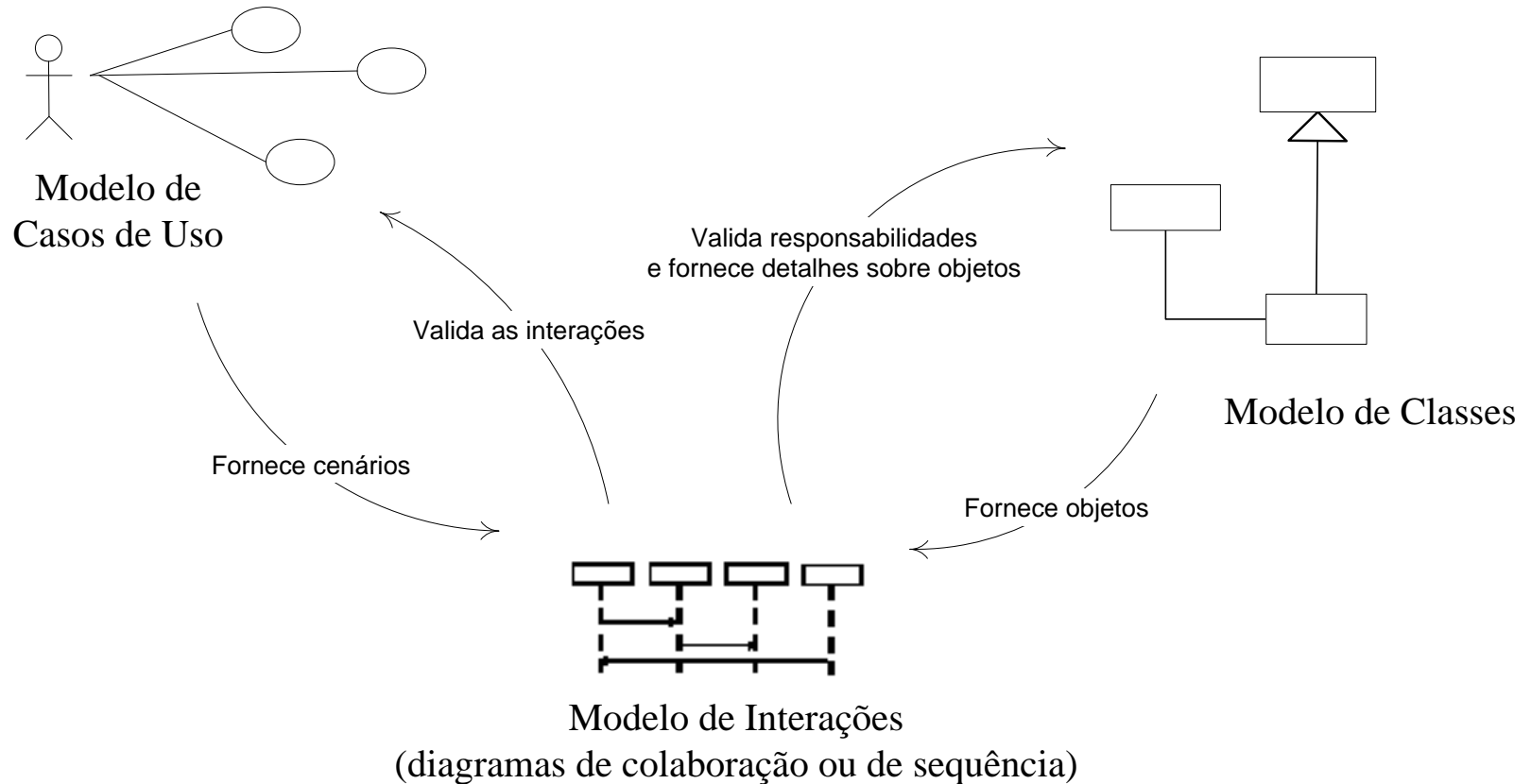
# Diagrama de Seqüência



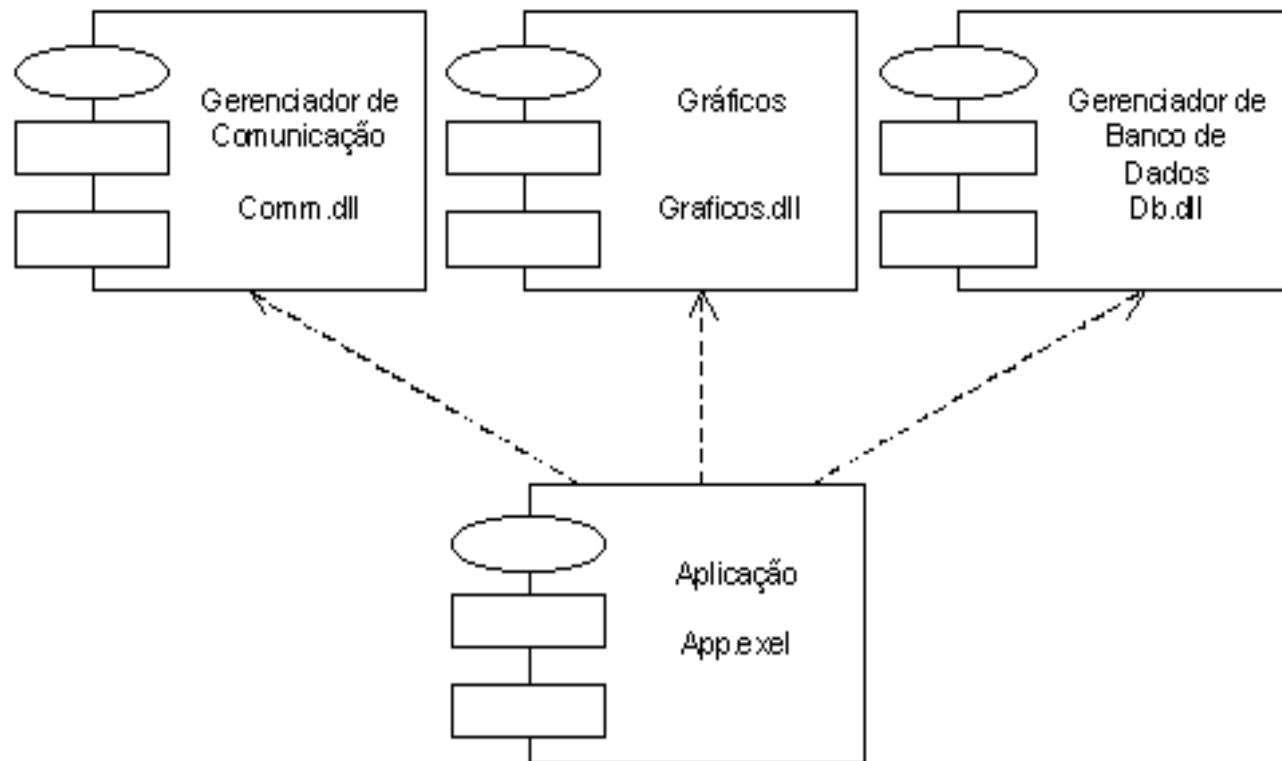
# Diagrama de Colaboração



# Interações, Casos de Uso e Classes



# Diagrama de Componentes



# Diagrama de Implantação

