



Modelos de Sistema

UNIP - Araraquara

Curso: Ciência da Computação

Disciplina: Engenharia de Software

Profº: João Paulo Moreira dos Santos

Roteiro

- Diagramação de Fluxo de Dados (DFD)
- Modelagem de Dados
 - Entidade
 - Relacionamento
 - Atributos
- UML

Modelos de Sistema

- Existem diversas formas de diagramar um projeto de software.
 - Fluxogramas;
 - Diagrama de Fluxo de Dados (DFD)
 - Diagrama de Entidade Relacionamento (DER)

Diagrama de fluxo de dados

- Um diagrama de fluxo de dados mapeia o fluxo de informações para qualquer processo ou sistema.
- Utiliza símbolos definidos para mostrar entradas e saídas de dados, pontos de armazenamento e as rotas entre cada destino.
 - Retângulos;
 - Círculos;
 - Flechas;
 - Rótulos de textos breves.
- Eles podem ser usados para analisar um sistema existente ou modelar um novo.

Diagrama de fluxo de dados

- Diagramas de fluxo de dados entraram em grande uso no final da década de 1970.
- Impulsionados pelo livro *Structured Design* ('Projeto estruturado de sistemas')
 - Ed Yourdon e Peter Coad.
- Os especialistas Tom DeMarco, Chris Gane e Trish Sarson também contribuíram para a maior divulgação e uso da metodologia DFD.

Diagrama de fluxo de dados

► Símbolos e notações utilizadas:

1. **Entidade externa:** um sistema externo que envia ou recebe dados, comunicando-se com o sistema sendo diagramado.
2. **Processo:** qualquer processo que altere os dados, produzindo uma saída. Pode realizar cálculos, classificar dados com base na lógica ou direcionar o fluxo de dados com base em regras de negócios.
3. **Armazenamento de dados:** arquivos ou repositórios que armazenam informações para uso posterior, como uma tabela de banco de dados ou um formulário de adesão.
4. **Fluxo de dados:** a rota dos dados entre entidades externas, processos e armazenamentos de dados.

Diagrama de fluxo de dados

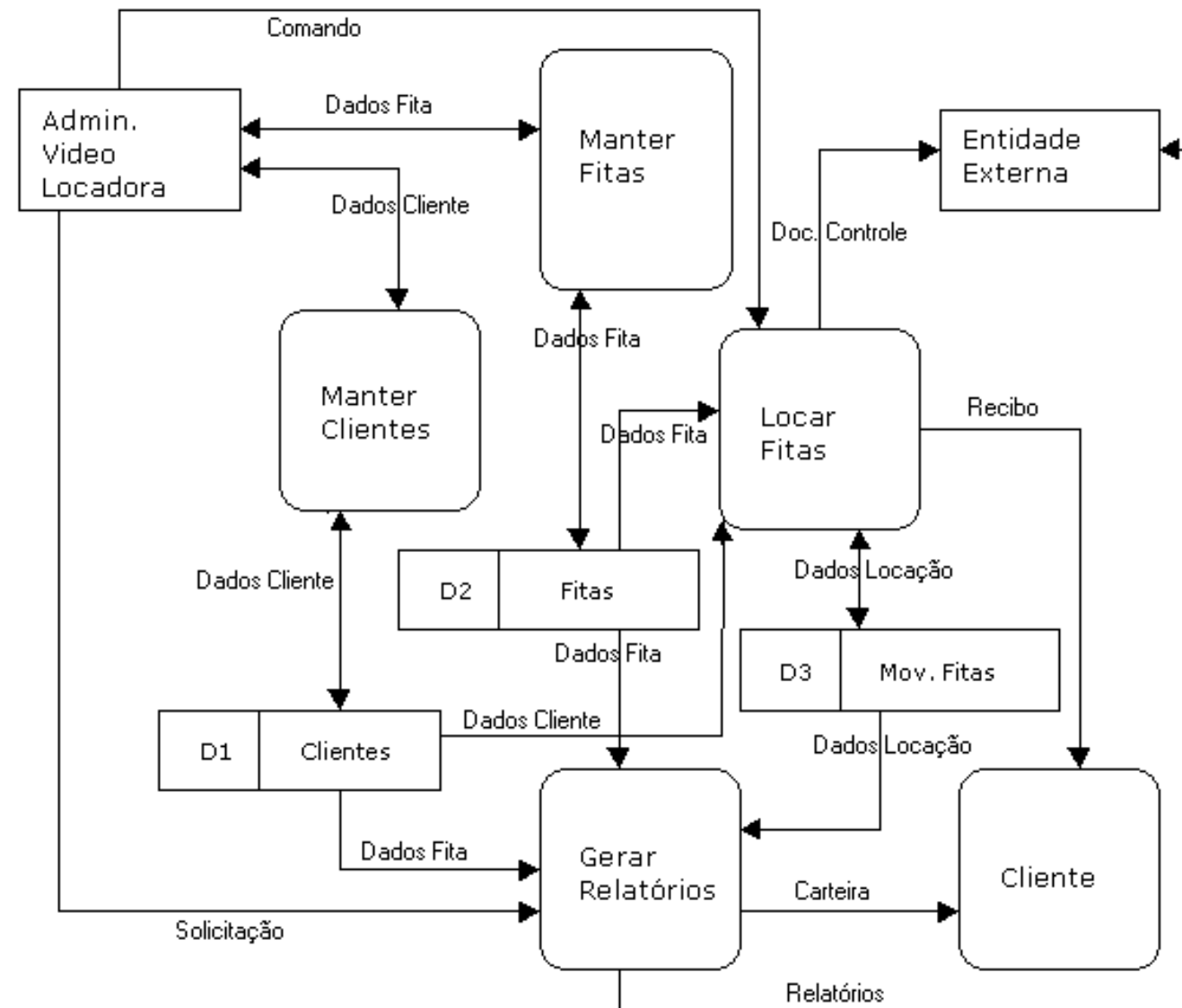


Diagrama de fluxo de dados

► Regras e dicas de DFDs:

- Cada processo deve ter, pelo menos, uma entrada e uma saída.
- Cada armazenamento de dados deve ter, pelo menos, um fluxo de dados de entrada e de um fluxo de dados de saída.
- Dados armazenados em um sistema devem passar por um processo.
- Todos os processos em um DFD vão para outro processo ou um armazenamento de dados.
- Dados armazenados em um sistema devem passar por um processo.

Diagrama de fluxo de dados

- Níveis e camadas de DFDs (0, 1 e 2):
- O DFD nível 0 é também chamado de diagrama de contexto. É uma visão geral básica de todo o sistema ou processo a ser analisado ou modelado. É projetado para ser uma visão de bater o olho, mostrando o sistema como um único processo de alto nível, com o seu relacionamento com entidades externas.
- O DFD nível 1 oferece maiores detalhes de peças do diagrama de contexto. Ele destaca as principais funções desempenhadas pelo sistema na medida em que você expande o processo de alto nível do diagrama de contexto em subprocessos.
- O DFD nível 2 aprofunda ainda mais partes do nível 1. Pode ser preciso mais texto para chegar ao nível necessário de detalhes em relação ao funcionamento do sistema.

Diagrama de fluxo de dados

► DFD x UML

- Enquanto o DFD ilustra como os dados fluem em um sistema, a UML, uma linguagem de modelagem utilizada em design de software orientado a objetos, oferece uma visão mais detalhada.
- O DFD fornece um bom ponto de partida, porém, durante o desenvolvimento do sistema, desenvolvedores podem recorrer aos diagramas UML, incluindo diagramas de classes e de estrutura, para atingir a especificidade necessária.

Modelagem de dados

- Modelo Entidade Relacionamento (MER).
 - Modelo conceitual utilizado na Engenharia de Software para descrever os objetos (**entidades**) envolvidos em um domínio de negócio, com suas características (**atributos**) e como elas se relacionam entre si (**relacionamentos**).
 - Baseado na percepção do mundo real, que consiste em um conjunto de objetos básicos e seus relacionamentos
 - Facilita o projeto de banco de dados, possibilitando especificar a estrutura lógica geral do banco de dados.

Modelagem de dados (MER)

► Entidades:

- É uma representação abstrata de um objeto do mundo real.
- São nomeadas com substantivos concretos ou abstratos que representem de forma clara sua função dentro do domínio.
- Ex: Cliente, Produto, Venda, entre outros.

Modelagem de dados (MER)

► Relacionamentos:

► Uma vez que as entidades são identificadas, deve-se definir como se dá o relacionamento entre elas.

► Relacionamento 1..1 (um pra um)

► Relacionamento 1..n ou 1..* (um pra muitos)

► Relacionamento n..n ou *..* (muitos para muitos)

► São nomeados com verbos ou expressões que representam a forma como as entidades interagem, ou a ação que uma exerce sobre a outra.

► Ex: Um ator escreve vários livros, enquanto um livro é escrito por vários autores.

Modelagem de dados (MER)

► Atributos:

- São características que descrevem cada entidade dentro do domínio.
- Alguns atributos representam valores únicos que identificam a entidade dentro do domínio e não podem se repetir.
 - Ex: Em um cadastro de cliente, poderia ser o CPF. (**Chave Primária**)
- Já outros atributos são chamados de **Chave Estrangeira** e geralmente estão ligados à chave primária da outra entidade.
 - Ex: Cliente tem a chave primária CPF, assim, a Venda possui também um campo “CPF do cliente” que se relaciona com o campo CPF da entidade Cliente.

Modelagem de dados

- Enquanto **MER** é um modelo conceitual, o Diagrama Entidade Relacionamento (**DER**) é a sua representação gráfica e principal ferramenta.
- O diagrama facilita a comunicação entre os integrantes da equipe, pois oferece uma linguagem comum utilizada tanto pelo analista, responsável por levantar os requisitos, e os desenvolvedores, responsáveis por implementar aquilo que foi modelado.

Modelagem de dados (DER)

- Em sua notação original, proposta por Peter Chen, as entidades deveriam ser representadas por retângulos, seus atributos por elipses e os relacionamentos por losangos, ligados às entidades por linhas, contendo também sua cardinalidade (1..1, 1..n ou n..n).



- Porém, notações mais modernas abandonaram o uso de elipses para atributos e passaram a utilizar o formato mais utilizado na UML, em que os atributos já aparecem listados na própria entidade. Essa forma torna o diagrama mais limpo e fácil de ser lido.

Modelagem de dados (DER)

► **Proprietário** *contata* **Corretor**

Um proprietário pode contatar vários corretores e um corretor pode ser contactado por vários proprietários.

► **Corretor** *atende* **Inquilino**

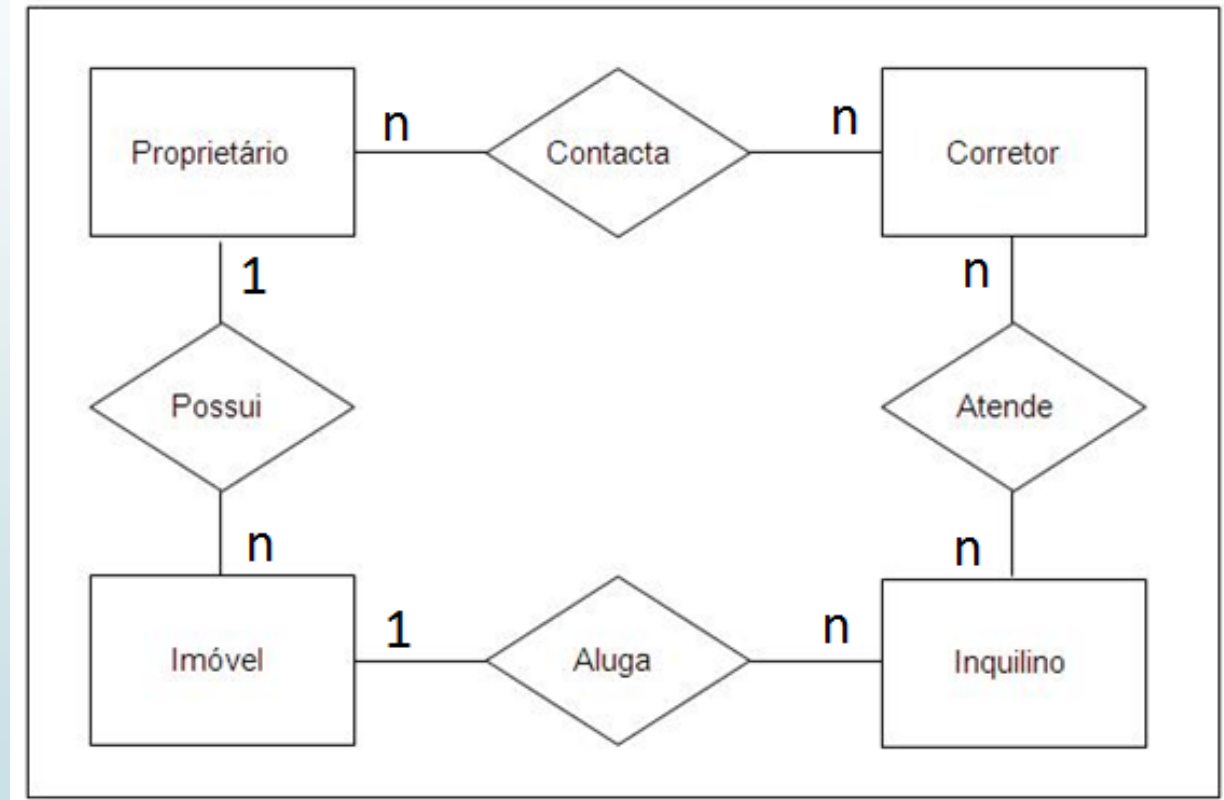
Um corretor pode atender vários inquilinos e um inquilino pode ser atendido por vários corretores.

► **Inquilino** *aluga* **Imóvel**

Um inquilino aluga um imóvel e um imóvel pode ser alugado por vários inquilinos.

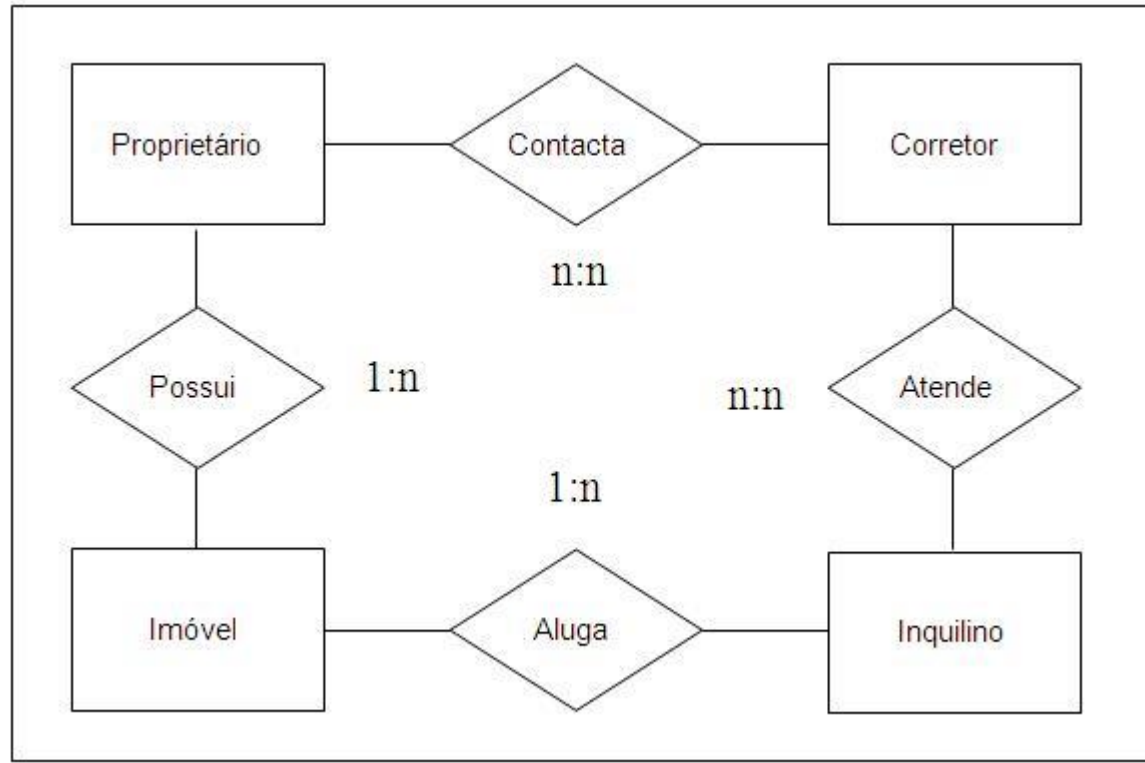
► **Proprietário** *possui* **Imóvel**

Um proprietário possui vários imóveis e um imóvel pertence a apenas um proprietário.



Modelagem de dados (DER)

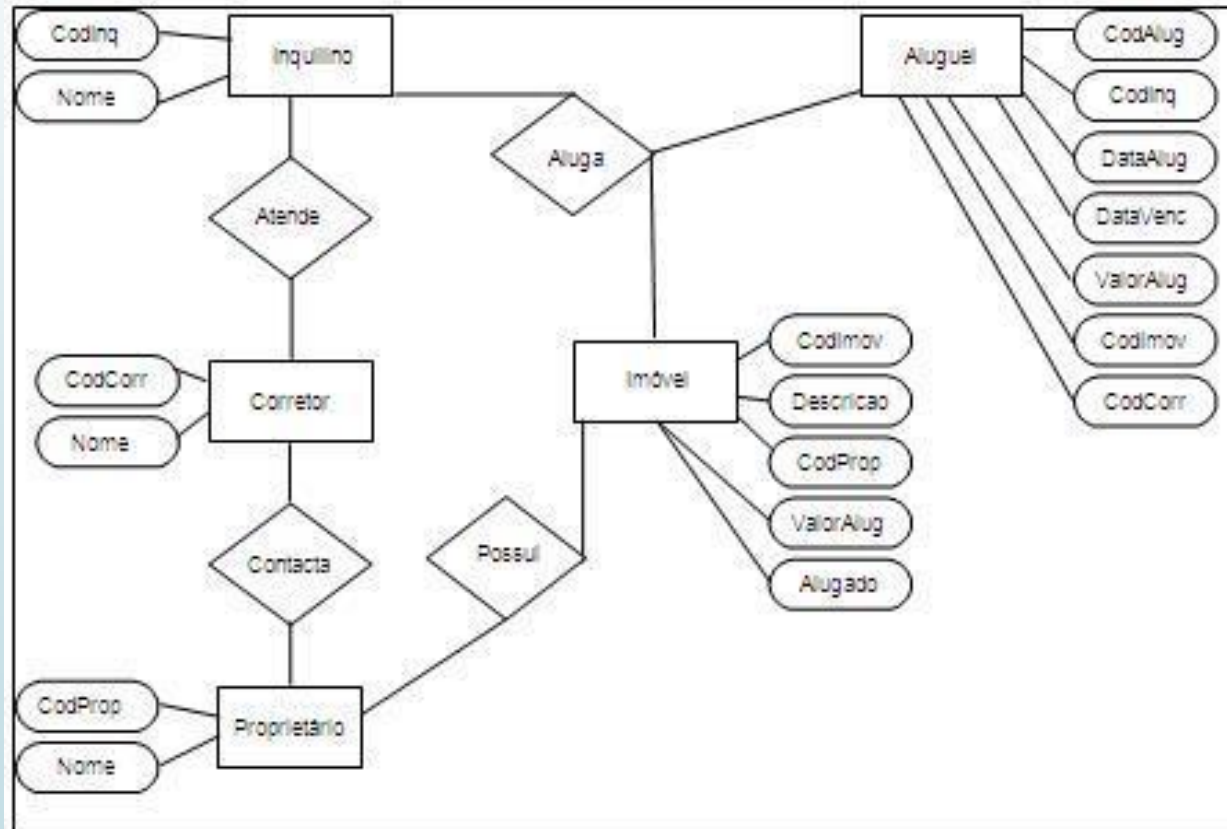
► Outras variações do diagrama:



- *1 ou 1 grupo possui 0 ou muitos produtos. Como de um lado temos “1 ou 1”, isso equivale a apenas “1”, pois não temos várias possibilidades. Já do lado do produto, indicamos que um grupo pode possuir nenhum produto, mas também pode possuir vários.*
- *0 ou várias vendas contém 1 ou muitos produtos. Ou seja, um produto pode nunca ser vendido (0 vendas) como também pode ser vendido várias vezes (n vendas). Já uma venda deve conter 1 ou vários produtos, pois uma venda não pode estar vazia (0 produtos).*

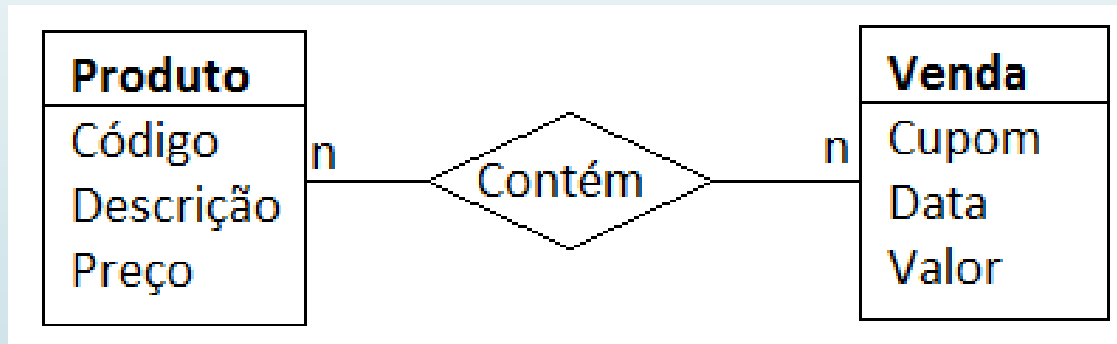
Modelagem de dados (DER)

- Os atributos podem aparecer no diagrama na forma de elipses ligadas às entidades. Essa foi a notação original proposta, ela deixa o diagrama com muitos itens e pode atrapalhar um pouco a organização destes.



Modelagem de dados (DER)

- Em uma notação mais atual, comumente utilizada na UML, os atributos aparecem listados dentro do próprio retângulo da entidade, enquanto o nome da entidade aparece no topo na forma de título.

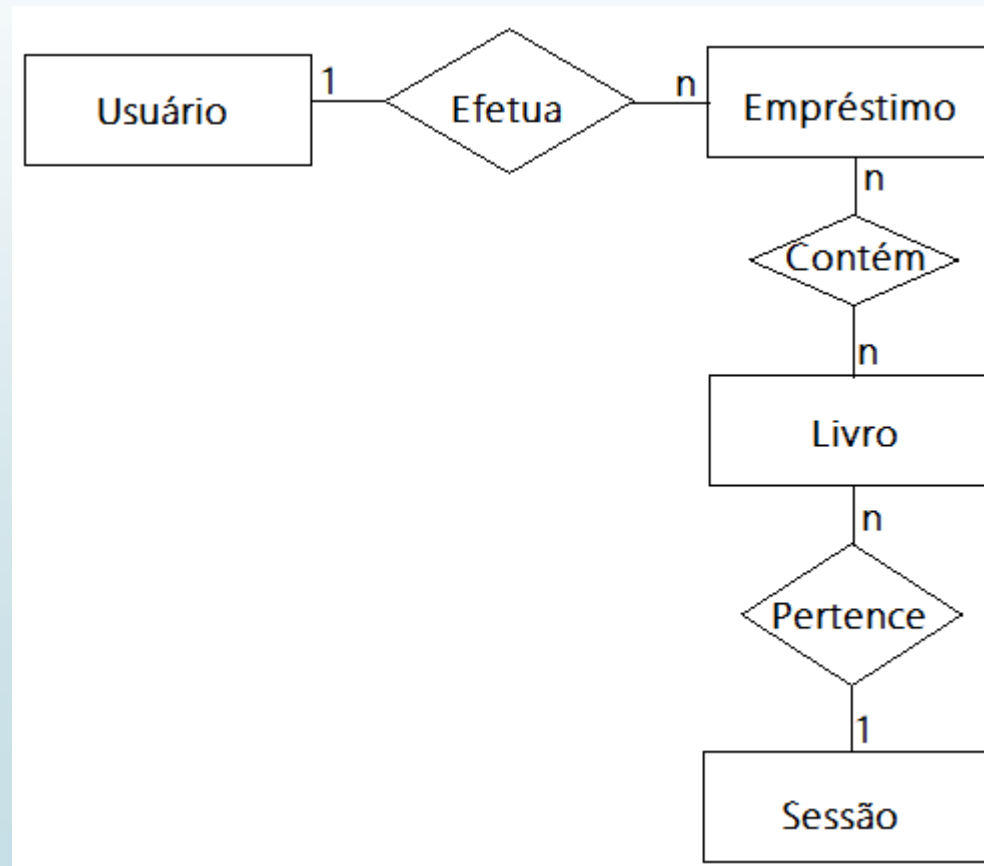


Modelagem de dados (DER)

- Exemplo: Sistema de biblioteca.
- Um usuário efetua um empréstimo de um livro da biblioteca. Todos os livros pertencem a uma sessão, na qual ajuda na organização das obras.
- Entidades:
 - Usuário
 - Empréstimo
 - Livro
 - Sessão

Modelagem de dados (DER)

► Exemplo: Sistema de biblioteca.





UML

Unified Modeling Language

Introdução à UML

- A UML é uma linguagem gráfica utilizada para: Visualizar, Especificar, Construir e Documentar artefatos de um sistema.
- O objetivo é descrever qualquer tipo de sistema.
- Características:
 - Abrange sempre qualquer característica de um sistema em um de seus diagramas;
 - É aplicada em diferentes fases do desenvolvimento e em qualquer tipo de sistema:
 - Sistemas de informação
 - Sistemas Técnicos
 - Sistemas de Tempo Real Integrados
 - Sistemas Distribuídos

Introdução à UML

► História da UML:



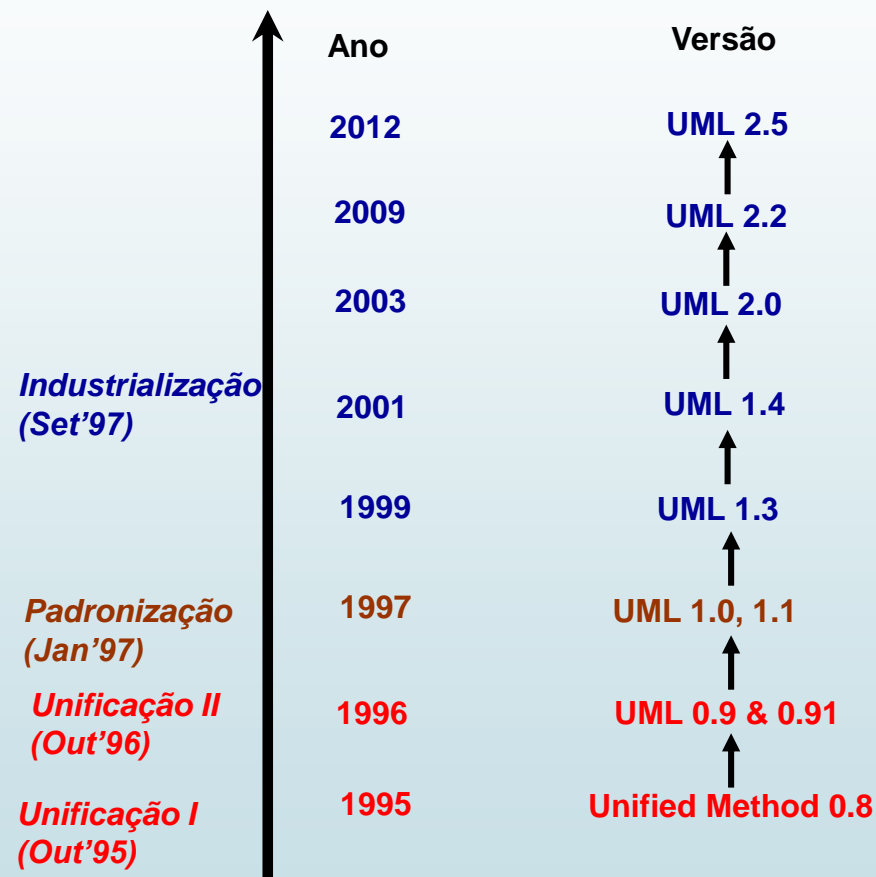
Grady Booch
(Booch)



Ivar Jacobson
(OOSE)



James Rumbaugh
(OMT)

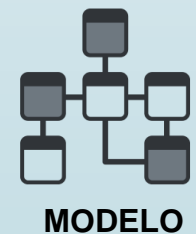


Introdução à UML

► Porque foram criados padrões de modelagem?



Não existe uma notação padronizada e realmente eficaz que abranja qualquer tipo de aplicação que se deseje





UML

► Vantagens:

- Desenvolvimento de programas de forma rápida, eficiente e efetiva;
- Revela a estrutura desejada e o comportamento do sistema;
- Permite a visualização e controle da arquitetura do sistema;
- Melhor entendimento do sistema que está sendo construído e gerenciamento de riscos.

Modelos de projetos

- Mostram os objetos ou classes de um sistema;
- Mostram associações e relacionamentos entre essas entidades;
- São a ponte entre os requisitos do sistema e a implementação de um sistema;
- Modelos dependem do tipo de sistema a ser desenvolvido.
- UML possui 14 modelos diferentes, os quais estão divididos em duas categorias:
 - Estruturais
 - Dinâmicos

Modelos de projetos

Estruturais

- Descrevem as estruturas estáticas do sistema;
- Usa classe de objetos e seus relacionamentos;

Dinâmicos

- Descrevem a estrutura dinâmica do sistema;
- Apresenta as interações entre os objetos do sistema;

Modelos de projetos

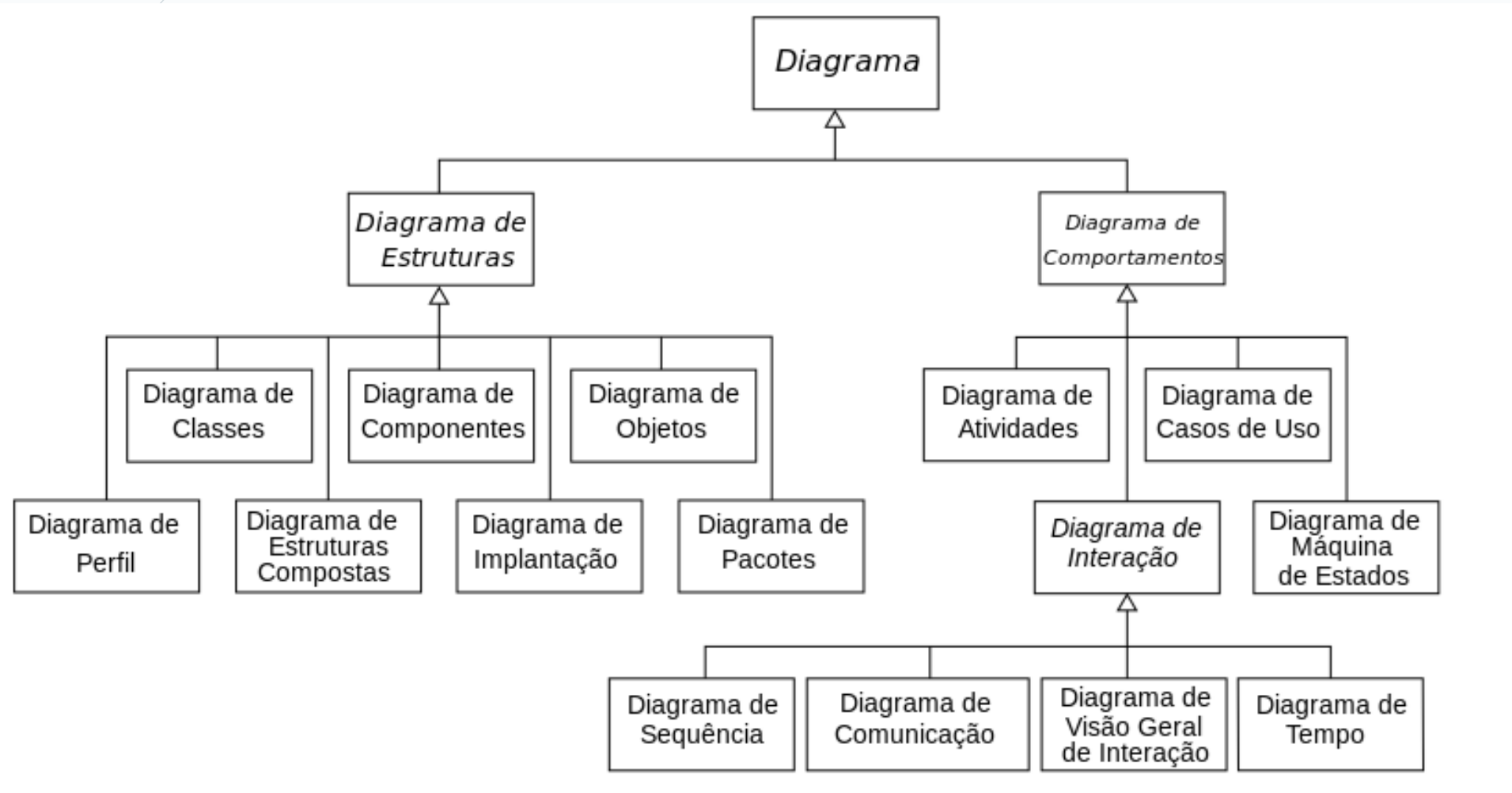


Diagrama de Classe

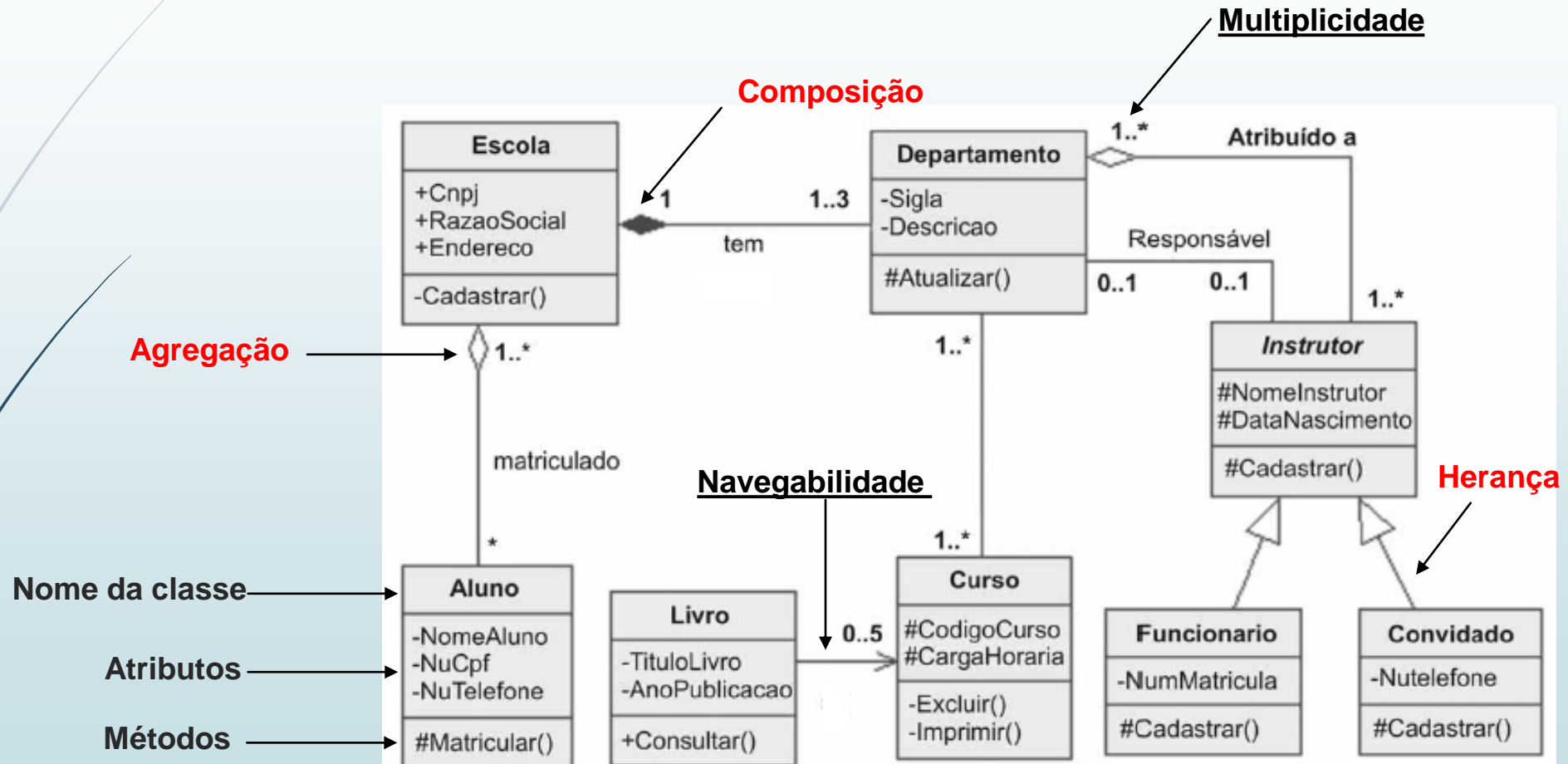


Diagrama de Sequência

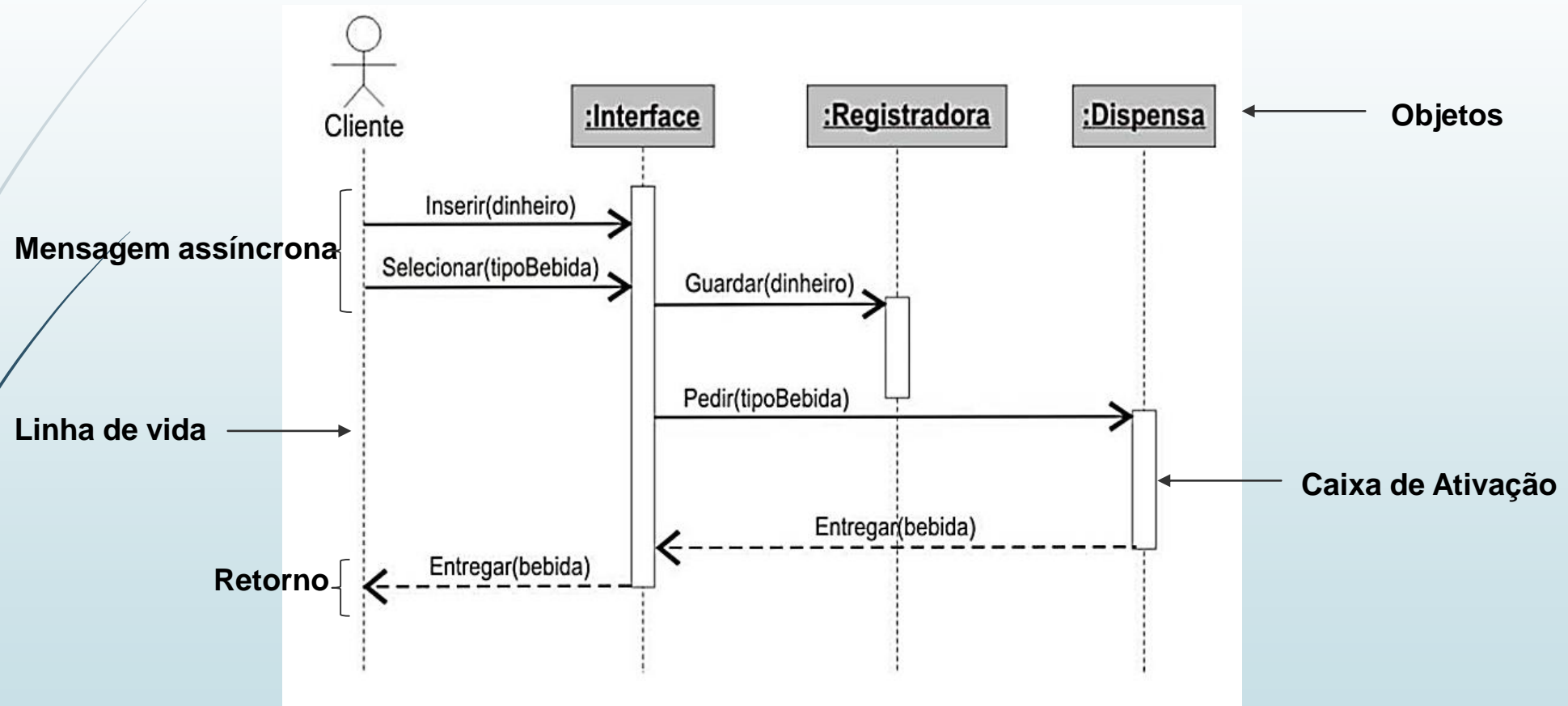


Diagrama de Estado

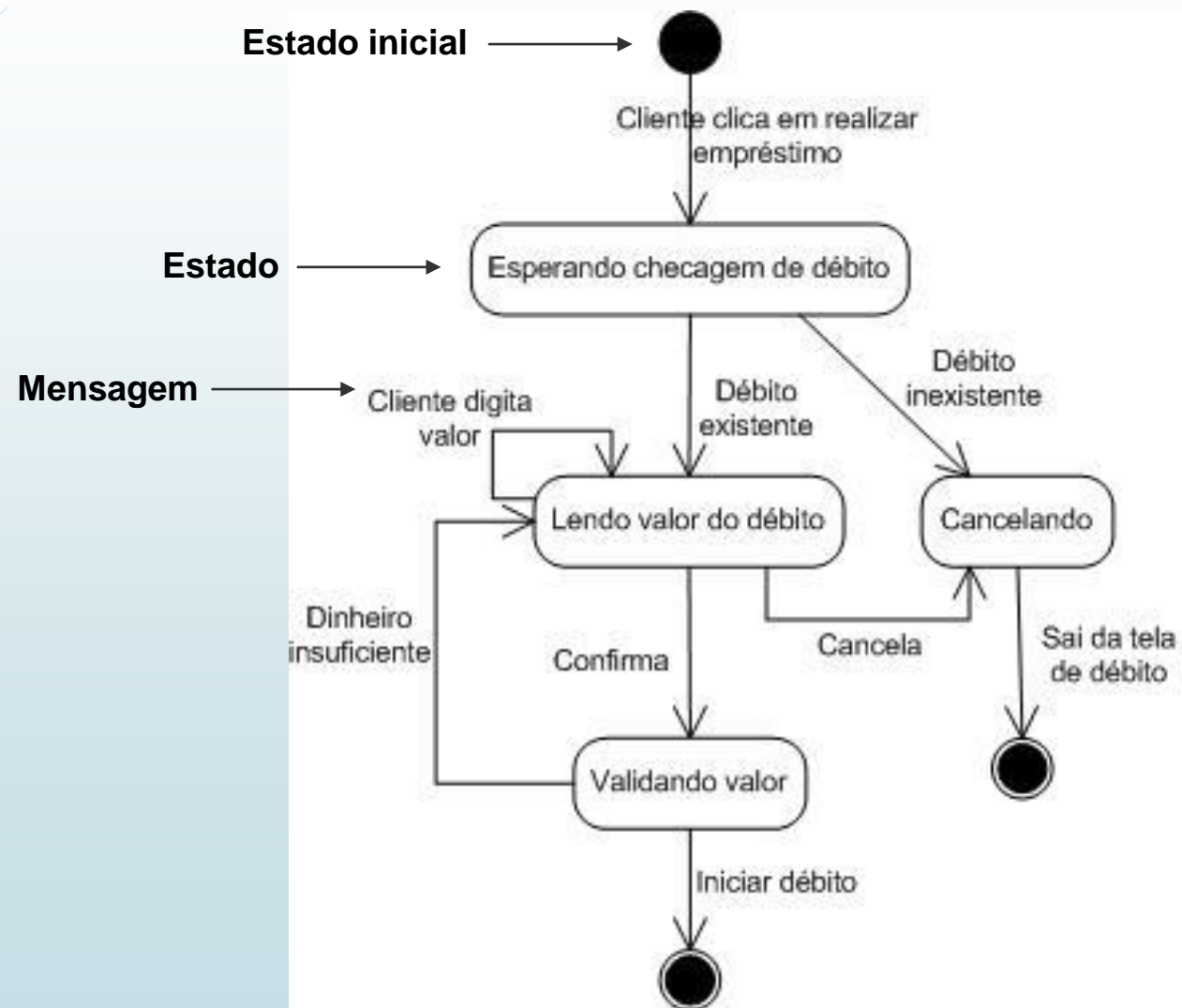


Diagrama de Atividades

