



# MÉTODOS FORMAIS DE ESPECIFICAÇÃO

## PROJETO ORIENTADO A OBJETOS

### INTRODUÇÃO

# Roteiro

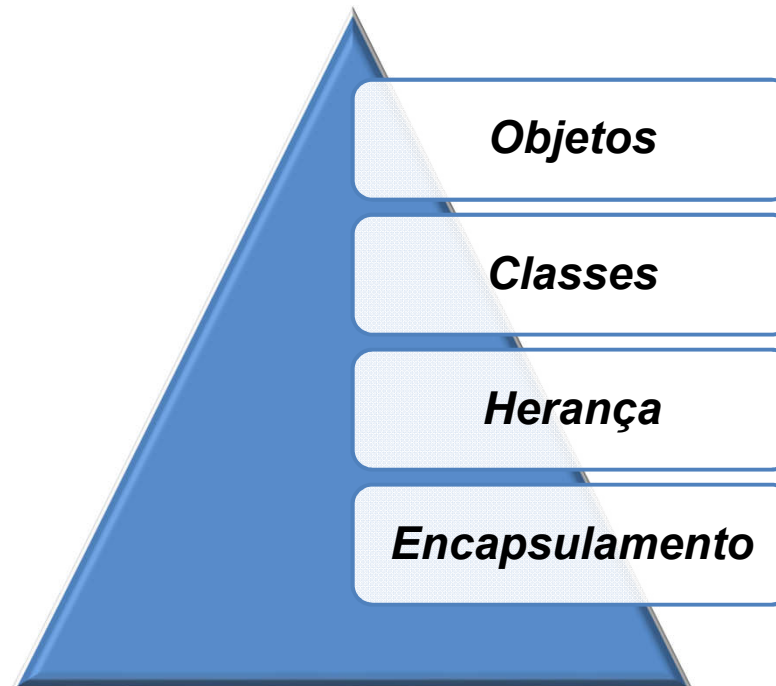
2

- Introdução
- Objetos e Classes
- Processo de especificação orientada a objetos

# Definição

O orientação a objetos significa organizar o mundo real como uma coleção de objetos que incorporam ***estrutura de dados e um conjunto de operações***.

Metodologias orientadas a objetos possuem as características básicas:



# Conceito Projeto Orientado a Objetos

4

**Método na qual a implementação é organizada como um conjunto de **classes** e **objetos** cooperantes, sendo cada objeto é representante de uma determinada **classe**, e as classes são organizadas através de uma relação de **herança**.**

# Projeto Orientado a Objetos

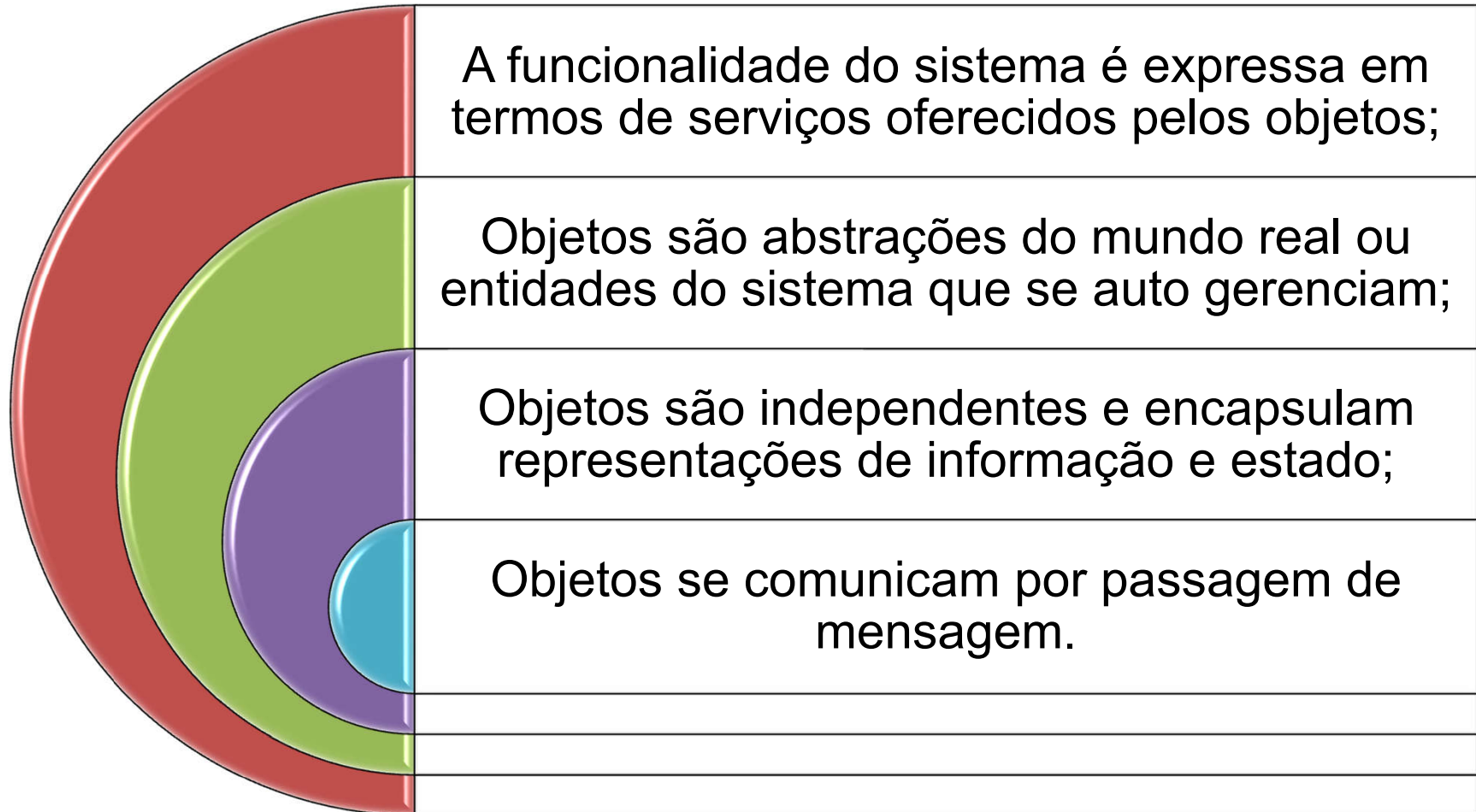
5

Objetivo:

Projetar sistemas usando objetos auto-contidos e classes de objetos.

# Características

6



# Estratégia Orientadas a Objetos

7

## Análise orientada a objetos

- Modelo de objetos do domínio da aplicação para **identificar requisitos**.

## Projeto orientado a objetos

- Modelo orientado a objetos de sistema para **implementar requisitos**.

## Programação orientada a objetos

- Ênfase no projeto usando **linguagem de programação**, como Java ou C++, para implementação.

Análise Estrutural

X

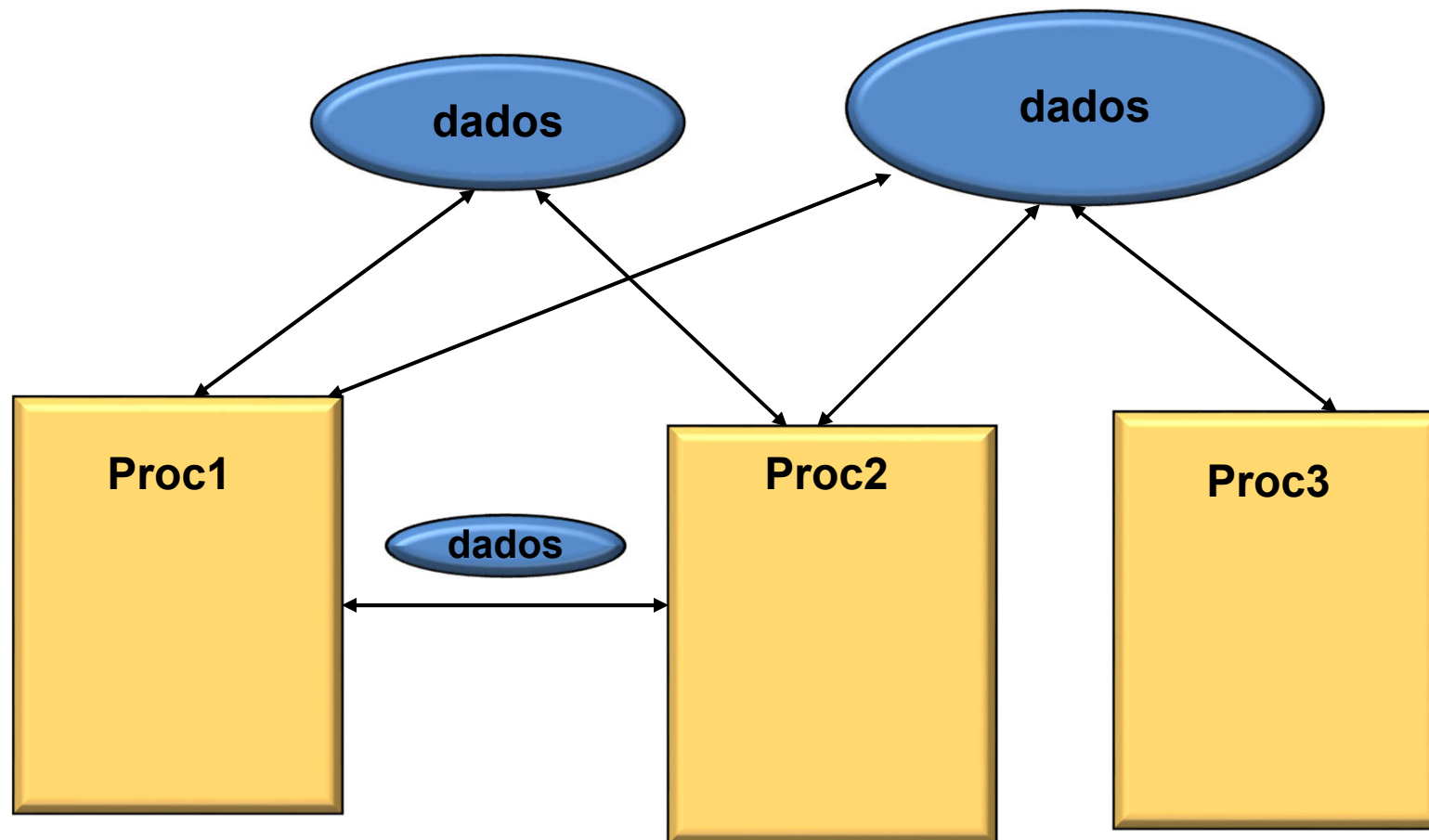
Análise Orientada a Objeto



# Análise Estrutural

## Programação Procedimental

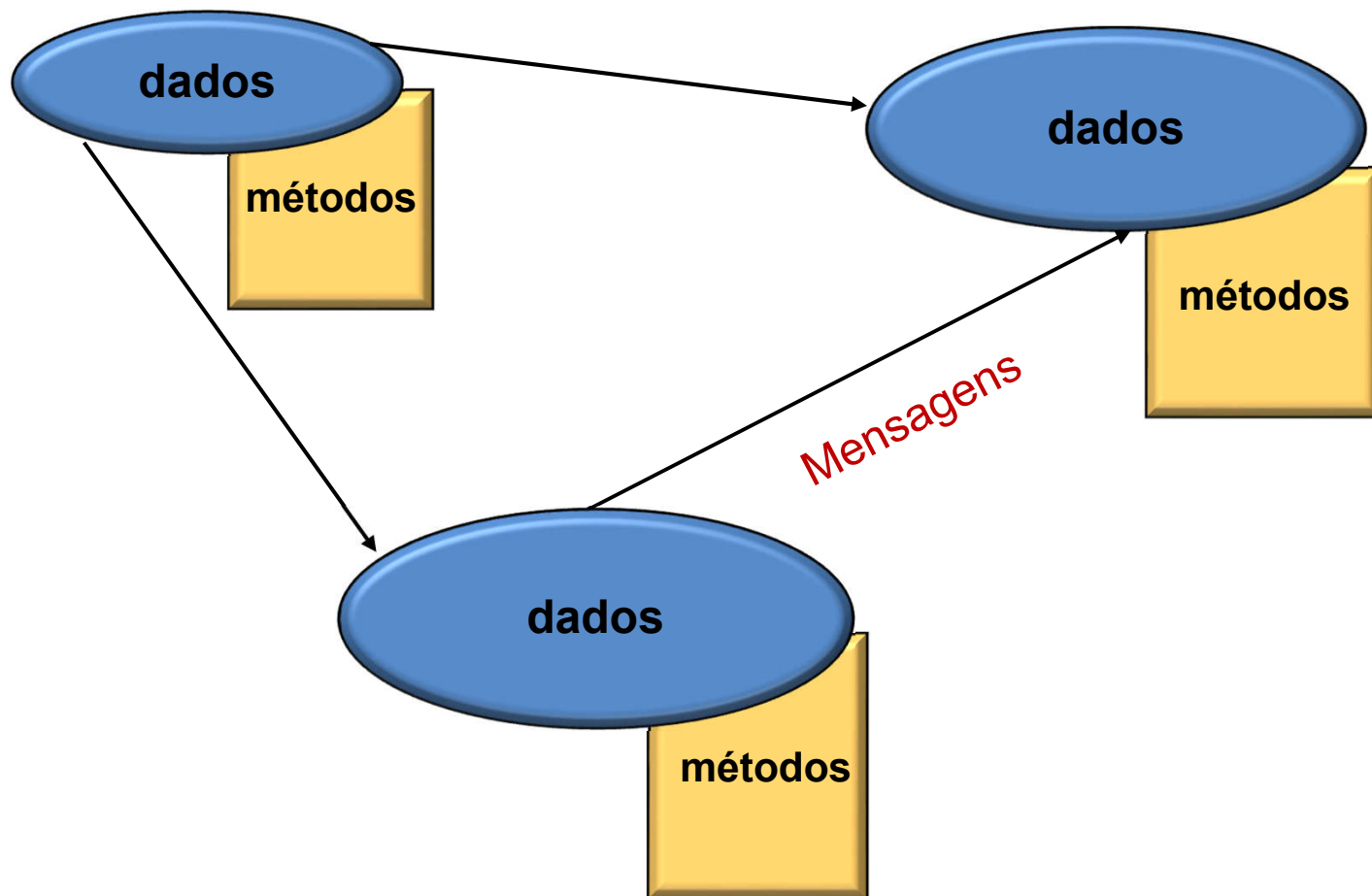
9



# Análise Orientada a Objetos

## Programação Orienta a Objetos

10



# Desenvolvimento Orientado a Objetos

11

A transição desses estágios de desenvolvimento deve ser contínua e direta;

Utilizar o mesmo padrão de processo e notação nos estágios;

Para ir ao próximo estágio deve-se aprimorar o estágio anterior

# Classes e Objeto

12



Classe

- **modelo estático** que permite especificar um conjunto de características e comportamentos



Objeto

- **entidade dinâmica** criada a partir de uma classe e possui os dados sobre os quais são realizadas as operações disponíveis em sua classe.

# Classes e Objetos

## Representação em UML



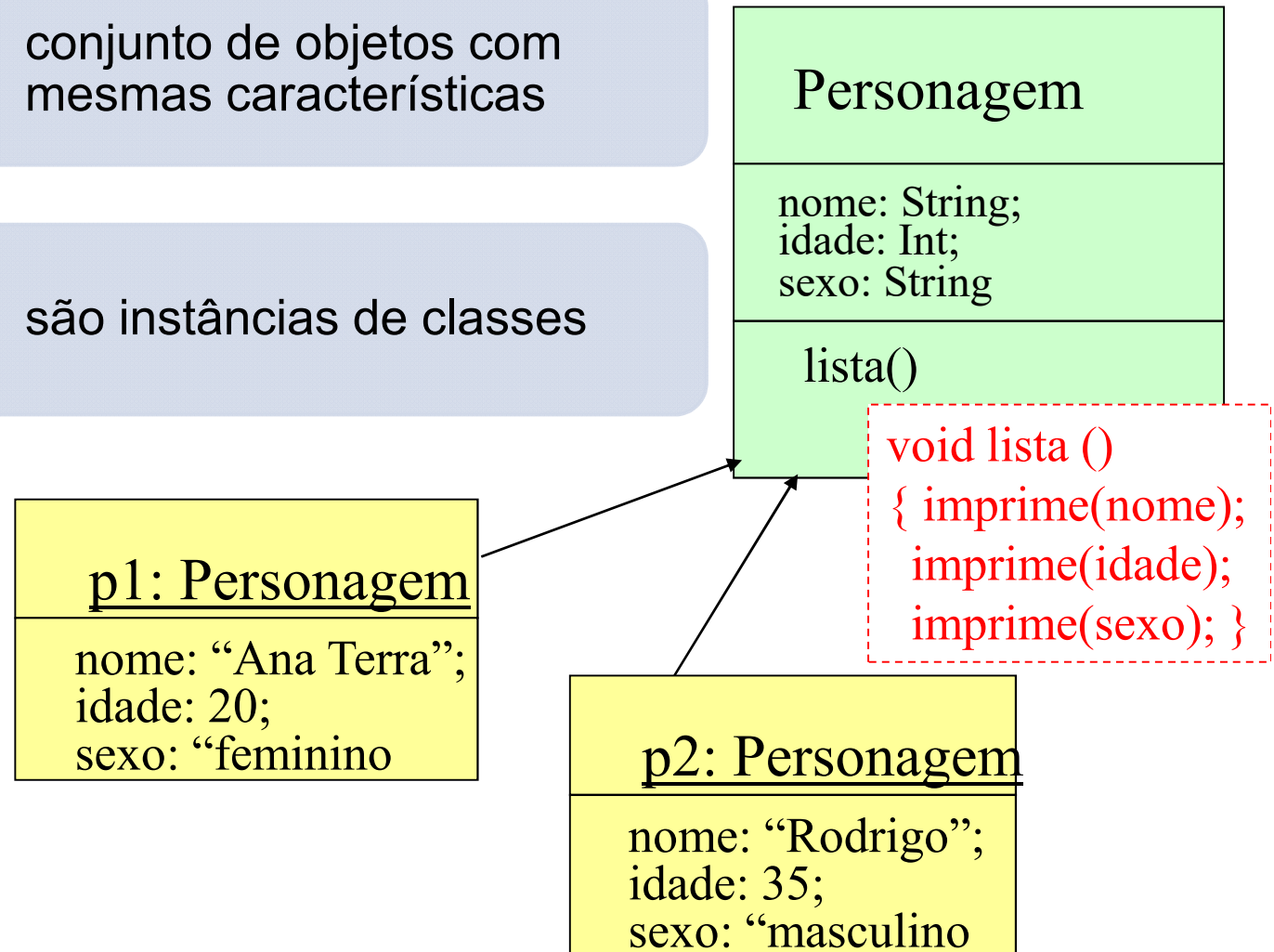
13

classe

- conjunto de objetos com mesmas características

objetos

- são instâncias de classes



# Fluxo do processo da engenharia de requisitos

## Matriz CRUD

The screenshot shows the astah software interface with a project named 'Sample'. The left sidebar displays a tree view of the project structure, including folders for Activity, Class, CompositeStru, CRUD, DataFlow, and Deployment. The main workspace displays a 'CRUD / CRUD [CRUD]' matrix. A 'Set Value' dialog box is open, allowing the user to assign keyboard shortcuts to the matrix cells.

	ER Diagram(EI)	Customer	Order	OrderDetail	Product	Total
UseCase Diagram	CR	CR	CR	CR	R	CR
Place Order	CR	R	C	C	R	CR
Supply Customer Data	C	C				C
Order Product	R				R	R
Arrange Payment	U	R	R	R	R	R
Request Catalog	R	R			R	
Total	CR	CR	CR	CR	R	CR

The 'Set Value' dialog box contains the following options:

- ☒ C (Alt+C)
- ☐ R (Alt+R)
- ☐ U (Alt+U)
- ☐ D (Alt+D)

Buttons: Select All, OK, Cancel

# MVC {get, set}; em C#

```
public class Genre {  
    public string Name { get; set; }  
}
```

Equivalente a:

```
private string name;  
  
public string Name {  
    get { return this.name; }  
    set { this.name = value; }  
}
```

# Em camadas

## Exemplo 1

```
using System;
```

```
namespace PropriedadesAutoImplementadas
```

```
{
```

```
    class Program
```

```
    {
```

```
        public class Cliente
```

```
        {
```

```
            //propriedade ID - somente leitura
```

```
            public int ID { get; private set; }
```

```
            public string Nome { get; set; }
```

```
            public string Email { get; set; }
```

```
            public Cliente(int id){
```

```
                ID = id;
```

```
            }
```

```
            public override string ToString(){
```

```
                return ID + "\t" + Nome + "\t" + Email;
```

```
            }
```

```
        }
```

```
        static void Main(string[] args){
```

```
            Cliente p = new Cliente(1);
```

```
            //Modificando uma propriedade
```

```
            p.Nome = "UESC";
```

```
            p.Email = "colcic@uesc.br";
```

```
            Console.WriteLine(p);
```

```
            Console.ReadLine(); }
```

```
        }
```

```
    }
```



# Exemplo 2

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

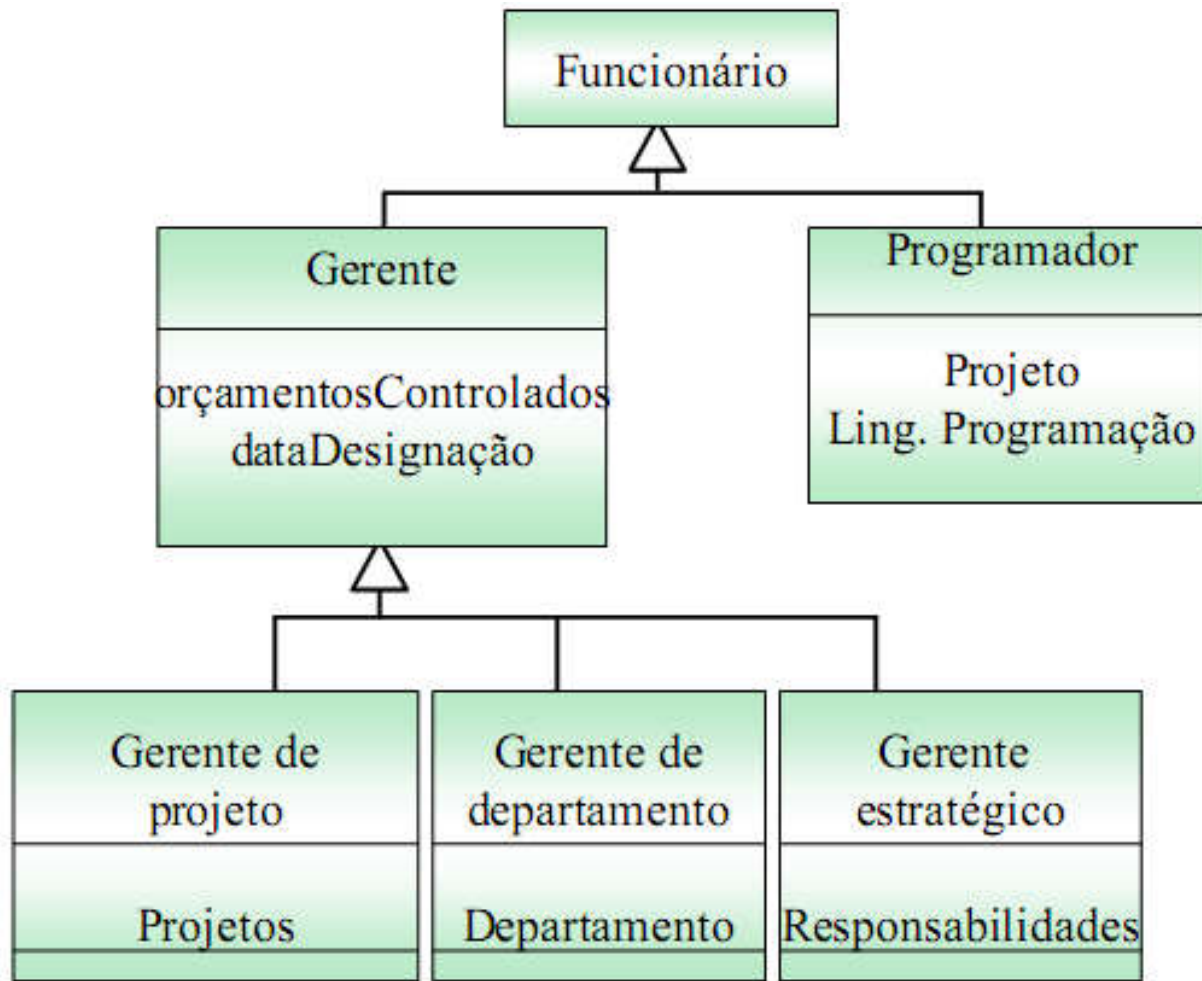
namespace Teste_P00
{
    class Produto
    {
        public string Descricao { get; set; }
        public string Localizacao { get; set; }
        public string ObtemNome()
        {
            return Descricao + " " + Localizacao;
        }
    }

    private void button2_Click(object sender, EventArgs e)
    {
        Produto prod = new Produto();
        prod.Descricao = "Monitor video...";
        prod.Localizacao = "Lab 16B";

        string nomeproduto = prod.ObtemNome();
        textBox2.Text = nomeproduto;
    }
}
```

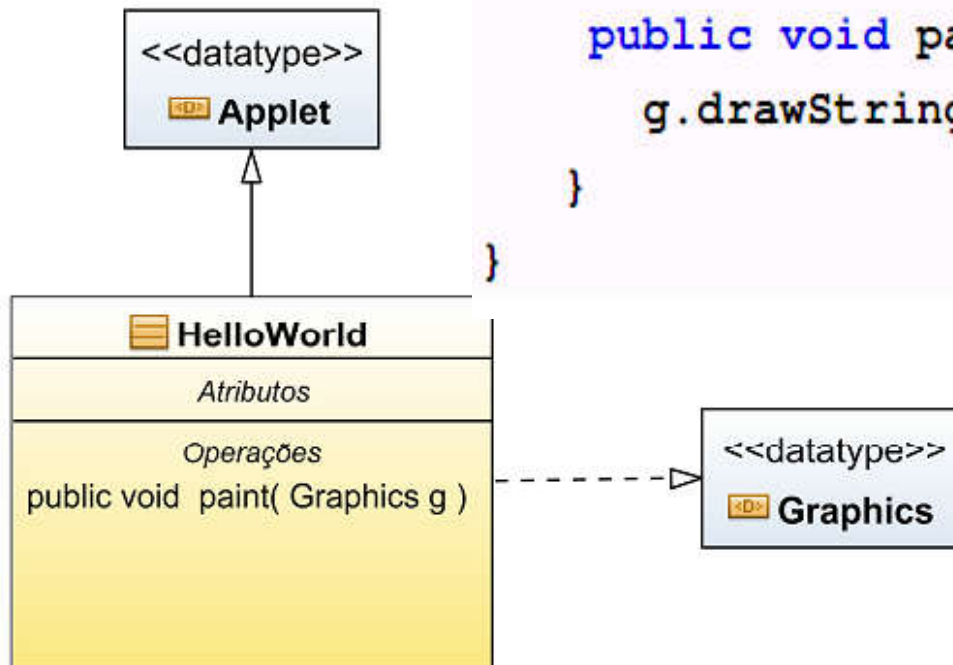
# Generalização e Herança

18



# Herança

19



```
import java.awt.Graphics;

class HelloWorld extends java.applet.Applet {
    public void paint(Graphics g){
        g.drawString("Hello World", 10, 10);
    }
}
```

# Vantagens da Herança

20

É um mecanismo de abstração que pode ser usado para classificar entidades.

É um mecanismo de reutilização tanto a nível de projeto quando de programação.

O grafo de herança organiza o conhecimento sobre o domínio e os sistemas.

# Associação entre Objetos

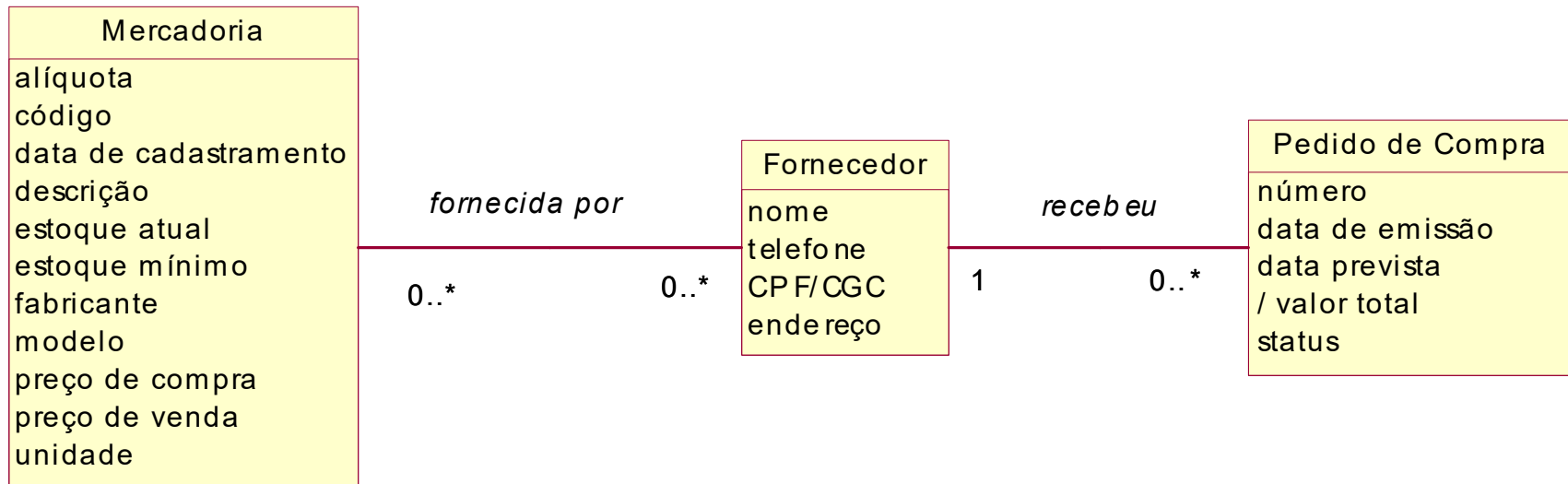
21

- ❑ Objetos e classes de objeto participam de relacionamentos com outros objetos e classes de objeto.
- ❑ Na UML, um relacionamento generalizado é indicado por uma associação.
- ❑ Associações são gerais, mas podem indicar que o atributo de um objeto é um objeto associado, ou que um método conta com um objeto associado.

# Associação entre Objetos

22

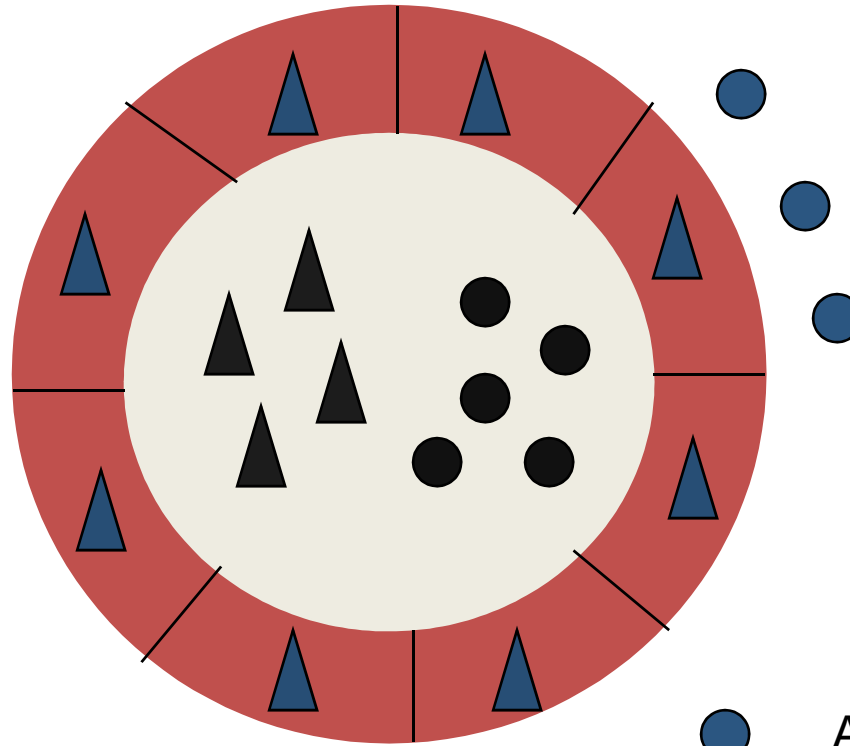
- A multiplicidade permite identificar quantos objetos podem participar de um dado relacionamento.



# Encapsulamento em uma classe

(Ref. Dominando o Java, Makron)

23



Métodos públicos



Métodos privados



Atributos públicos



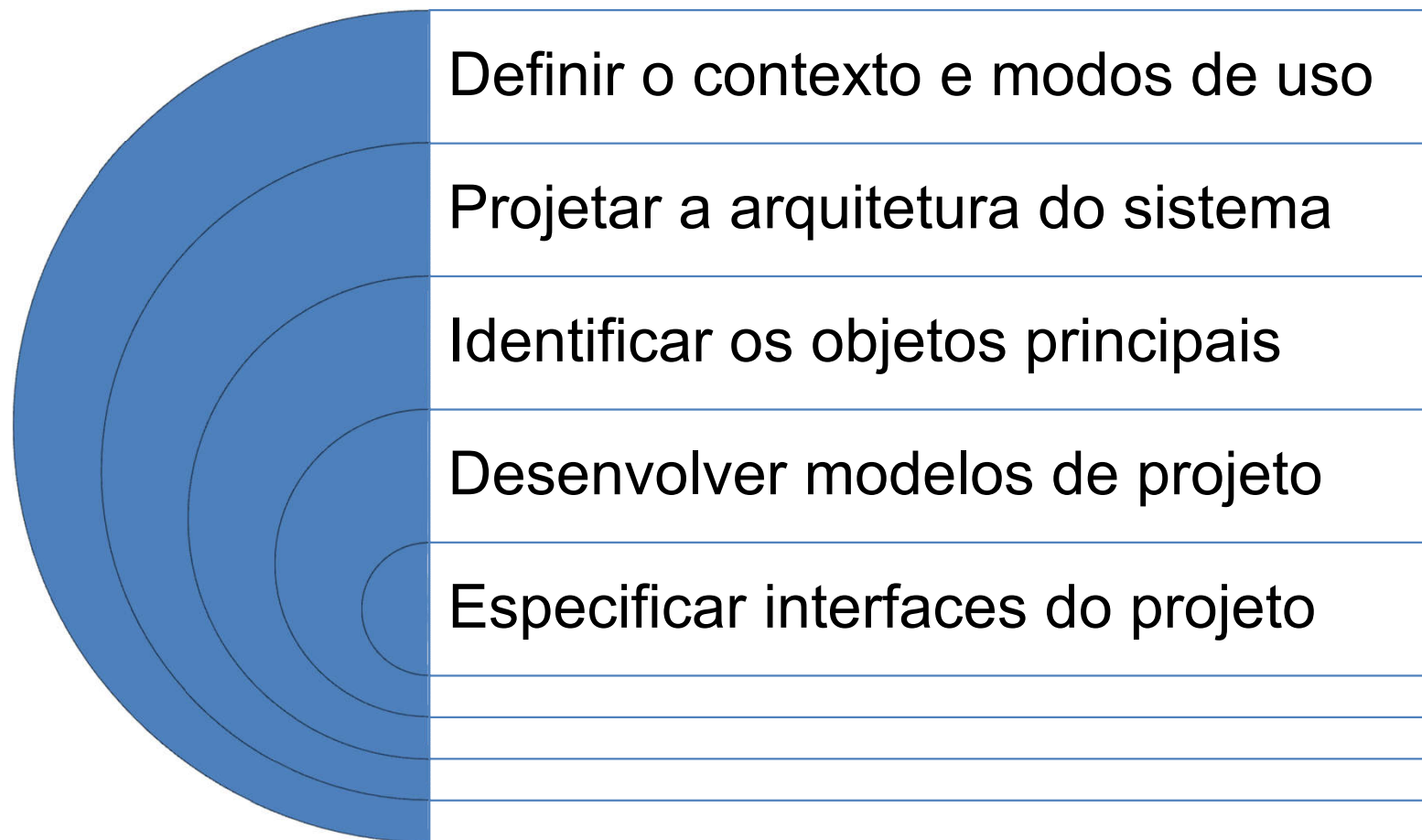
Atributos privados

# O Processo de Especificação



# Desenvolvimento Orientado a Objetos - Processo Geral

25



# Definir o contexto e modos de uso

26

## Contexto de sistema

- modelo estático

## Modelos de uso do sistema

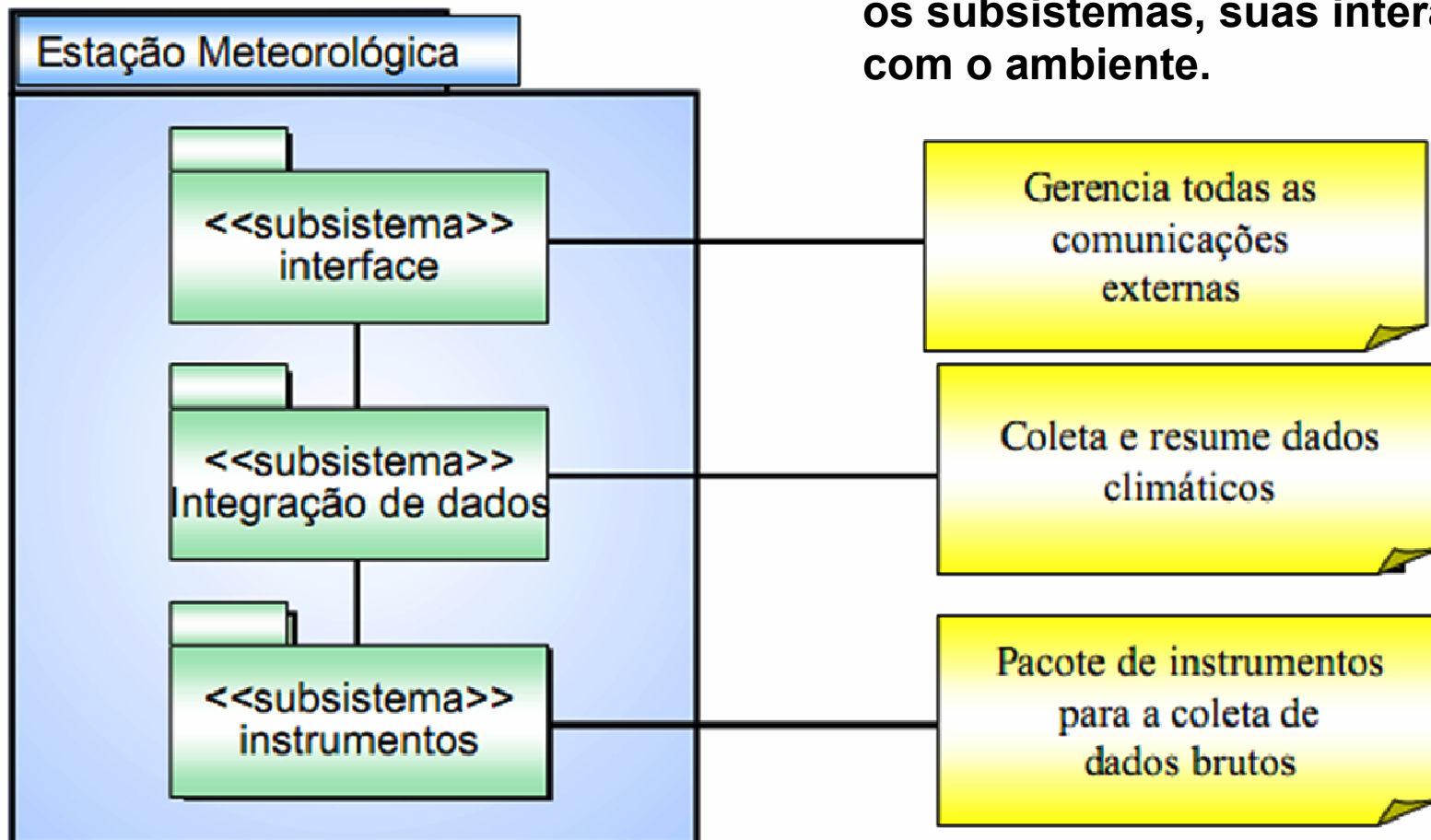
- modelo dinâmico

# Modelo de Contexto



27

**Modelo de contexto do sistema representa as associações entre os subsistemas, suas interações com o ambiente.**



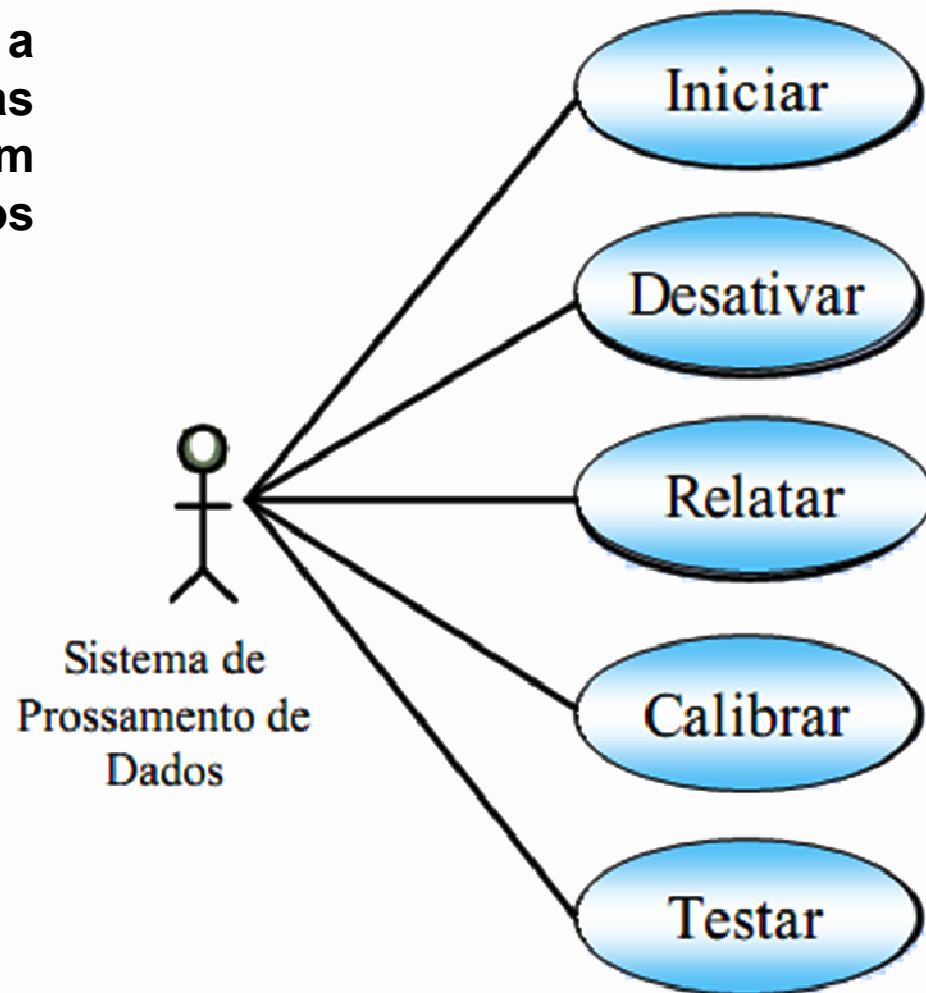
(Sommerville, 2006)

# Modelo de Uso do Sistema



28

**Modelo de uso mostra a interação entre o sistema e as entidades externas, que podem ser operadores ou outros sistemas conectados.**



(Sommerville, 2006)

# Caso de Uso - Descrições

29

- É preciso desenvolver descrições para todos os casos de uso representados no modelo de caso de uso.
- Utilidade de casos de uso
- Identificar objetos no sistema
- Identificar operações no sistema

# Formulário - Casos de Uso

30

Sistema	Estação Meteorológica
Use-case	Relatar
Agentes	Sistema de processamento de dados sobre o clima, Estação meteorológica.
Dados	A estação meteorológica envia para o sistema de processamento de dados climáticos um resumo de dados sobre o clima, que foram coletados a partir de instrumentos, no período de coleta. Os dados enviados referem-se às temperaturas máximas, mínimas e médias do solo e do ar; à pressão máxima, mínima e média do ar; às velocidades máxima, mínima e média do vento, conforme amostragem a cada intervalo de cinco minutos
Estímulo	O sistema de processamento de dados sobre o clima estabelece um link de modem com a estação meteorológica e requisita a transmissão dos dados
Resposta	Os dados resumidos pelo sistema de coleta de dados sobre o clima são enviados ao sistema de processamento de dados.
Comentários	Em geral, as estações meteorológicas recebem um pedido de relatório por hora, mas essa frequência pode diferir de uma estação para outra a ser modificada no futuro.

Slide 27

# Projeto de Arquitetura

31

Definidas as interações entre o sistema e o seu ambiente, utiliza-se essas informações para estabelecer a arquitetura do sistema.

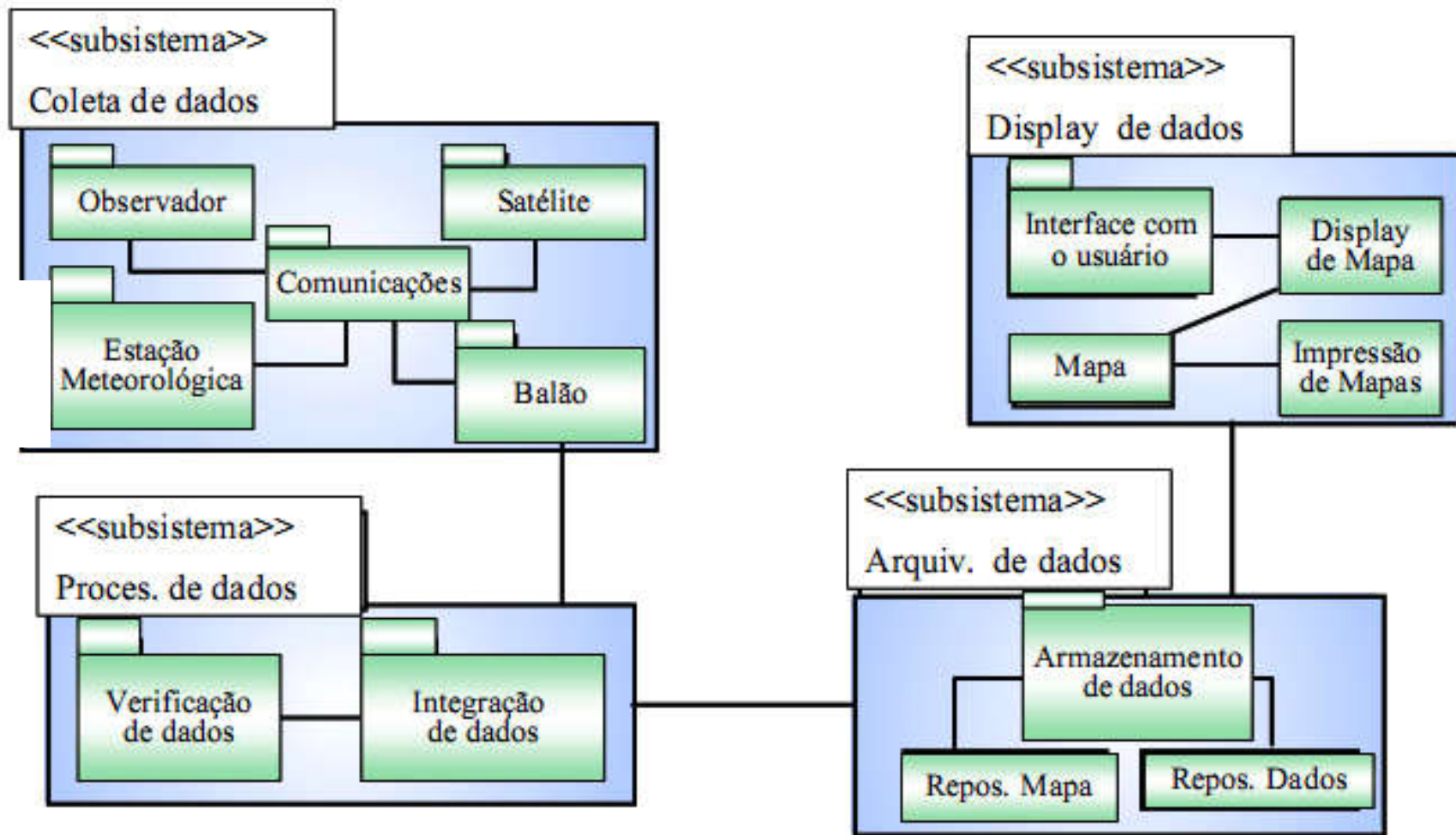
Uma arquitetura em camadas é apropriada e permite a decomposição em subsistemas.

Utiliza-se diagrama de pacotes para representar arquitetura do sistema.



# Diagrama de Componentes

32





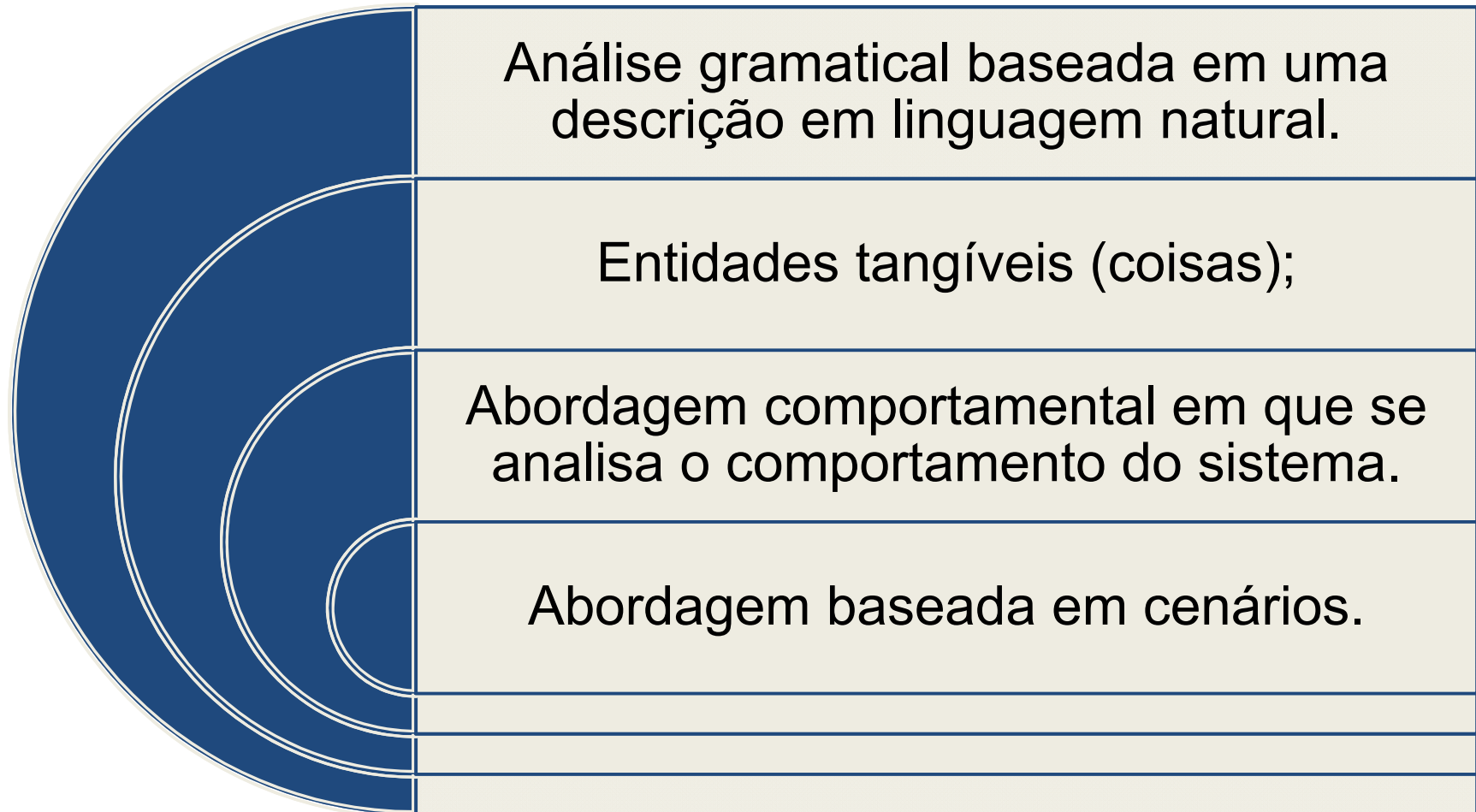
# Identificação dos Objetos

33

- A identificação de objeto é um processo iterativo. É improvável que se obtenha todos os objetos num primeiro esboço.
- Nesse estágio os objetos essenciais do sistema já foram levantados na etapa de análise.
- Na etapa de projeto, refina-se os objetos identificados na análise e define-se outros objetos relevantes na solução do problema.

# Identificação das Classes

34



# Identificação de Classes

## Análise Gramatical

35

EstaçãoMeteorológica
Identificador
RelatarClima() Calibrar(instrumentos) testar() iniciar(instrumentos) desativar(instrumentos)

DadosMeteorológicos
TemperaturasdoAr TemperaturasdoSolo VelocidadesdoVento DireçõesdoVento Pressões precipitação
Coletar() Resumir()

Termômetro de solo
temperatura
Testar() calibrar()

Anemômetro
velocidadedoVento direçõesdoVento
Testar()

Barômetro
Pressão altura
Testar() Calibrar()

Um sistema de mapeamento meteorológico é necessário para gerar mapas meteorológicos regularmente, utilizando **dados coletados** a partir de **estações meteorológicas** remotas, sem que seus funcionários estejam presentes, e de outras fontes de dados, como **observadores de tempo, balões e satélites meteorológicos**.

# Modelos de Projeto

36

## Modelos de subsistema

- Diagrama de classes

## Modelos de sequência

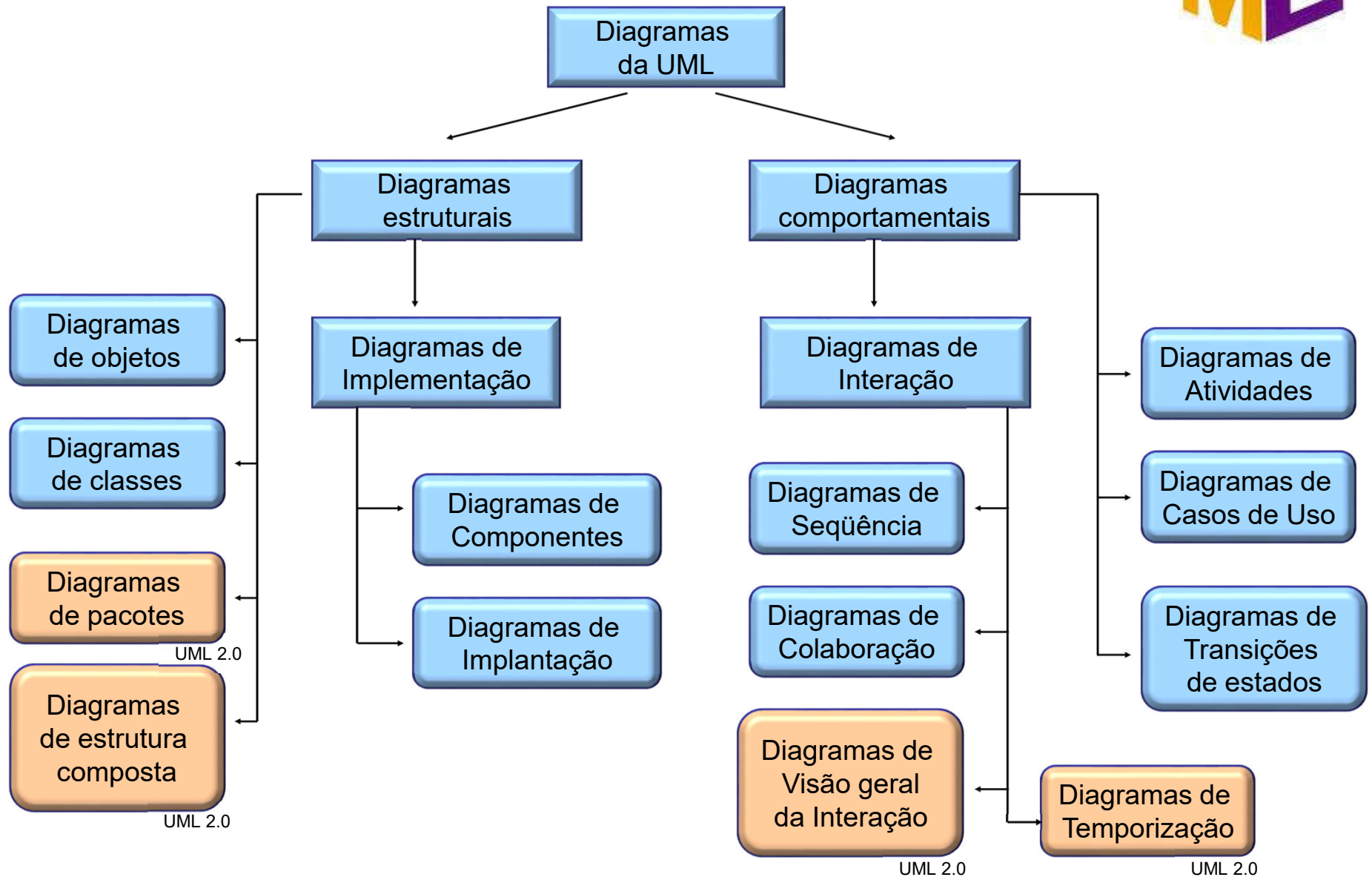
- Diagrama de sequência

## Modelos de máquina de estado

- Diagrama gráficos de estado

Mais modelos UML ...

# Diagrama UML



# Modelos de Sequência

38

- ❑ Modelo de sequência mostra a seqüência de interações (mensagens) entre os objetos para a realização de uma operação do sistema.
- ❖ Os objetos envolvidos na operação são organizados horizontalmente, com uma linha vertical ligada a cada objeto.
- ❖ O tempo é representado verticalmente, assim os modelos são lido de cima para baixo.
- ❖ Interações entre objetos são representadas por setas rotuladas. As setas representam mensagens ou eventos, que são fundamentais para a interação.
- ❖ Um retângulo estreito na linha de um objeto representa o tempo pelo qual o objeto é o objeto controlador (ativo) no sistema.

# Diagrama de Sequência



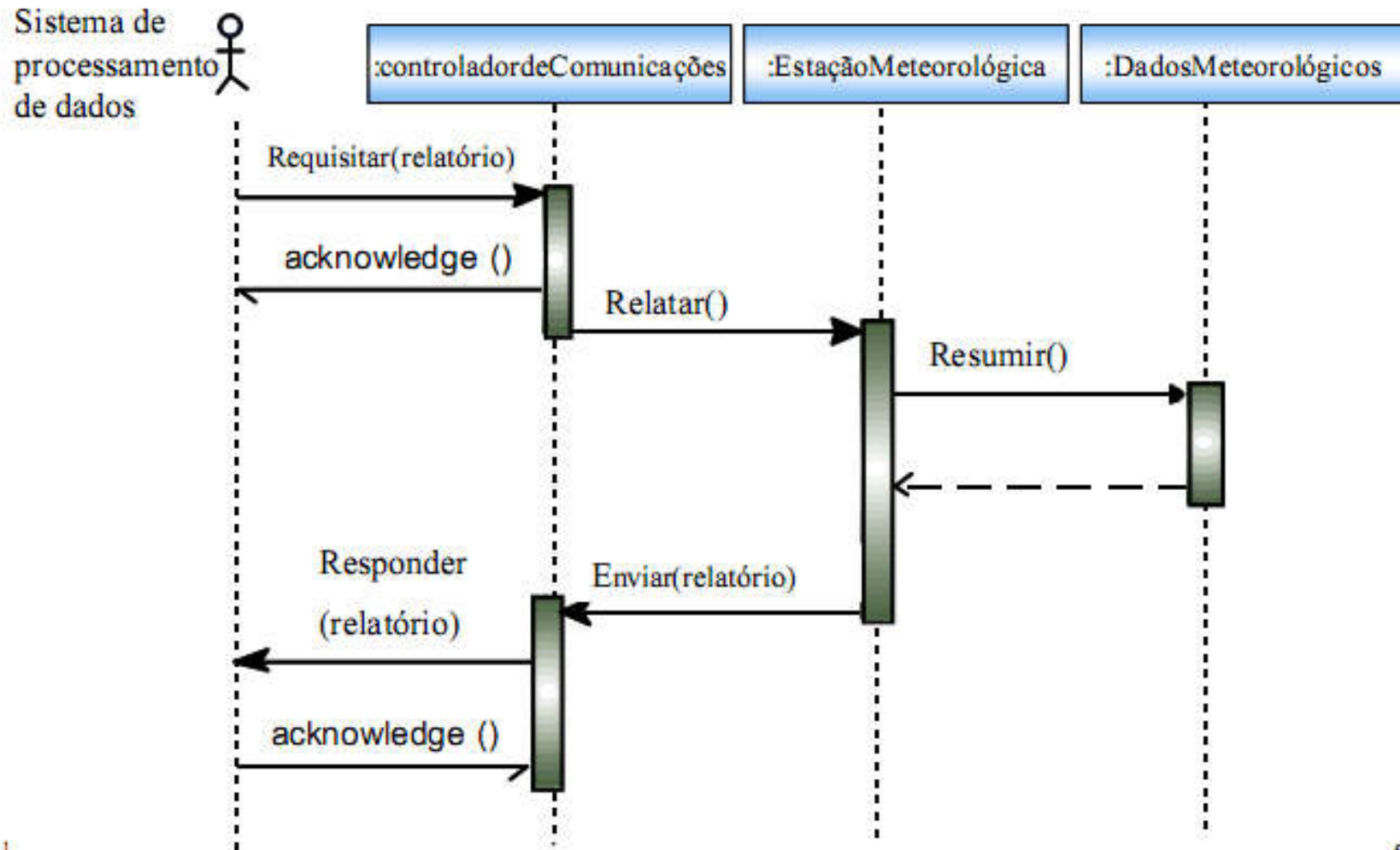
39

- É preciso produzir um diagrama de seqüência para cada interação significativa (cada operação do sistema);
- Deve haver um diagrama de seqüência para cada caso de uso identificado;
- DS é usado para modelar o comportamento combinado em um grupo de objetos.

# Diagrama de Sequência



40





# Modelos de Estado

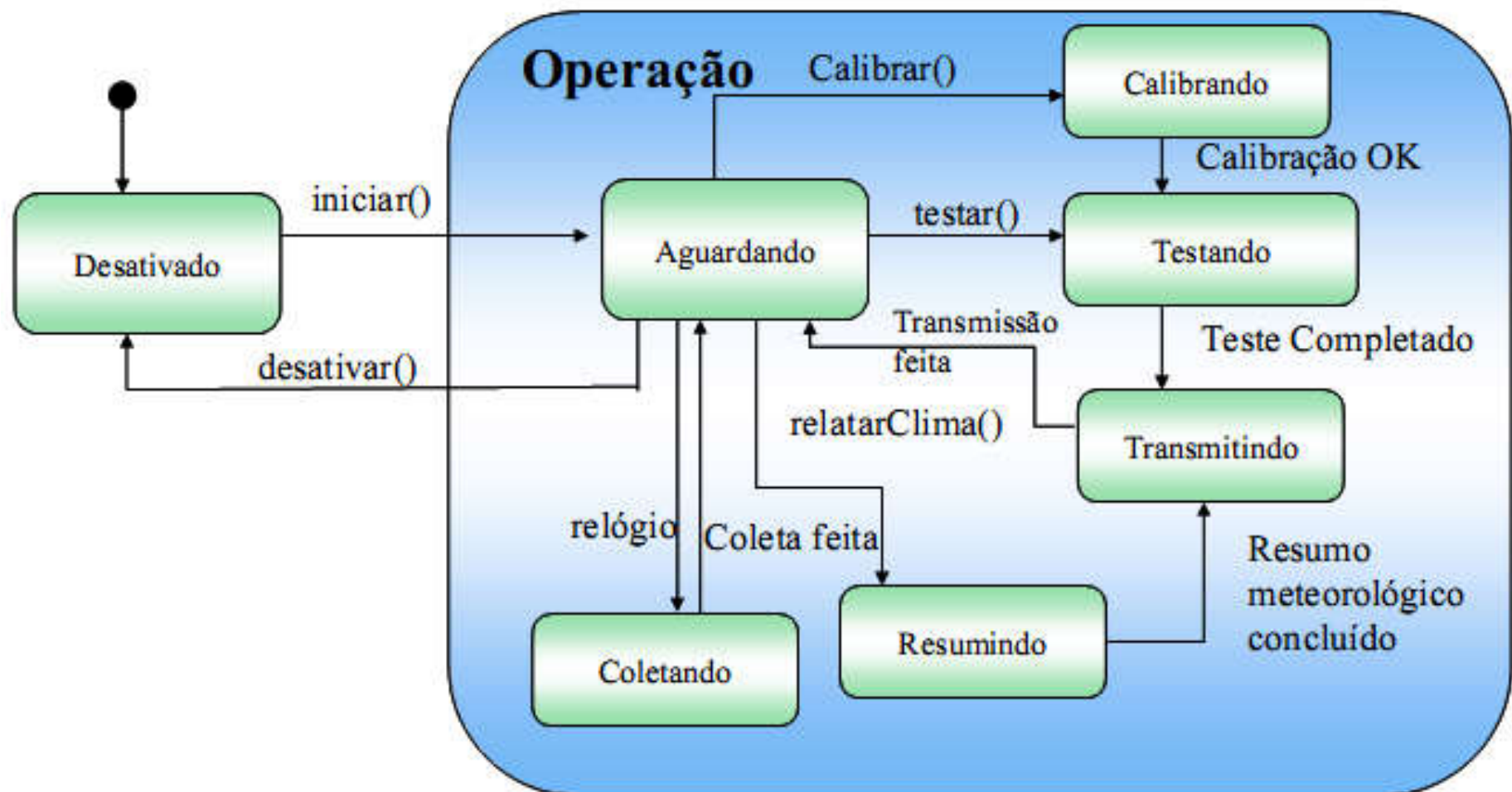
41

- Através de uma máquina de estados (statecharts) pode-se mostrar o comportamento de um único objeto em resposta a diferentes mensagens que ele pode processar;
- Basicamente, o modelo de máquina de estados mostra como o objeto muda de estado, dependendo das mensagens que ele recebe;
- De modo geral, não é normalmente necessário produzir um *statechart* para todos os objetos definidos.

# Diagrama de Estado



42



# Especificação de interface entre objetos

43

- ❑ Interfaces são os serviços que os objetos oferecem a outros objetos;
- ❑ Após o desenvolvimento dos diagramas de sequência para todas as operações do sistema, faz-se uma análise de cada objeto presente nesses diagramas;
- ❑ Toda mensagem recebida pelo objeto é um serviço que ele deve oferecer, e portanto faz parte de sua interface.

# Especificação de interface entre objetos

44

- É a especificação dos detalhes da interface para um objeto ou um grupo de objetos.
- Significa definição das assinaturas e a semântica definida pelos serviços oferecidos pelos objetos.
- Facilita o desenvolvimento em paralelo

# Referências

BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivan. **UML: guia do usuário**. Rio de janeiro: Campus, 2000. 472p.

LARMAN, Craig. Utilizando UML e padrões: uma introdução à análise e ao projeto orientado a objetos. Bookman, Porto Alegre, 2000.

VIANA, Thiago Affonso de M. N. Viana. Apresentação (slide), Ufpe. 2009.