



Instituto Federal de Educação, Ciência e Tecnologia da Bahia - IFBA  
Departamento de Ciência da Computação  
Tecnólogo em Análise e Desenvolvimento de Sistemas

## Projeto e implementação de software

André L. R. Madureira <[andre.madureira@ifba.edu.br](mailto:andre.madureira@ifba.edu.br)>  
Doutorando em Ciência da Computação (UFBA)  
Mestre em Ciência da Computação (UFBA)  
Engenheiro da Computação (UFBA)

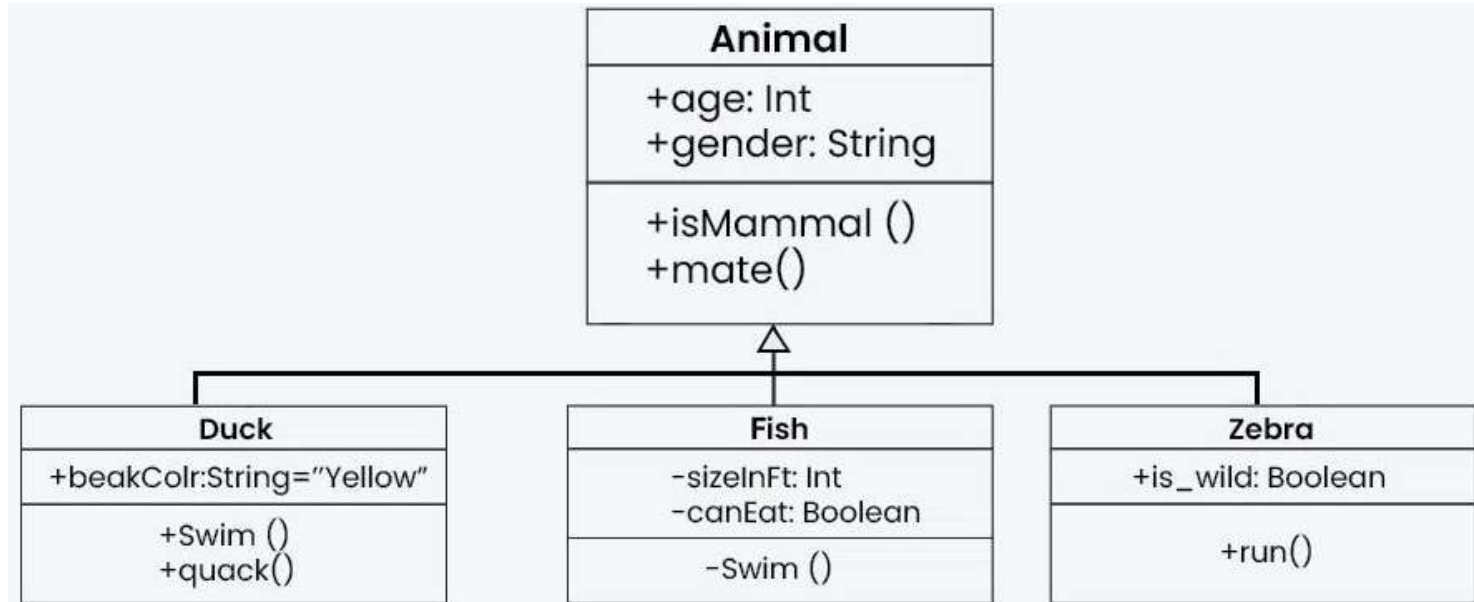
# Projeto e implementação de software

---

- Estágio do processo no qual um sistema de software executável é desenvolvido
  - **Projeto:** *“como resolver um problema”*
  - **Implementação:** *“como concretizar a solução do problema”*
- Existem varias técnicas de projeto de sistemas, cada uma mais adequada para um paradigma de programação específico
  - **Ex:** Diagramas de Classes UML (orientação a objetos), Diagramas de eventos UML (orientação a eventos)

# Projeto orientado a objetos com UML

- Envolve o projeto de sistemas baseados em **classes**, **objetos** e os **relacionamentos** entre essas classes



# Projeto orientado a objetos com UML

---

- **Classes:** definem os objetos interativos do sistema
  - Inclui propriedades (**dados**) e métodos (**funções**)
- **Objetos:** definem instancias de classe, com estados locais
  - Cada objeto possui um **estado local**
    - **Estado:** dados armazenados nas propriedades (atributos) do objeto, em um dado instante de tempo
  - Objetos são criados dinamicamente a partir das classes

# Porque orientação à objetos?

---

- Os objetos podem ser entendidos e modificados como **entidades autônomas**
  - Objetos incluem os dados e as operações para manipulá-los
  - Alterar a implementação de um objeto ou adicionar serviços não deve afetar outros objetos do sistema
    - **Encapsulamento**: O estado de um objeto é privado e não pode ser acessado diretamente, de fora do objeto

# Porque orientação à objetos?

---

- Muitas vezes existe um mapeamento claro entre entidades do mundo real (como componentes de hardware) e objetos
  - **Ex:** cadeiras, casas, pessoas, etc
- Esse mapeamento
  - Melhora a inteligibilidade do código do sistema
  - Consequentemente, melhora a manutenibilidade do projeto

# Processo de projeto orientado a objetos

---

- Composto pelas seguintes atividades:
  - Definição do contexto e interações externas do sistema
  - Projeto da arquitetura do sistema
  - Identificação dos principais objetos
  - Desenvolvimento dos modelos de projeto
  - Especificação das interfaces

# Definição do contexto e interações do sistema

---

- Esta atividade é responsável por
  - Descrever como oferecer as funcionalidades requeridas
  - Estruturar como o sistema se comunicará com o ambiente externo
  - Estabelecer os **limites do sistema**
    - Quais recursos serão implementados no sistema
    - Quais recursos estão em outros sistemas associados



# Definição do contexto e interações do sistema

---

- Para esta atividade, precisamos de modelos que apresentam visões complementares dos relacionamentos entre o sistema e o ambiente
  - **Modelos de contexto (estruturais ou estáticos)**
    - Modelos estruturais do sistema e do ambiente
  - **Modelos de interação (ou dinâmicos)**
    - Modelos dinâmicos que mostram como o sistema (em execução) interage com seu ambiente

# Exercício

---

Considerando o projeto e implementação de software, marque a alternativa que contém **somente** as assertivas VERDADEIRAS.

I - O projeto orientado a objetos com UML utiliza somente diagramas de classes UML para modelagem do sistema.

II - No projeto orientado a objetos, objetos podem ser entendidos como entidades autônomas.

III - O encapsulamento define que objetos tem estados que podem ser acessados diretamente por outros objetos.

IV - A especificação das interfaces é a ultima atividade do processo de projeto orientado a objetos.

☐ Somente I e III.

☐ Somente II.

☐ Somente II e IV.

☐ Somente IV.

☐ Nenhuma das alternativas anteriores.

# Exercício

Considerando o projeto e implementação de software, marque a alternativa que contém **somente** as assertivas VERDADEIRAS.

I - O projeto orientado a objetos com UML utiliza **somente** diagramas de classes UML para modelagem do sistema. **F**

II - No projeto orientado a objetos, objetos podem ser entendidos como entidades autônomas.

III - O encapsulamento define que objetos tem estados que podem ser acessados diretamente por outros objetos.

IV - A especificação das interfaces é a ultima atividade do processo de projeto orientado a objetos.

- ☐ Somente I e III.
- ☐ Somente II.
- ☐ Somente II e IV.
- ☐ Somente IV.
- ☐ Nenhuma das alternativas anteriores.

# Exercício

Considerando o projeto e implementação de software, marque a alternativa que contém **somente** as assertivas VERDADEIRAS.

I - O projeto orientado a objetos com UML utiliza **somente** diagramas de classes UML para modelagem do sistema. **F**

II - No projeto orientado a objetos, objetos podem ser entendidos como entidades autônomas. **V**

III - O encapsulamento define que objetos tem estados que podem ser acessados diretamente por outros objetos.

IV - A especificação das interfaces é a ultima atividade do processo de projeto orientado a objetos.

- ☐ Somente I e III.
- ☐ Somente II.
- ☐ Somente II e IV.
- ☐ Somente IV.
- ☐ Nenhuma das alternativas anteriores.

# Exercício

Considerando o projeto e implementação de software, marque a alternativa que contém **somente** as assertivas VERDADEIRAS.

I - O projeto orientado a objetos com UML utiliza **somente** diagramas de classes UML para modelagem do sistema. **F**

II - No projeto orientado a objetos, objetos podem ser entendidos como entidades autônomas. **V**

III - O encapsulamento define que objetos tem estados que **podem ser acessados** diretamente por outros objetos. **F**

IV - A especificação das interfaces é a ultima atividade do processo de projeto orientado a objetos.

- ☐ Somente I e III.
- ☐ Somente II.
- ☐ Somente II e IV.
- ☐ Somente IV.
- ☐ Nenhuma das alternativas anteriores.

# Exercício

Considerando o projeto e implementação de software, marque a alternativa que contém **somente** as assertivas VERDADEIRAS.

I - O projeto orientado a objetos com UML utiliza **somente** diagramas de classes UML para modelagem do sistema. **F**

II - No projeto orientado a objetos, objetos podem ser entendidos como entidades autônomas. **V**

III - O encapsulamento define que objetos tem estados que **podem ser acessados** diretamente por outros objetos. **F**

IV - A especificação das interfaces é a ultima atividade do processo de projeto orientado a objetos. **V**

☐ Somente I e III.

☐ Somente II.

☒ Somente II e IV.

☐ Somente IV.

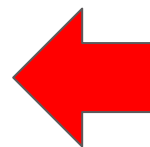
☐ Nenhuma das alternativas anteriores.

# Modelos de Contexto (estruturais ou estáticos)

---

- Descrevem a estrutura de um sistema

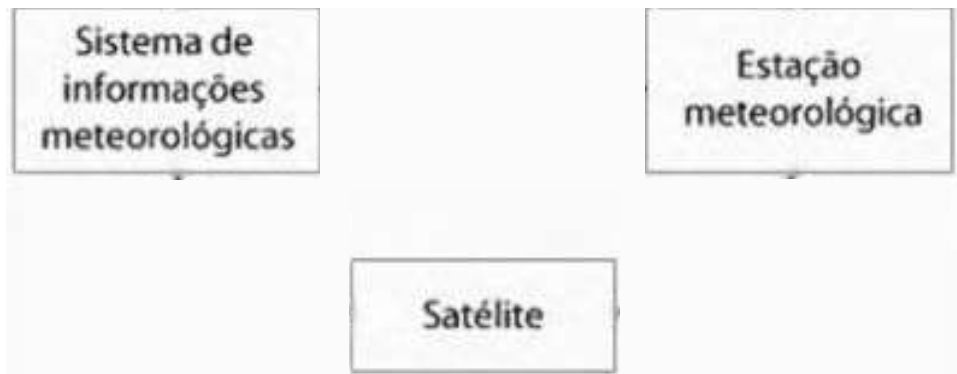
- **Diagrama de Entidade-Relacionamento (E-R)**
- **Diagrama de Classe**
- Modelo de Pacotes
- Outros diagramas



**Mais  
usados**

# Modelo de Entidade-Relacionamento

- Descrevem o sistema através de **entidades** e **associações** entre elas

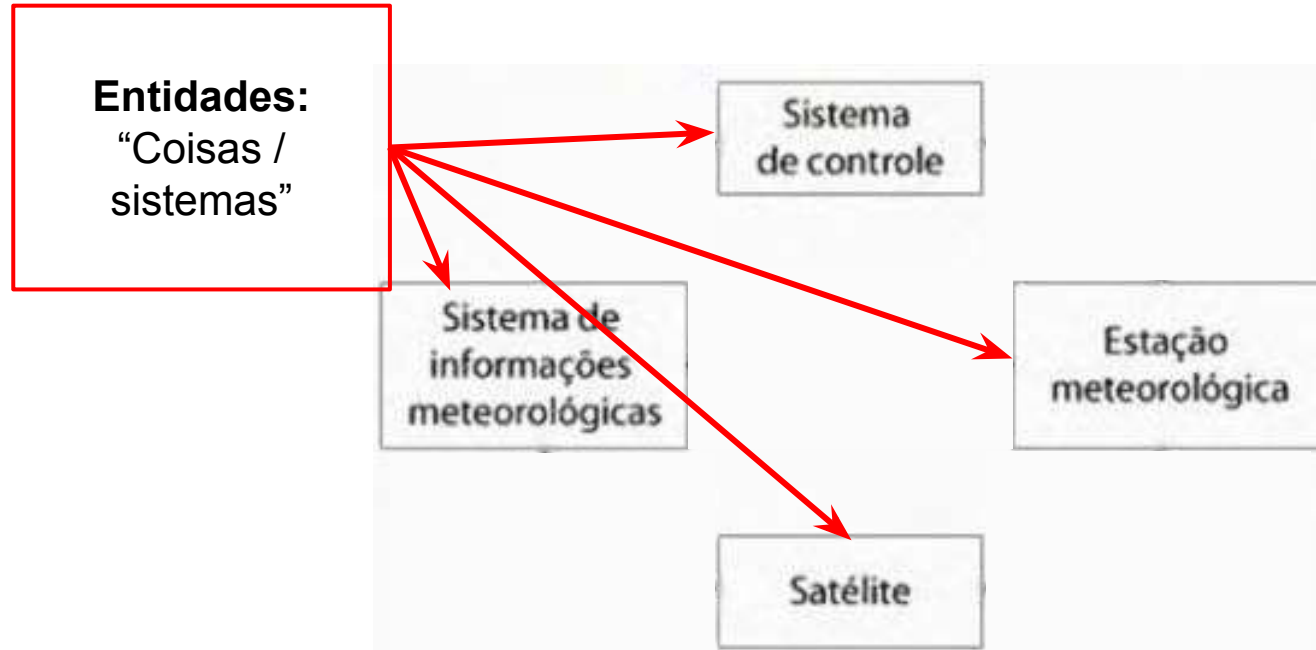


Exemplo de Diagrama de Entidade-Relacionamento  
(E-R)

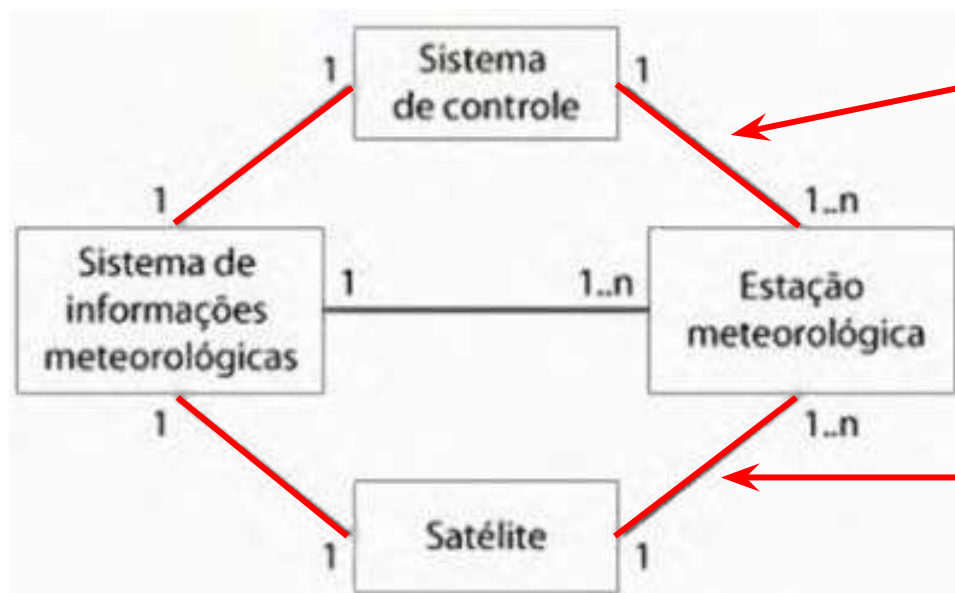


# Modelo de Entidade-Relacionamento

---



# Modelo de Entidade-Relacionamento



**Associações**

# Modelo de Entidade-Relacionamento



**Cardinalidade**

**Leia-se:**

**1** Sistema de controle

INTERAGE COM

**1 à n** estações meteorológicas

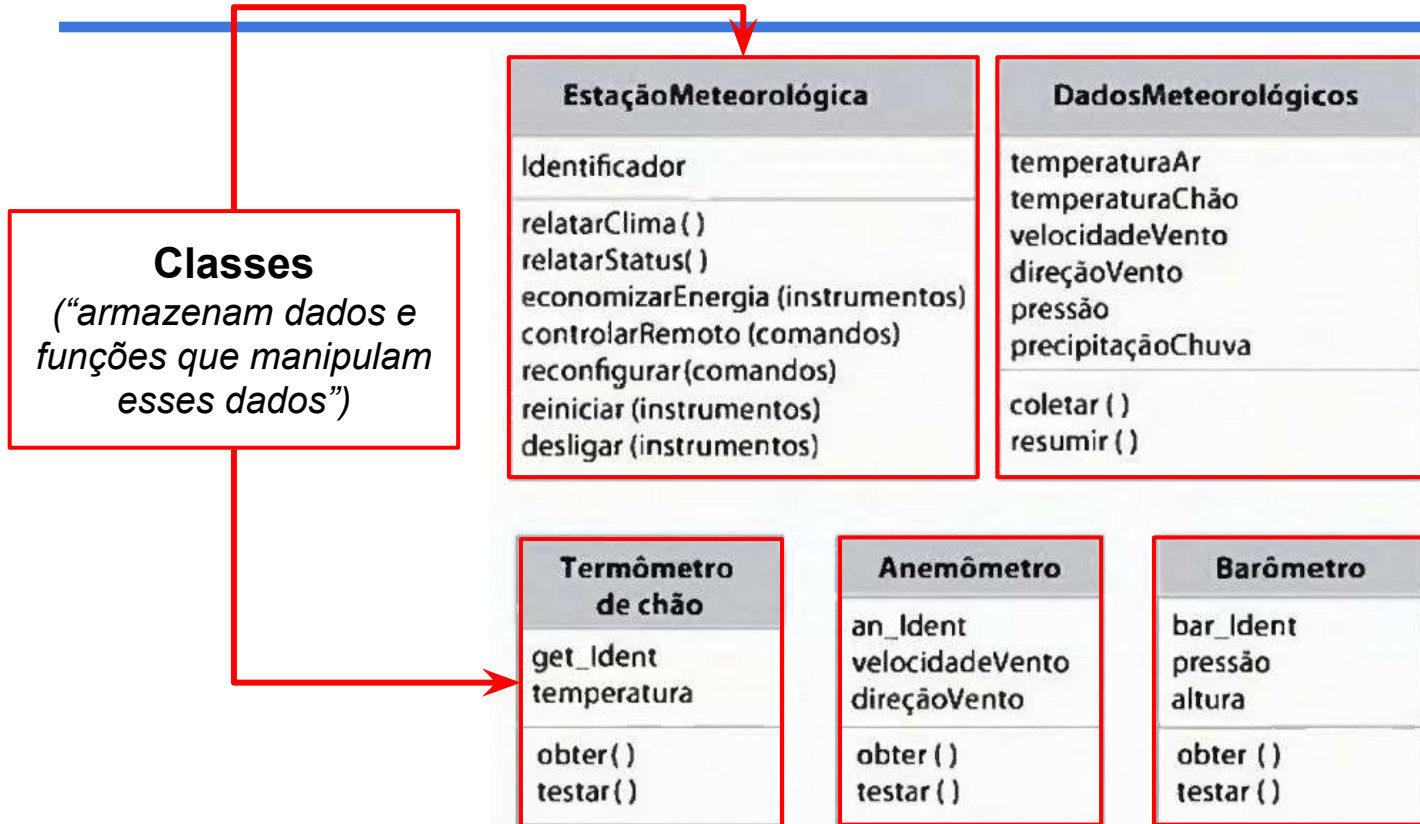
# Diagrama de Classes UML

---

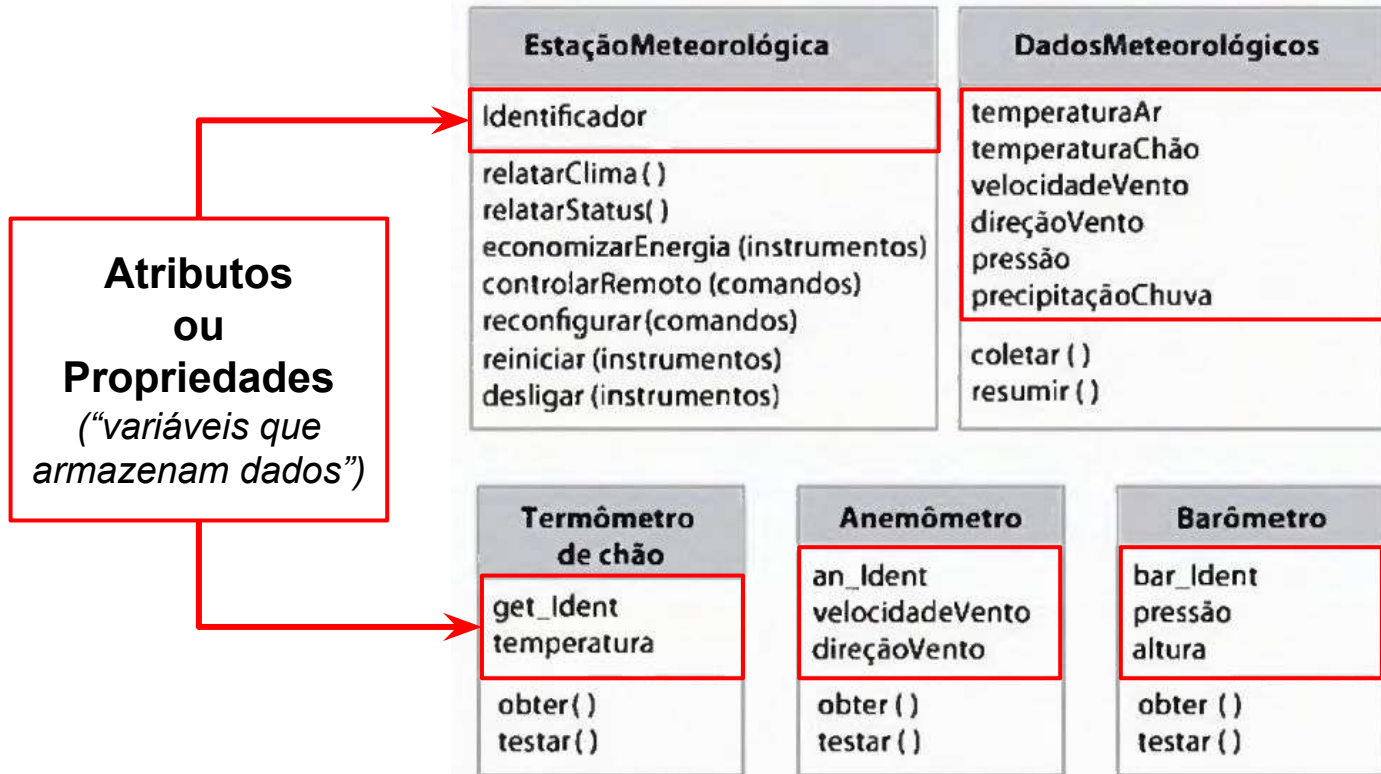
- Descreve a estrutura do sistema a partir de classes
  - Cada classe instancia um objeto, que:
    - Interage com o sistema
    - Possui estado



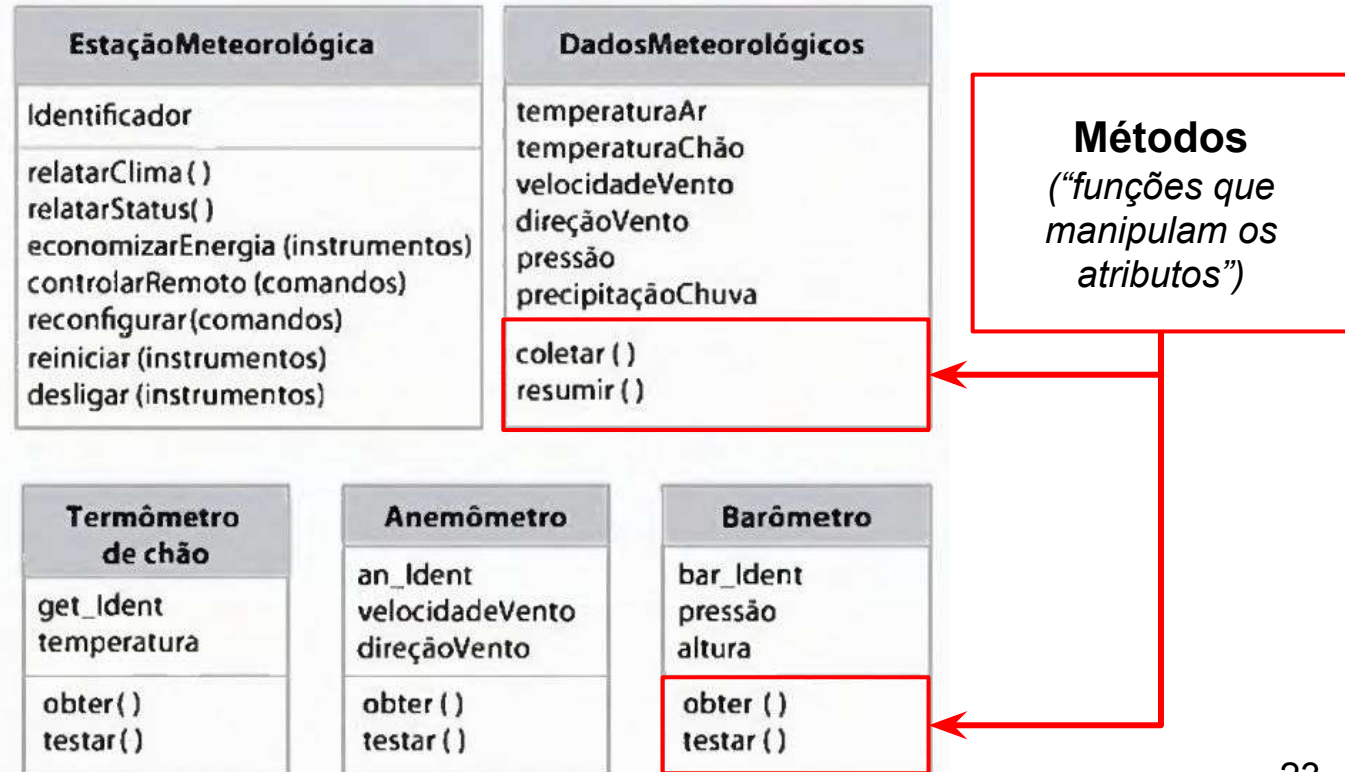
# Exemplo de Diagrama de Classes UML



# Exemplo de Diagrama de Classes UML



# Exemplo de Diagrama de Classes UML



# Exercício

Considerando as atividades do processo de projeto orientado a objetos, marque a alternativa que contém **somente** as assertivas VERDADEIRAS.

I - A definição do contexto e interações do sistema é responsável por estabelecer os limites do sistema em relação aos recursos implementados.

II - A definição do contexto e interações do sistema é responsável por estruturar como as comunicações com ambiente externo serão realizadas pelo sistema.

III - A definição do contexto e interações do sistema utiliza modelos de interação contextual para identificar os relacionamentos entre o sistema e o ambiente.

IV - Modelos estáticos descrevem a estrutura do sistema.

- ☐ Somente I e II.
- ☐ Somente I, II e III.
- ☐ Somente II e IV.
- ☐ Somente IV.
- ☐ Nenhuma das alternativas anteriores.



# Exercício

Considerando as atividades do processo de projeto orientado a objetos, marque a alternativa que contém **somente** as assertivas VERDADEIRAS.

I - A definição do contexto e interações do sistema é responsável por estabelecer os limites do sistema em relação aos recursos implementados. **V**

II - A definição do contexto e interações do sistema é responsável por estruturar como as comunicações com ambiente externo serão realizadas pelo sistema.

III - A definição do contexto e interações do sistema utiliza modelos de interação contextual para identificar os relacionamentos entre o sistema e o ambiente.

IV - Modelos estáticos descrevem a estrutura do sistema.

- ☐ Somente I e II.
- ☐ Somente I, II e III.
- ☐ Somente II e IV.
- ☐ Somente IV.
- ☐ Nenhuma das alternativas anteriores.

# Exercício

Considerando as atividades do processo de projeto orientado a objetos, marque a alternativa que contém **somente** as assertivas VERDADEIRAS.

I - A definição do contexto e interações do sistema é responsável por estabelecer os limites do sistema em relação aos recursos implementados. **V**

II - A definição do contexto e interações do sistema é responsável por estruturar como as comunicações com ambiente externo serão realizadas pelo sistema. **V**

III - A definição do contexto e interações do sistema utiliza modelos de interação contextual para identificar os relacionamentos entre o sistema e o ambiente.

IV - Modelos estáticos descrevem a estrutura do sistema.

- ☐ Somente I e II.
- ☐ Somente I, II e III.
- ☐ Somente II e IV.
- ☐ Somente IV.
- ☐ Nenhuma das alternativas anteriores.

# Exercício

Considerando as atividades do processo de projeto orientado a objetos, marque a alternativa que contém **somente** as assertivas VERDADEIRAS.

I - A definição do contexto e interações do sistema é responsável por estabelecer os limites do sistema em relação aos recursos implementados. **V**

II - A definição do contexto e interações do sistema é responsável por estruturar como as comunicações com ambiente externo serão realizadas pelo sistema. **V**

III - A definição do contexto e interações do sistema utiliza **modelos de interação contextual** para identificar os relacionamentos entre o sistema e o ambiente. **F**

IV - Modelos estáticos descrevem a estrutura do sistema.

- ☐ Somente I e II.
- ☐ Somente I, II e III.
- ☐ Somente II e IV.
- ☐ Somente IV.
- ☐ Nenhuma das alternativas anteriores.

# Exercício

Considerando as atividades do processo de projeto orientado a objetos, marque a alternativa que contém **somente** as assertivas VERDADEIRAS.

I - A definição do contexto e interações do sistema é responsável por estabelecer os limites do sistema em relação aos recursos implementados. **V**

II - A definição do contexto e interações do sistema é responsável por estruturar como as comunicações com ambiente externo serão realizadas pelo sistema. **V**

III - A definição do contexto e interações do sistema utiliza **modelos de interação contextual** para identificar os relacionamentos entre o sistema e o ambiente. **F**


IV - Modelos estáticos descrevem a estrutura do sistema. **V**

☐ Somente I e II.

☐ Somente I, II e III.

☐ Somente II e IV.

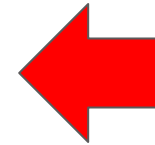
☐ Somente IV.

 ☒ Nenhuma das alternativas anteriores.

# Modelos de Interação (ou dinâmicos)

---

- Descreve as interações de um sistema com seu ambiente
  - **Diagrama de Objetos**
  - **Diagrama de Casos de Uso**
  - **Diagrama de Sequencia**
  - **Diagrama de Máquina de Estados**
  - Diagrama temporal
  - Outros diagramas UML



**Modelos mais  
usados**

# Diagrama de Objetos UML

Para representamos instancias de uma classe usamos **diagramas de objeto**  
(*modelo de interação – foco na representação do estado das instâncias*)

Termômetro	
cor	
temperatura	
obter()	
testar()	



T1:  
Termometro

cor = "azul"  
temp = 39



T2:  
Termometro

cor = "cinza"  
temp = 37

# Diagrama de Objetos UML

**Cada objeto tem suas próprias instâncias de atributos**  
*(“objetos diferentes podem ter valores diferentes”)*



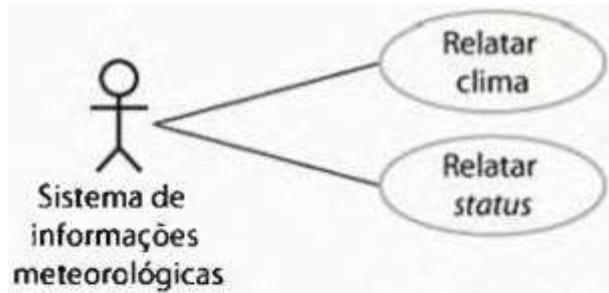
T1: Termometro
cor = “ <b>azul</b> ” temp = <b>39</b>



T2: Termometro
cor = “ <b>cinza</b> ” temp = <b>37</b>

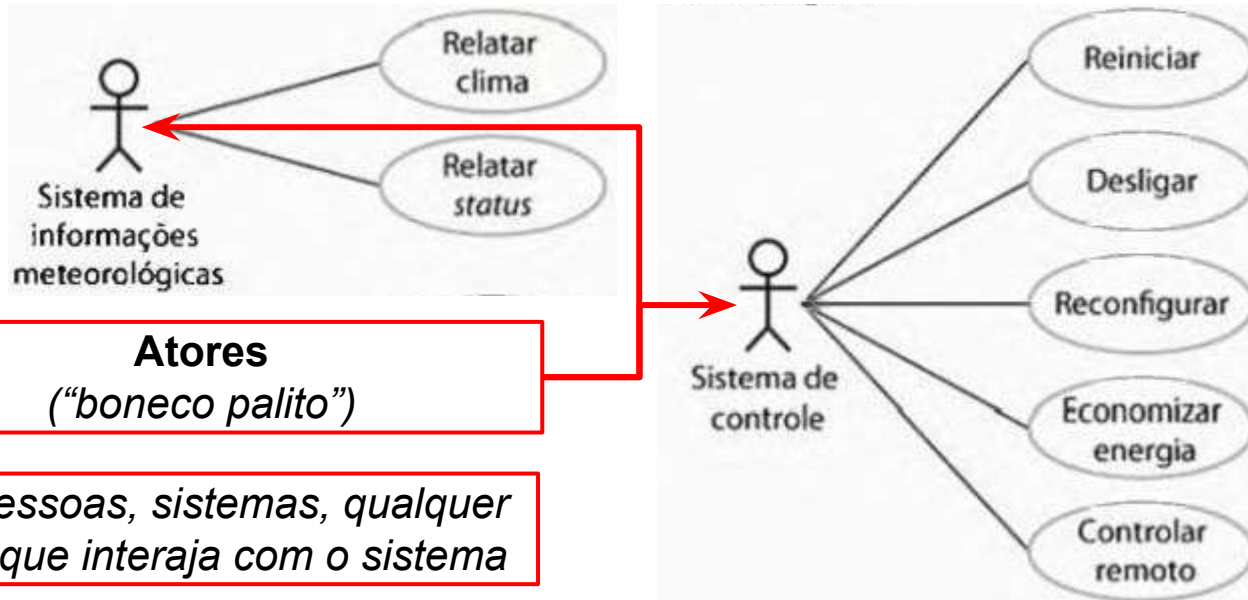
# Diagrama (ou modelo) de Casos de Uso

---

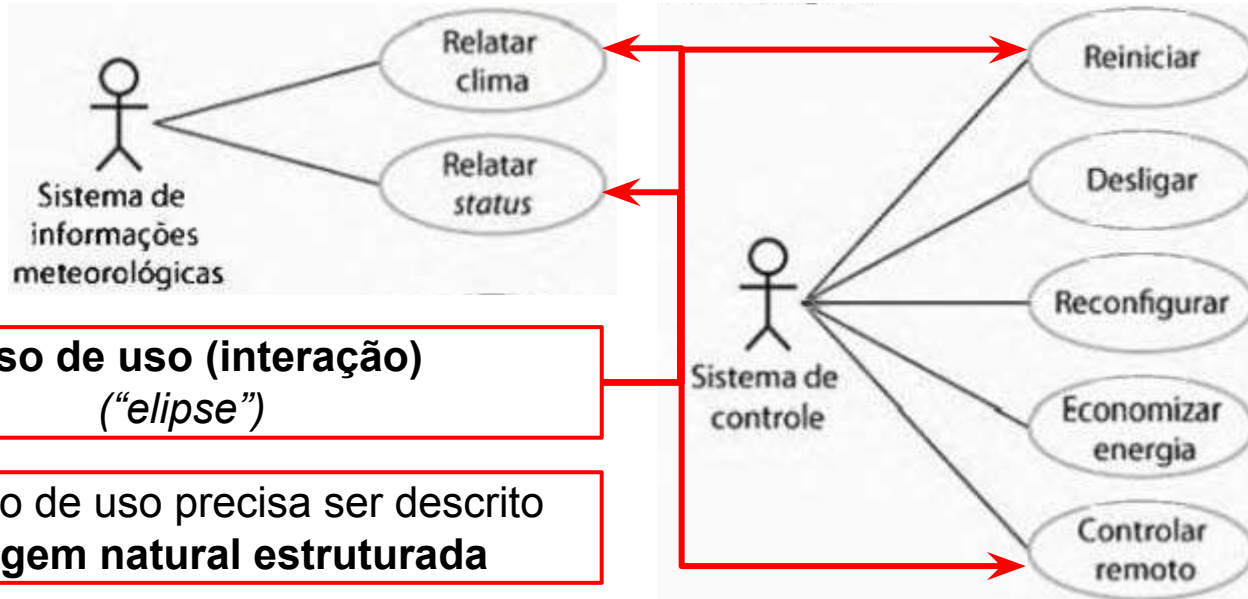




# Modelo de Casos de Uso



# Modelo de Casos de Uso



# Exemplo de descrição de caso de uso

## “Relatar clima”

---

**Caso de uso:** Relatar clima

**Atores:** Sistema de informações meteorológicas, estação meteorológica

**Dados:** A estação meteorológica envia um resumo dos dados meteorológicos coletados a partir dos instrumentos, no período de coleta, para o sistema de informações meteorológicas. Os dados enviados são o máximo, mínimo e médio das temperaturas de solo e de ar; a máxima, mínima e média da pressão do ar; a velocidade máxima, mínima e média do vento; a precipitação de chuva total e a direção do vento, amostrados a cada cinco minutos.

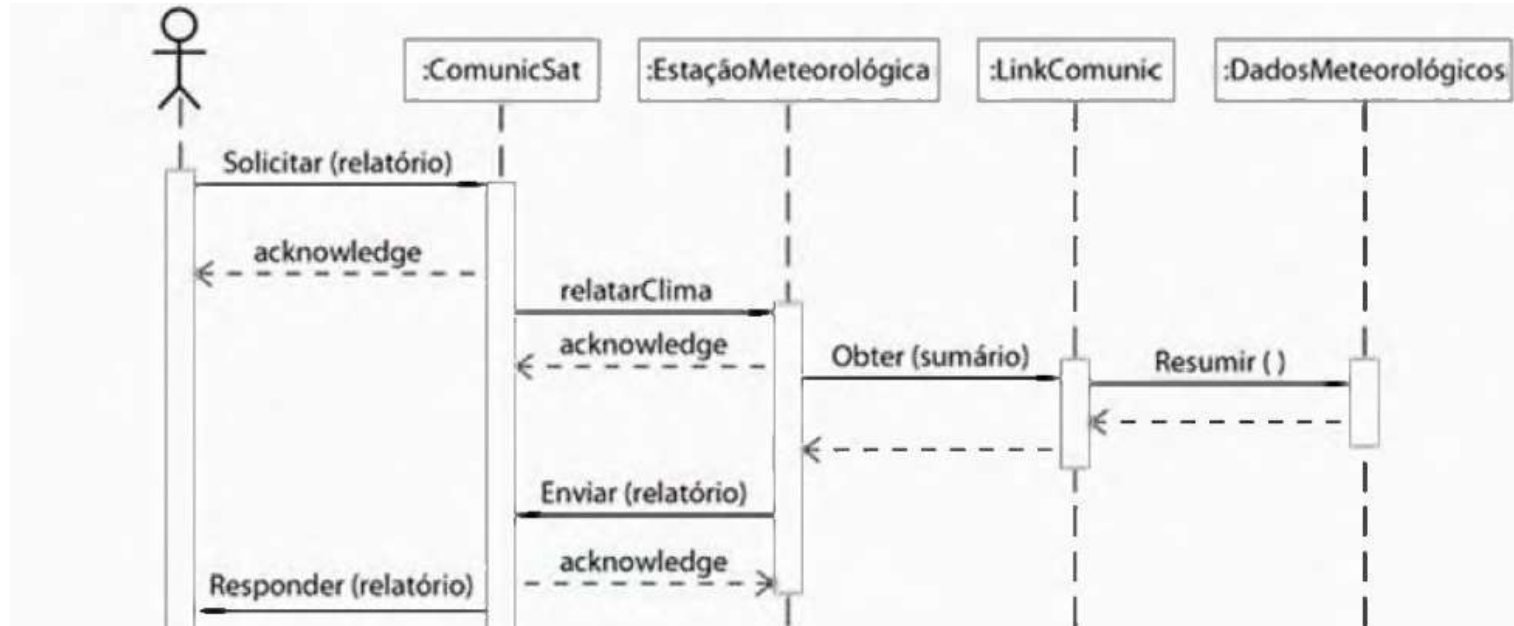
**Estímulo:** O sistema de informações meteorológicas estabelece um link de comunicação via satélite com a estação e solicita a transmissão dos dados.

**Resposta:** Os dados resumidos são enviados para o sistema de informações meteorológicas.

**Comentários:** Geralmente, solicita-se que as estações meteorológicas enviem relatórios a cada hora, mas essa frequência pode diferir de uma estação para a outra e pode ser modificada no futuro.

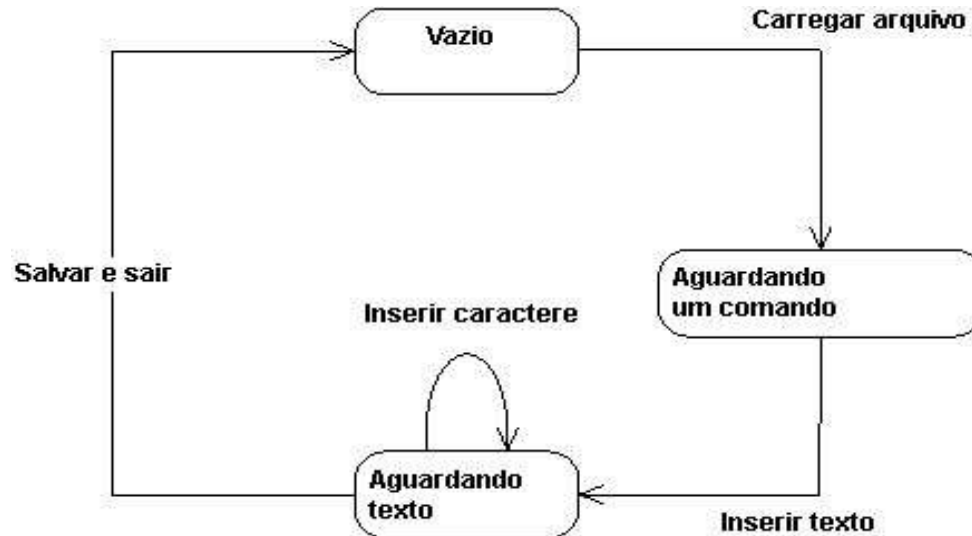
# Diagrama de Sequencia

- Descreve o comportamento de um grupo de objetos interagindo uns com os outros

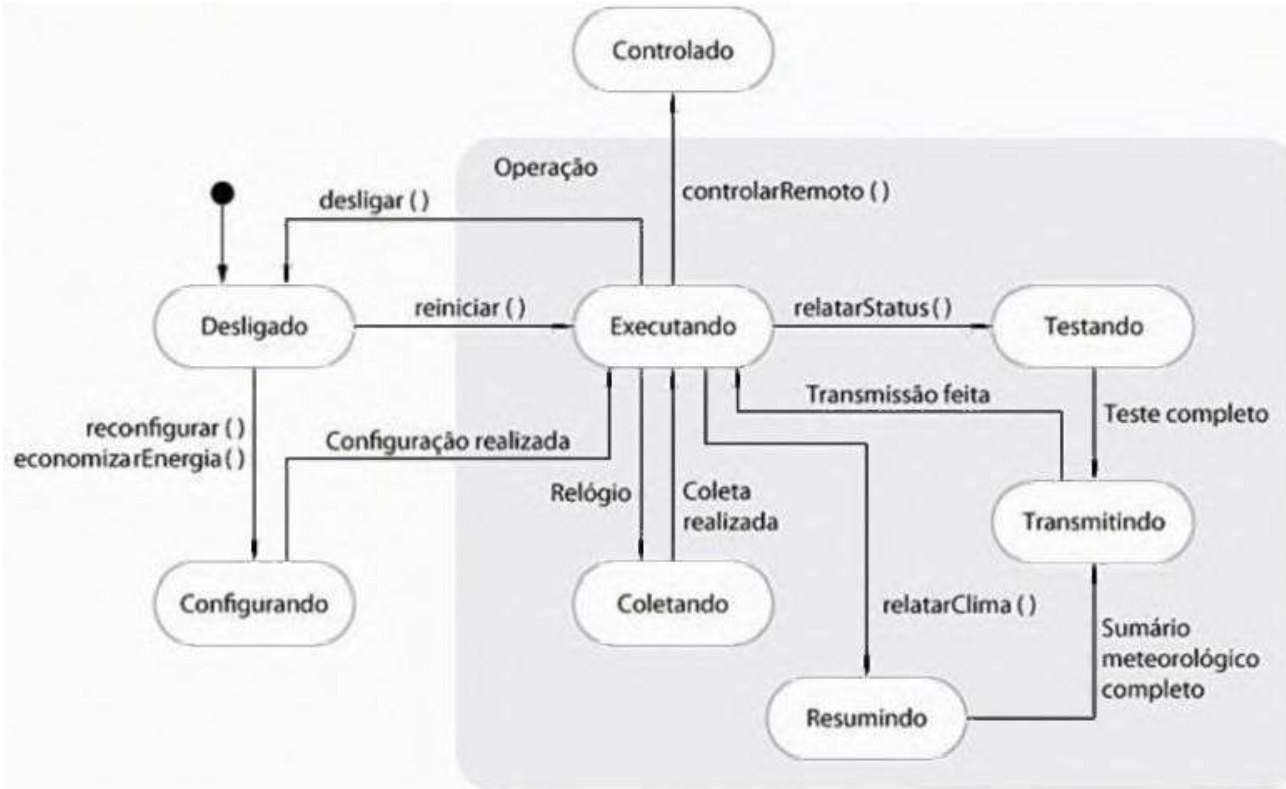


# Diagrama de Máquina de Estados

- Descreve o comportamento de um objeto ou de um subsistema em resposta à mensagens e eventos
  - Mostra a **mudança de estado**, dependendo das mensagens recebidas



# Exemplo de Diagrama de Máquina de Estados



# Projeto da arquitetura do sistema

---

- Utiliza os modelos de contexto e de interação para
  - Identificar casos, cenários de uso e ambiente de execução do sistema
  - Identificar os principais componentes do sistema e suas interações
  - Organizar os componentes usando um padrão de arquitetura conhecido
    - **Ex:** Arquitetura cliente-servidor, modelo em camadas

# Exercício

---

Considerando o processo de projeto orientado a objetos, marque a alternativa que contém **somente** as assertivas VERDADEIRAS.

I - O diagrama de objetos representa o estado de instancias de entidades.

II - Os diagramas de objetos, casos de uso e de sequencia são exemplos de modelos de interação.

III - Objetos diferentes compartilham instancias de atributos.

IV - No modelo de casos de uso, cada elipse representa um ator que interage com os casos de uso, representados por "bonecos palito".

☐ Somente I e II.

☐ Somente II.

☐ Somente II, III e IV.

☐ Somente IV.

☐ Nenhuma das alternativas anteriores.



# Exercício

Considerando o processo de projeto orientado a objetos, marque a alternativa que contém **somente** as assertivas VERDADEIRAS.

I - O diagrama de objetos representa o estado de instancias de entidades. **F**

II - Os diagramas de objetos, casos de uso e de sequencia são exemplos de modelos de interação.

III - Objetos diferentes compartilham instancias de atributos.

IV - No modelo de casos de uso, cada elipse representa um ator que interage com os casos de uso, representados por "bonecos palito".

☐ Somente I e II.

☐ Somente II.

☐ Somente II, III e IV.

☐ Somente IV.

☐ Nenhuma das alternativas anteriores.

# Exercício

Considerando o processo de projeto orientado a objetos, marque a alternativa que contém **somente** as assertivas VERDADEIRAS.

I - O diagrama de objetos representa o estado de instancias de entidades. **F**

II - Os diagramas de objetos, casos de uso e de sequencia são exemplos de modelos de interação. **V**

III - Objetos diferentes compartilham instancias de atributos.

IV - No modelo de casos de uso, cada elipse representa um ator que interage com os casos de uso, representados por "bonecos palito".

☐ Somente I e II.

☐ Somente II.

☐ Somente II, III e IV.

☐ Somente IV.

☐ Nenhuma das alternativas anteriores.

# Exercício

Considerando o processo de projeto orientado a objetos, marque a alternativa que contém **somente** as assertivas VERDADEIRAS.

I - O diagrama de objetos representa o estado de instancias de entidades. **F**

II - Os diagramas de objetos, casos de uso e de sequencia são exemplos de modelos de interação. **V**

III - Objetos diferentes compartilham instancias de atributos. **F**

IV - No modelo de casos de uso, cada elipse representa um ator que interage com os casos de uso, representados por "bonecos palito".

☐ Somente I e II.

☐ Somente II.

☐ Somente II, III e IV.

☐ Somente IV.

☐ Nenhuma das alternativas anteriores.

# Exercício

Considerando o processo de projeto orientado a objetos, marque a alternativa que contém **somente** as assertivas VERDADEIRAS.

I - O diagrama de objetos representa o estado de instancias de entidades. **F**

II - Os diagramas de objetos, casos de uso e de sequencia são exemplos de modelos de interação. **V**

III - Objetos diferentes compartilham instancias de atributos. **F**

IV - No modelo de casos de uso, cada elipse representa um ator que interage com os casos de uso, representados por "bonecos palito". **F**

☐ Somente I e II.

☒ Somente II.

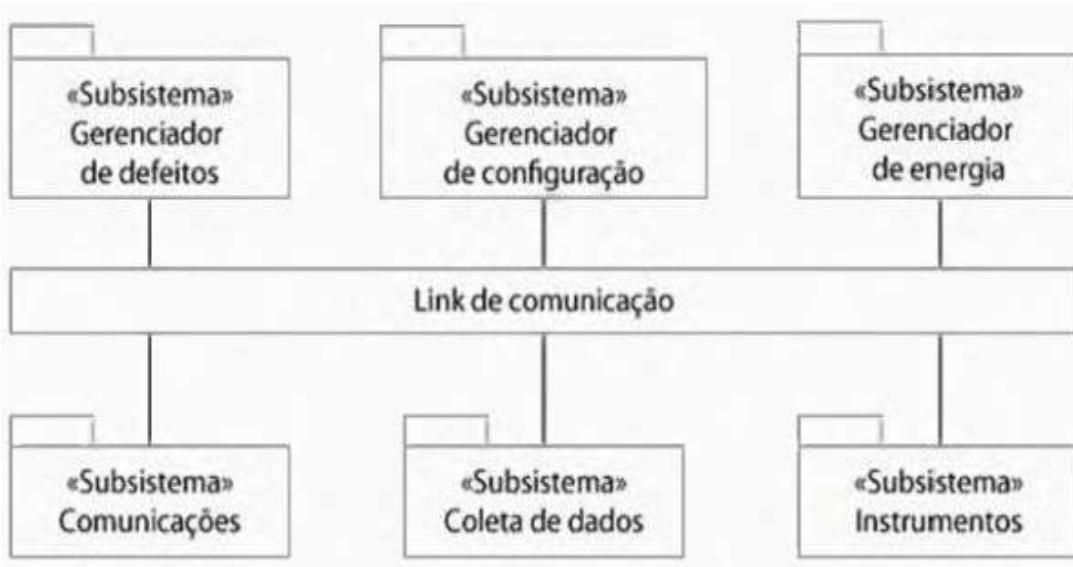
☐ Somente II, III e IV.

☐ Somente IV.

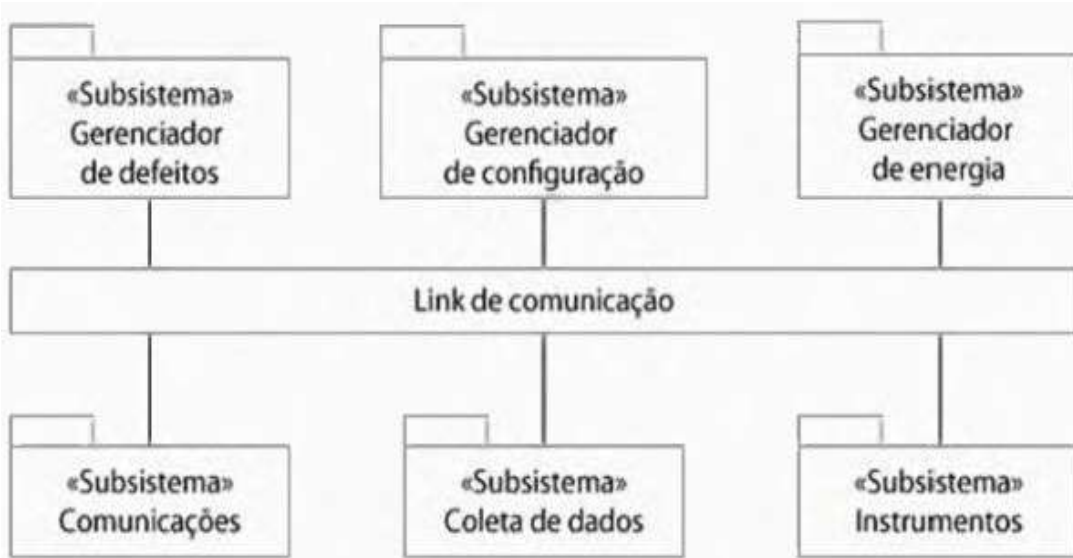
☐ Nenhuma das alternativas anteriores.

# Exemplo de arquitetura para o sistema de estação meteorológica

---



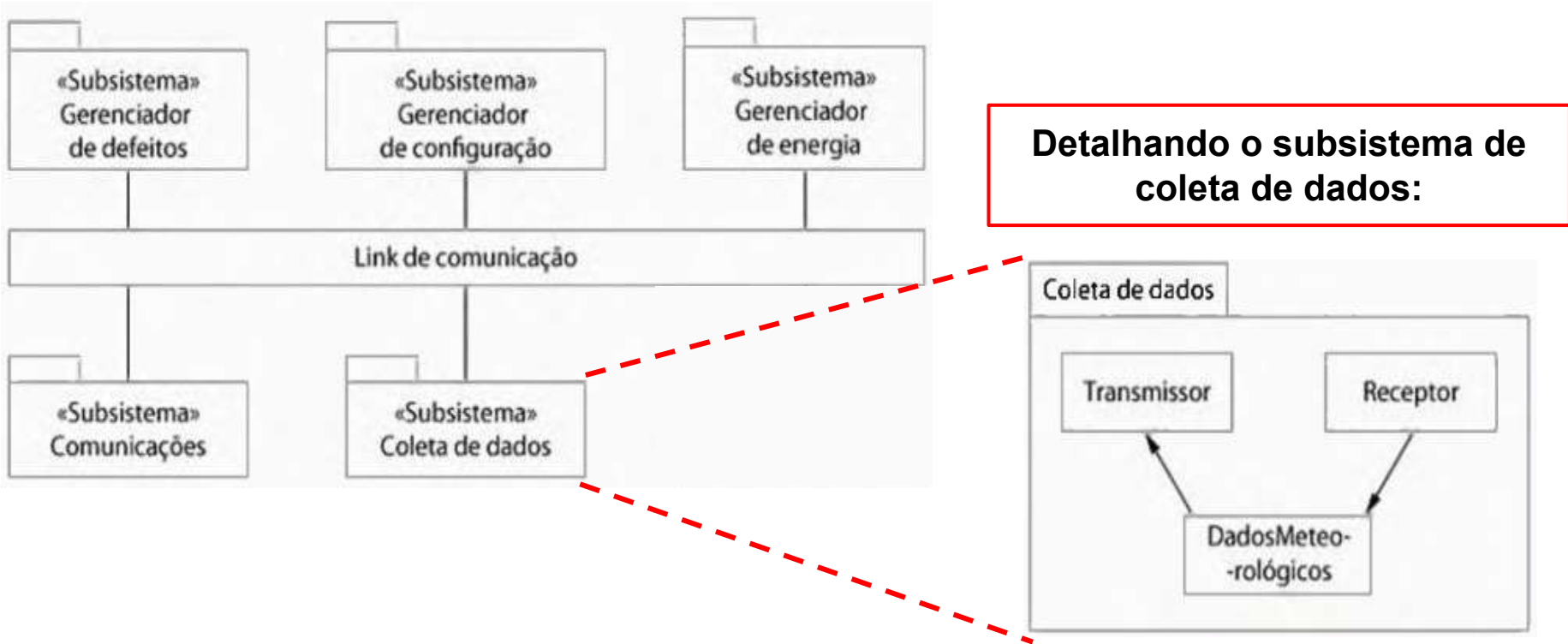
# Exemplo de arquitetura para o sistema de estação meteorológica



**Exemplo de diagrama de pacotes ou subsistemas**  
(modelo de contexto)

Mostra os subsistemas e como eles se comunicam (interfaces)

# Exemplo de arquitetura para o sistema de estação meteorológica



# Identificação dos principais objetos do sistema

---

- Consiste em identificar as classes e objetos do sistema através de:
  - Observação do ambiente real do usuário
  - Análise das interações do sistema (modelos de interação)
    - Concepção de **objetos**
  - Análise da estrutura do sistema (modelos de contexto)
    - Concepção de **classes**



# Princípios de projeto (S.O.L.I.D.)

---

- Criar projetos mais fáceis de modificar, com menos efeitos colaterais
- Os principais princípios de projeto são:
  - **Princípio Aberto-Fechado (OCP)**
  - **Princípio da Substituição de Liskov (LSP)**
  - **Princípio da Inversão da Dependência (DIP)**
  - **Princípio da Segregação de Interfaces (ISP)**
  - **Princípio do Fechamento Comum (CCP)**

# Princípios de projeto (S.O.L.I.D.)

## **Princípio aberto-fechado (OCP):**

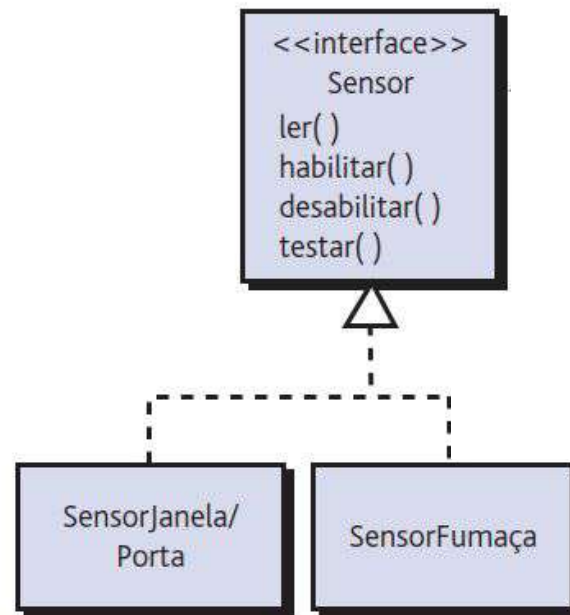
Classes (componentes) devem ser abertas para a extensão e fechada para modificações

## **Princípio da Substituição de Liskov (LSP):**

Qualquer subclasse pode ser utilizada em qualquer parte do código que espera uma superclasse

## **Princípio da Inversão de Dependência (DIP):**

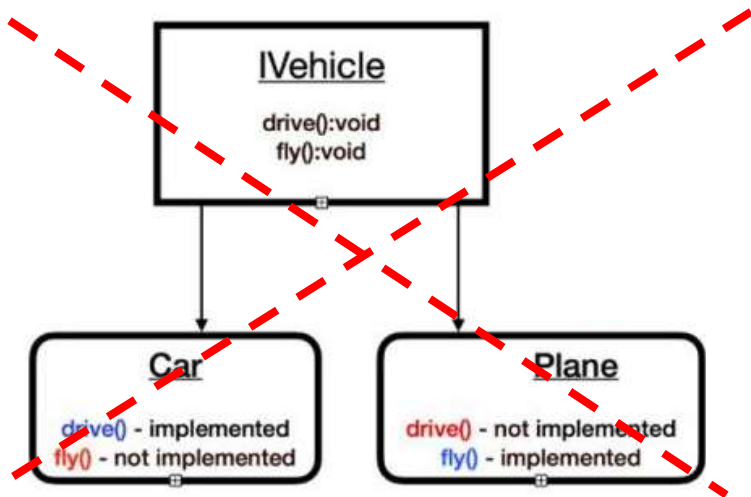
Um sistema deve depender de abstrações e não de concretizações



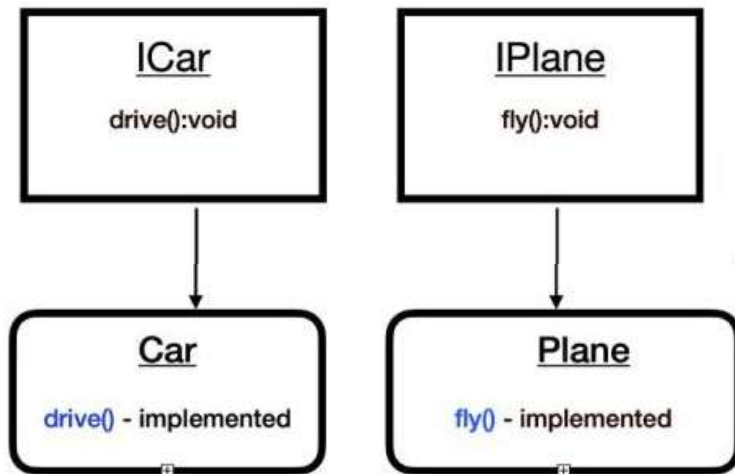
# Princípios da Segregação de Interfaces (ISP)

- É melhor usar várias interfaces específicas do cliente do que uma única interface de propósito geral

## Violação do ISP

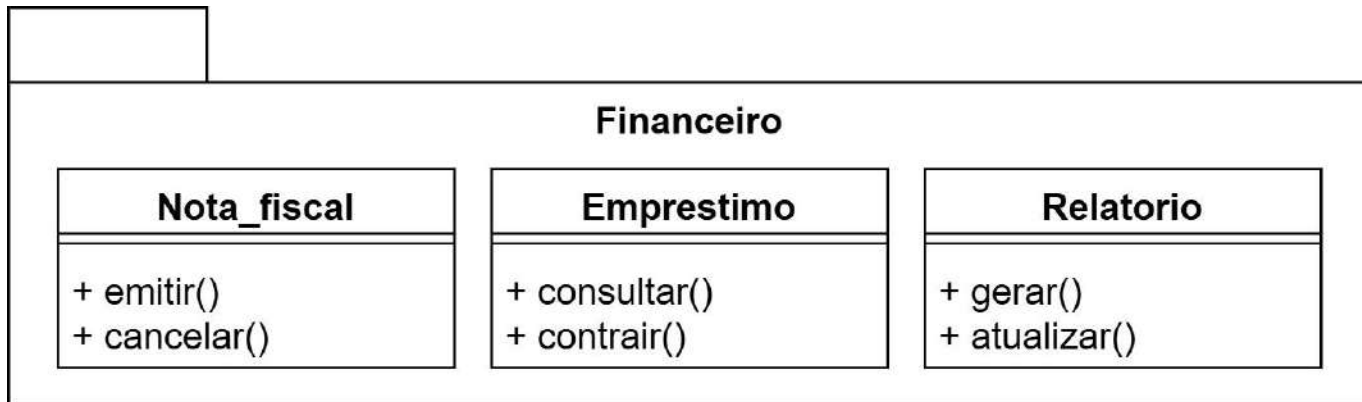


## Seguindo o ISP



# Princípio do Fechamento Comum (CCP)

- “Classes que mudam juntas, devem ficar juntas, em um mesmo pacote”
  - Cada pacote possui uma funcionalidade comum
  - Quando essa funcionalidade precisar mudar, é provável que apenas as classes contidas no pacote precisem ser modificadas



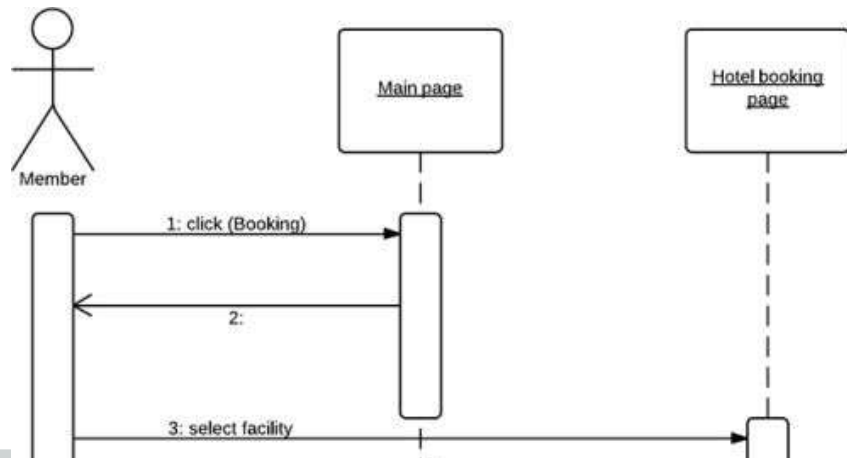
# Desenvolvimento dos modelos de projeto

---

- O desenvolvimento de **modelos** é feito através de
  - Diferentes níveis de detalhe
  - Usando diferentes modelos de projeto
    - **Deve-se minimizar o número de modelos produzidos para**
      - Reduzir custos do projeto
      - Reduzir tempo necessário para completar o sistema

# Desenvolvimento dos modelos de projeto

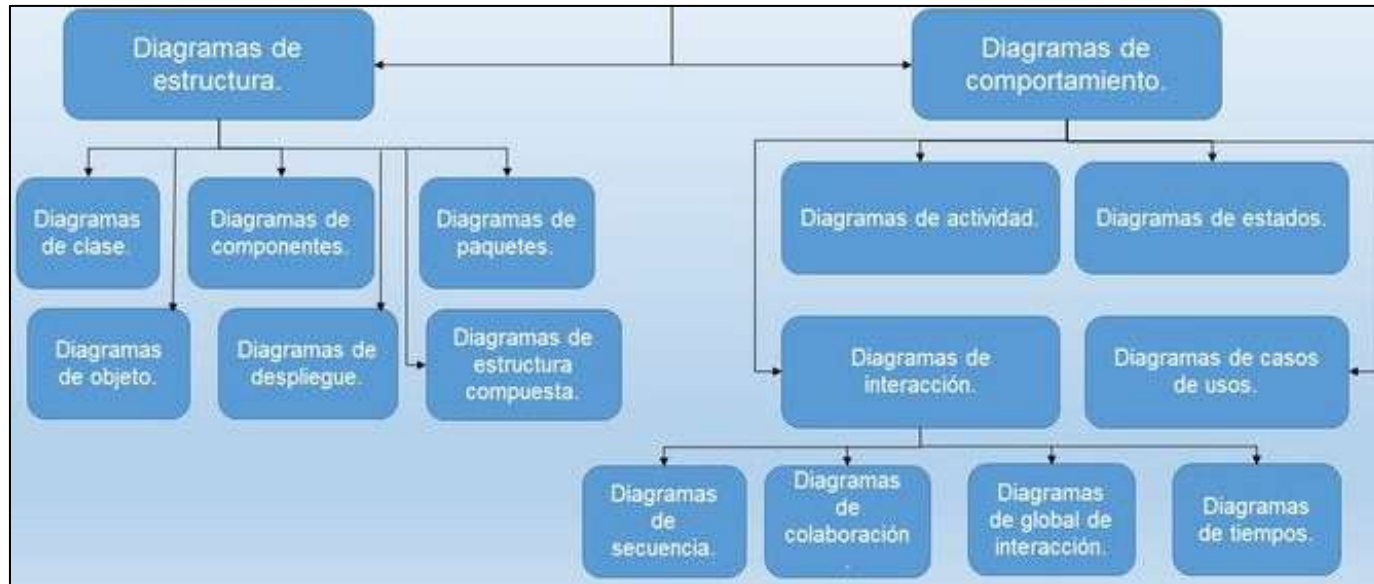
- **Modelos:** “ponte entre os requisitos e a implementação de um sistema”
  - Devem ser **abstratos** (ocultam detalhes desnecessários)
  - Incluem **detalhes suficientes** para que os programadores possam tomar decisões de implementação



**Exemplo de diagrama  
de sequencia UML**  
(modelo de interação)

# Desenvolvimento dos modelos de projeto

- A UML suporta mais de 10 de tipos diferentes de modelos
  - Raramente todos são usados



# Desenvolvimento dos modelos de projeto

---

- Modelos UML mais usados
  - **Modelos de estruturais (de contexto)**
    - Diagrama de classes
    - Diagrama de pacotes (ou de subsistemas)
  - **Modelos dinâmicos (de interação)**
    - Diagrama de sequencia
    - Diagramas de casos de uso



# Especificação das interfaces

- As interfaces devem ser especificadas de forma que os objetos e os subsistemas possam ser **projetados em paralelo**
  - Desenvolvedores trabalham em seus componentes **assumindo que**
    - A interface será implementada
    - A interface funciona da forma que foi especificada



# Estudo de caso - API REST (Web)



Uma interface **REST** permite desenvolvimento paralelo do frontend e backend

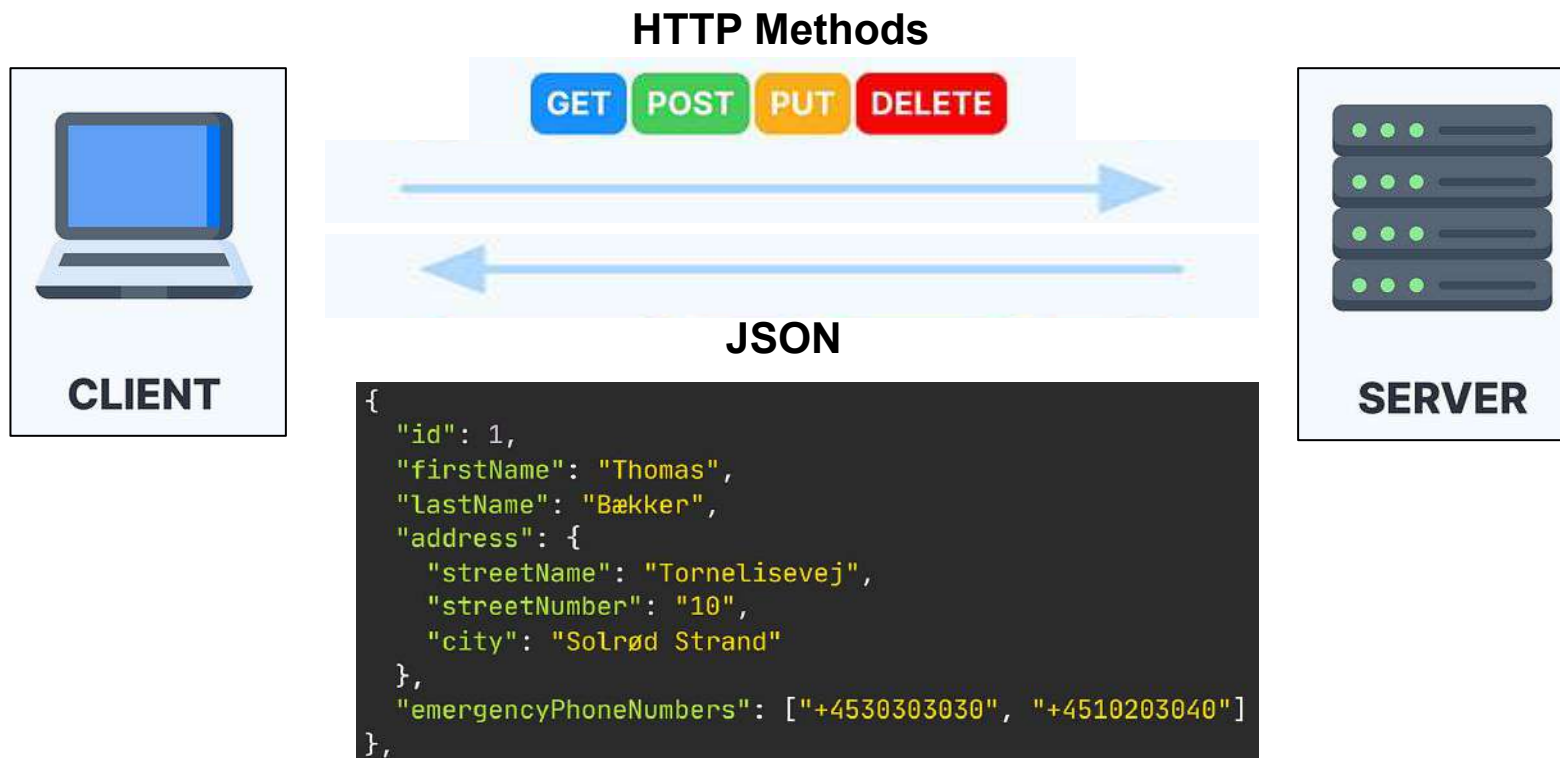
## **Desenvolvedor *Frontend*:**

Assume que o servidor sempre responde da mesma forma, usando um arquivo padronizado (ex: JSON, XML, HTML, etc)

## **Desenvolvedor *Backend*:**

Acessa DB e cria resposta em formato padronizado (ex: JSON, XML, HTML, etc)

# Estudo de caso - API REST (Web)



# Especificação das interfaces

---

- Define assinaturas e a semântica dos serviços fornecidos pelo objeto ou por um grupo de objetos
  - **Assinatura:** formato de resposta (JSON, XML, quantidade de variáveis, etc)
  - **Semantica:** o significado de cada informação, variável, função, etc
    - **Ex:** { id: 2, age: 18 }
      - Representa a idade do usuario de id “2” no sistema

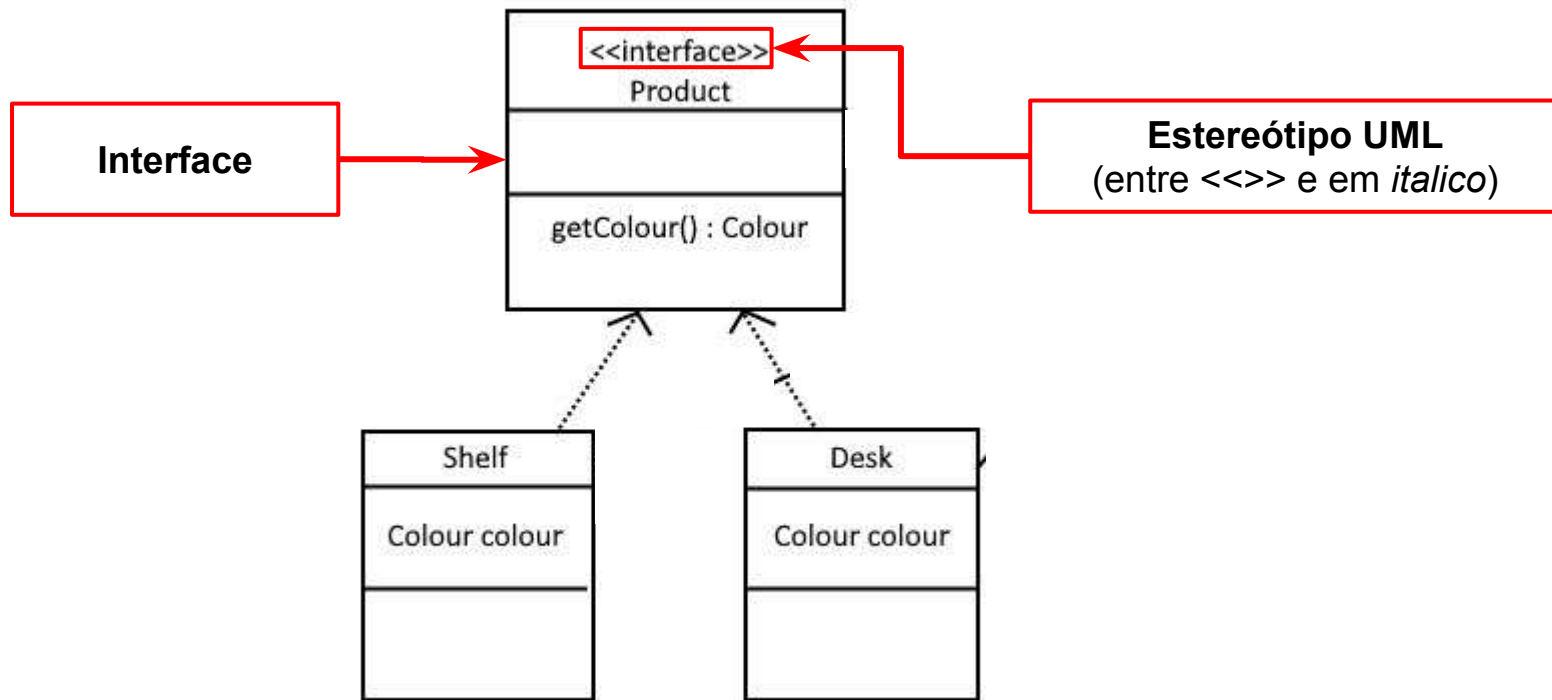
**Você não deve incluir detalhes sobre os tipos e estruturas de dados na interface**  
(no entanto, você pode incluir funções)

# Especificação das interfaces

---

- Porque não incluir a representação de dados em uma interface?
  - **Facilidade em alterar a representação de dados** sem afetar os objetos que usam esses dados
    - Interface permanece a mesma
    - **Ex:** trocar um vetor de dados por uma lista encadeada
  - **Facil manutenção do sistema**
    - Mudança na implementação não afeta a interface
    - Reduz dependência da implementação dos componentes

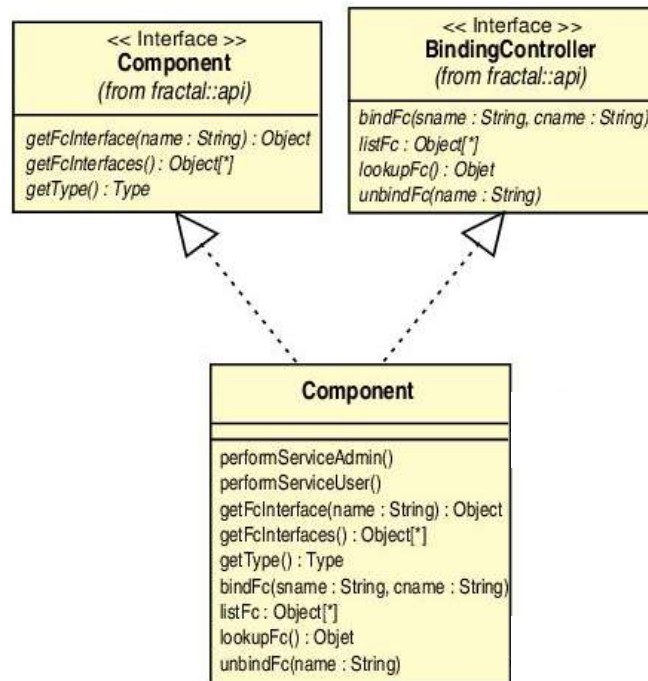
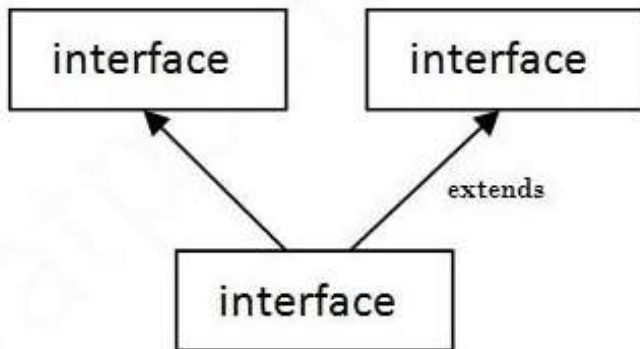
# Representação de interfaces ( << *interface* >> )



# Especificação das interfaces

- É possível ter mais de uma interface por objeto

○ **Ex:**



# Atividade em sala

---

- **Utilizem os princípios S.O.L.I.D. para melhorar o projeto do grupo**
  - **Tutoriais:**
    - <https://tinyurl.com/solid-alura>
    - <https://tinyurl.com/solid-estrategia>
- **Verifiquem os diagramas UML construídos usando PlantUML**
  - Corrijam eventuais inconsistências e erros nos diagramas
  - **Exemplos:** <https://tinyurl.com/plant-uml>



# Referencial Bibliográfico

---

- SOMMERVILLE, Ian. **Engenharia de Software**. 6. ed. São Paulo: Addison-Wesley, 2003.
- PRESSMAN, Roger S. **Engenharia de Software**. São Paulo: Makron Books, 1995.
- JUNIOR, H. E. **Engenharia de Software na Prática**. Novatec, 2010.