



Engenharia de Software I

Prof^a. Me. Cynara Leão Garcia

cynara.garcia@unicesumar.edu.br

The background features a stylized illustration of a computer monitor. The screen displays a geometric pattern of overlapping triangles in shades of blue and grey. A dark horizontal banner is superimposed over the middle of the screen, containing the title text. The monitor is set against a light blue background with some orange rectangular accents on the left and right sides.

Desenvolvimento Orientado a Objetos

Pensar Orientado a Objetos

- Onde quer que você olhe no mundo real, você vê objetos
 - Pessoas, animais, plantas, carros, etc.
- Humanos pensam em termos de objetos
 - Orientação a objetos é alto nível
 - Ou seja, mais próximo dos humanos que dos computadores



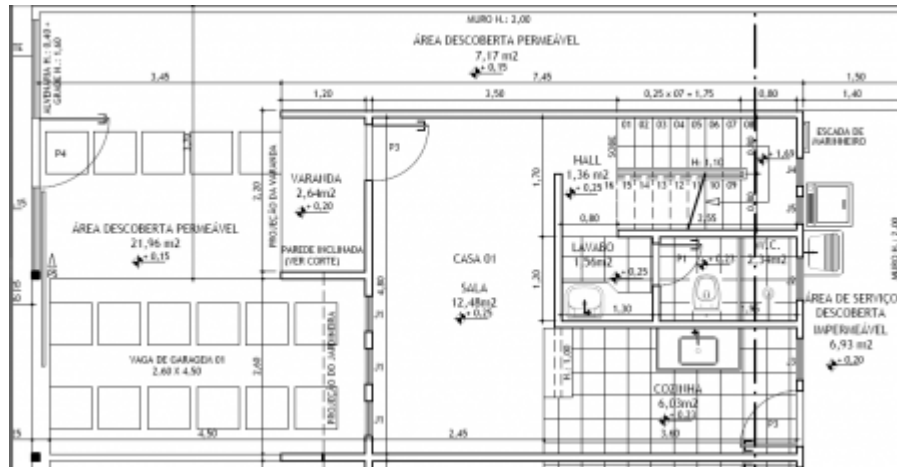
Características de Objetos

- Classificação
 - Animados: possuem vida, se movem...
 - Inanimados: não possuem vida
- Objetos possuem atributos
 - Tamanho, forma, cor, peso, etc.
- Objetos exibem comportamentos
 - Uma bola rola, um avião voa
 - ma pessoa anda, fala, pensa, etc.



Classe de Objetos

- Objeto é uma entidade que possui um estado e operações definidas sobre este estado
- Classe é um “esqueleto” para criação (instanciação) de objetos
 - Como a planta baixa é um “esqueleto” para criação de casas



Definições

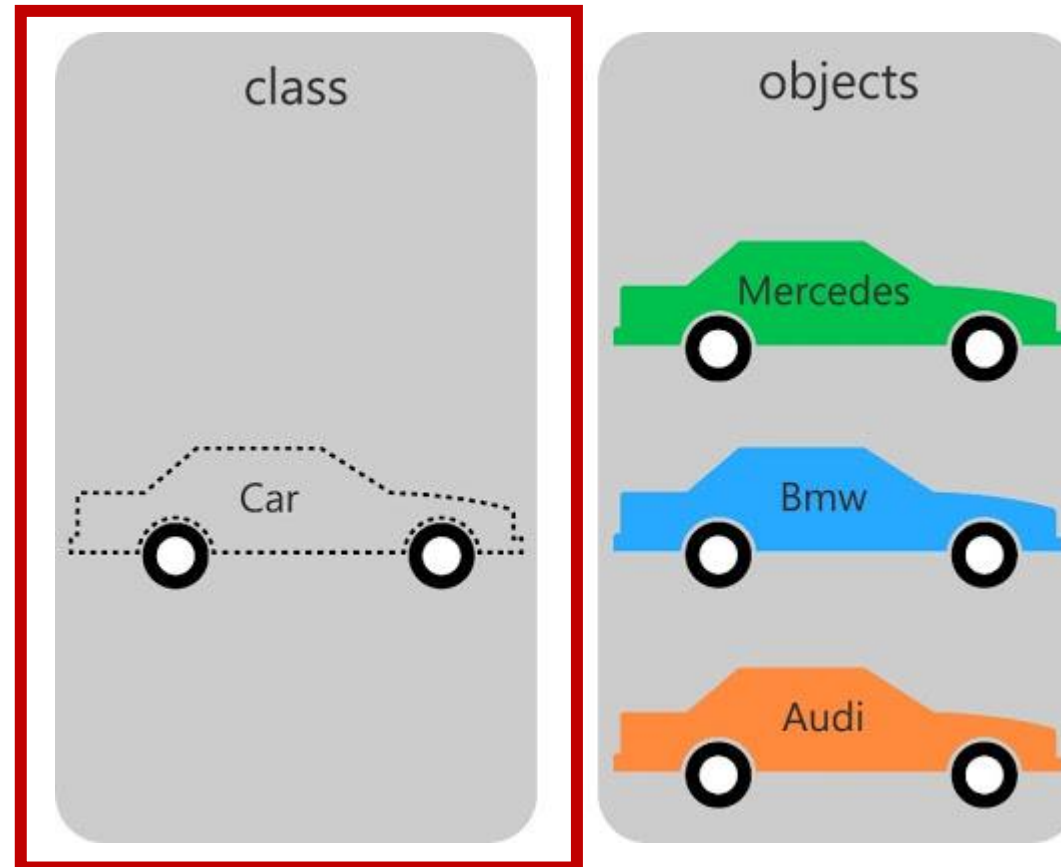
- Objeto
 - Entidade que descreve uma realidade
- Classe
 - Abstração que define o esqueleto dos objetos
- Instância
 - Objeto criado a partir de uma classe

**Vamos
exemplificar!**



Classes

- **Classes:** Uma classe é uma estrutura que **abstrai um conjunto de objetos com características similares**. Uma classe define o comportamento de seus objetos - através de métodos - e os estados possíveis destes objetos - através de atributos.



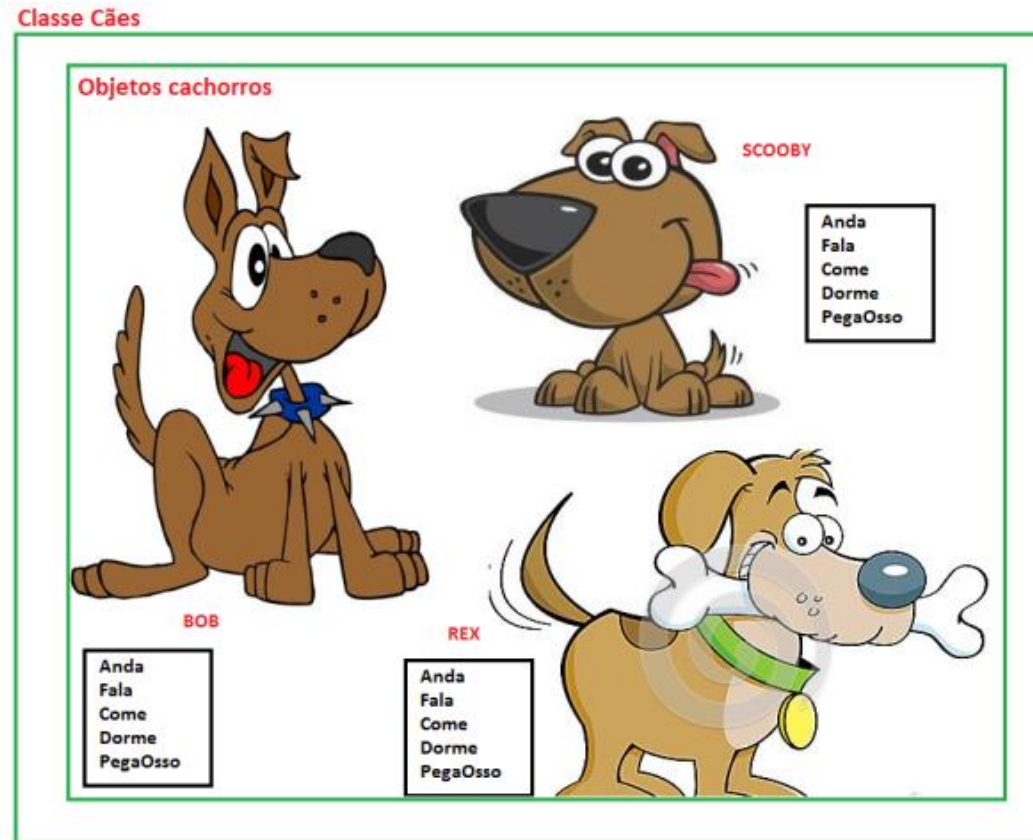
Exemplos de Classe

- ✓ pessoa
- ✓ animal
- ✓ automóvel
- ✓ publicação

- Desta forma, a primeira coisa que se deve fazer em uma análise orientado a objetos e identificar as entidades que se pretende considerar no sistema e imediatamente transformá-las em classes.

Objetos

- **Objetos:** é como chamamos a **materialização da classe**, que poderá ser usada para representar dados e executar operações. Possuem atributos e métodos.



Exemplos de Objeto

O objeto sempre será uma instância ou um elemento da uma classe. Exemplos:

- ✓ pessoa -> João
- ✓ pessoa -> Maria
- ✓ pessoa -> José

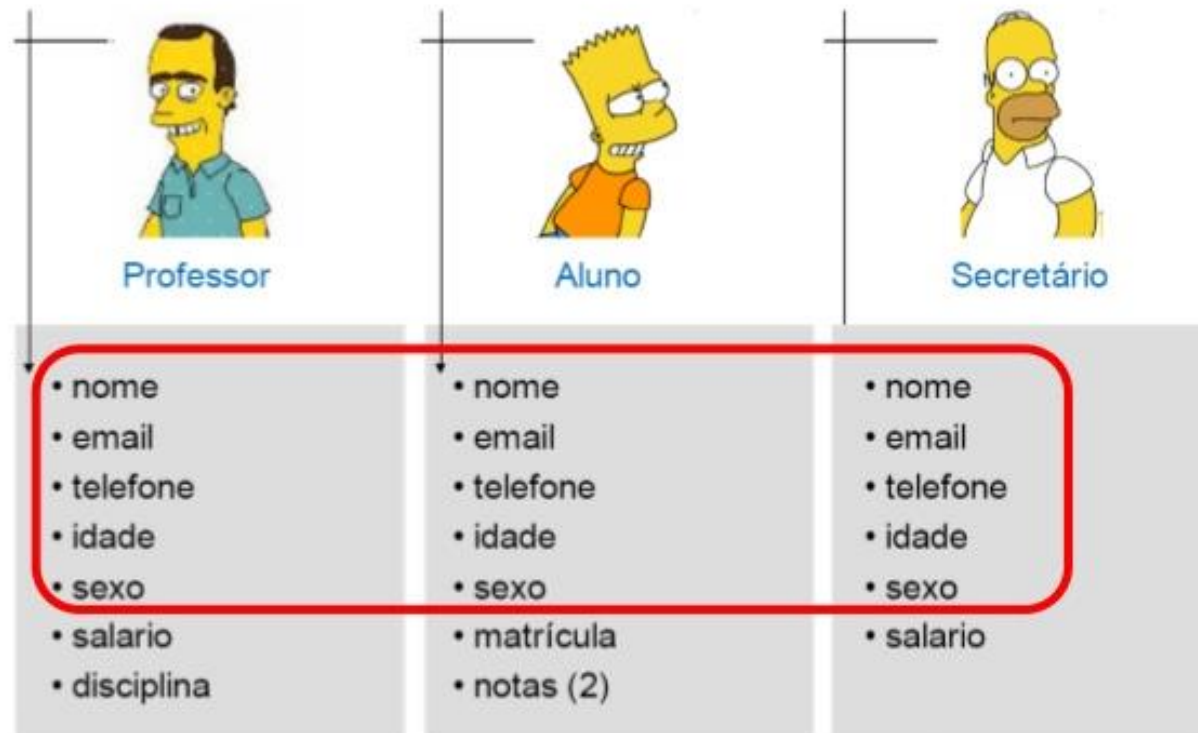
No exemplo acima pessoa é a classe enquanto que João, Maria e José são instâncias desta classe e, portanto objetos. Outros exemplos:

- ✓ animal -> Totó
- ✓ animal -> Bilú
- ✓ automóvel -> Gol
- ✓ automóvel -> Uno
- ✓ automóvel -> Fusca

Objetos

- **Atributos:** são as propriedades (ou dados) da classe.

Classe Pessoa



Métodos

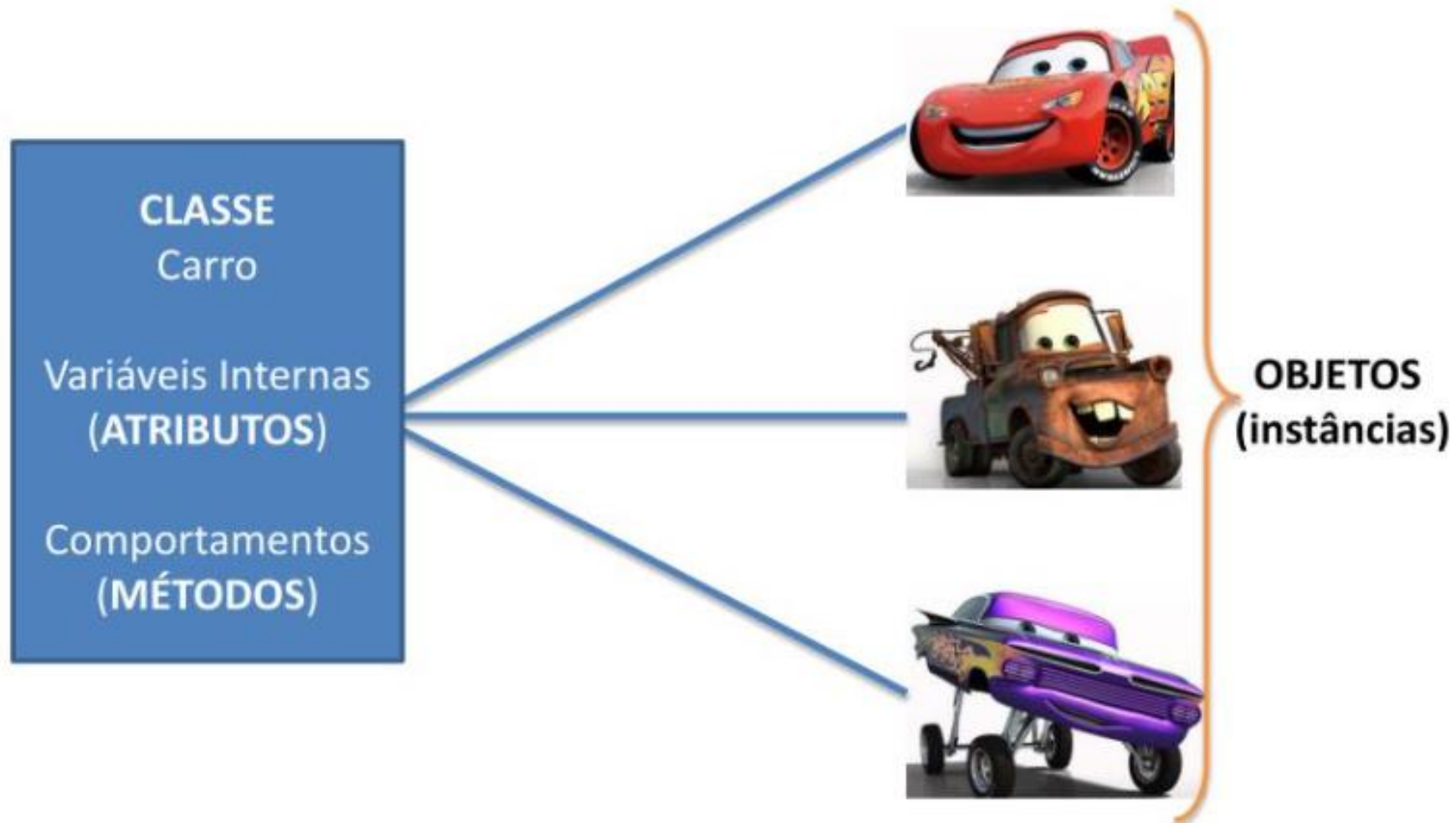
- **Métodos:** expressam as funcionalidades da classe.

Classe Pessoa

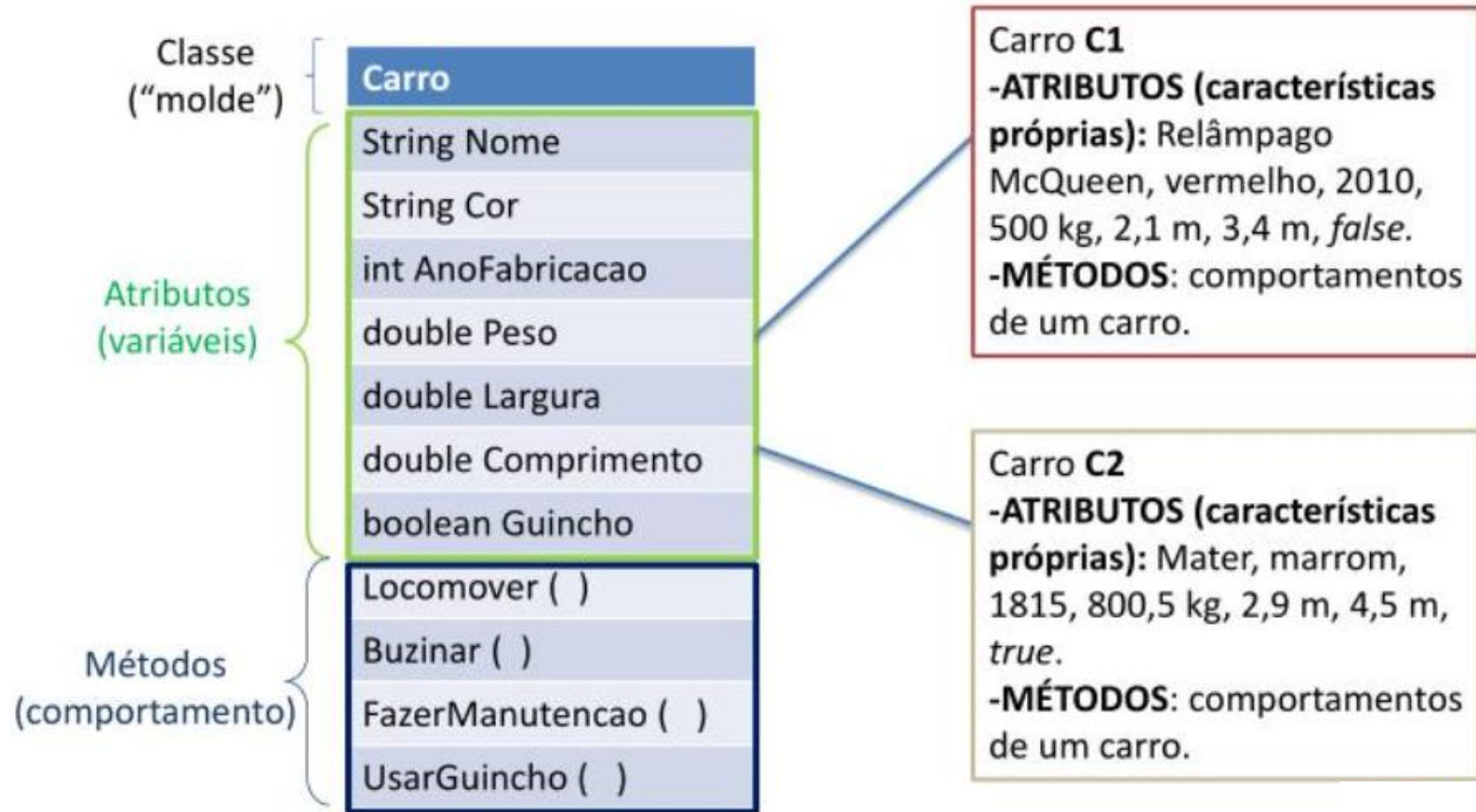


- **Atributos**
 - Nome
 - Peso
 - Altura

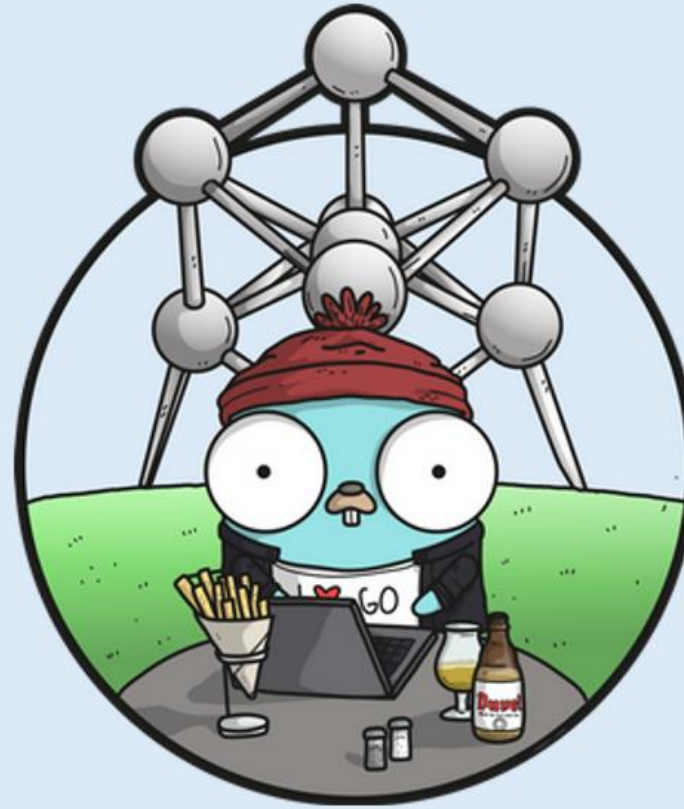
Exemplificando mais um pouco!!



Exemplificando melhor!!!



{POO}




Modelagem Orientada Objetos

Paralelo e Iterativo

As atividades não necessariamente são sequenciais

Geralmente é feito de forma iterativa

- 
- Define-se parte do contexto do sistema
 - Projeta-se parte da arquitetura
 - Identifica-se alguns objetos
 - Modela-se estes objetos
 - Define-se suas interfaces

Definir o Contexto do Sistema

Objetivo: compreensão do software que está sendo desenvolvido e de seu ambiente externo

Técnicas adotadas

- Diagramas de Casos de Uso
- Descrição dos Cenários, etc.

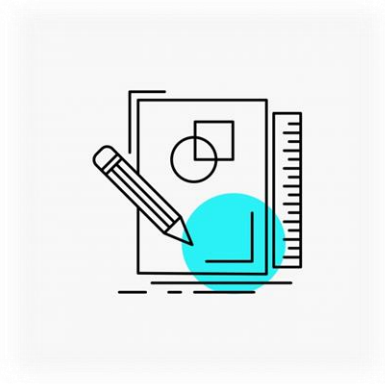
Ao definir o contexto, pode-se identificar alguns objetos do domínio

Projetar Arquitetura

Primeiro passo do projeto de sistema

O projeto arquitetural envolve

- Identificação dos componentes principais do sistema (sub-sistemas)
- Definição das interfaces de comunicação entre os componentes



Identificar os Objetos Principais

Identificação de objeto é um processo iterativo

- É improvável que você faça certo na primeira vez

Na verdade, identifica-se as classes de objetos

Não há fórmula mágica para a identificação de objetos



Uma abordagem para Identificação

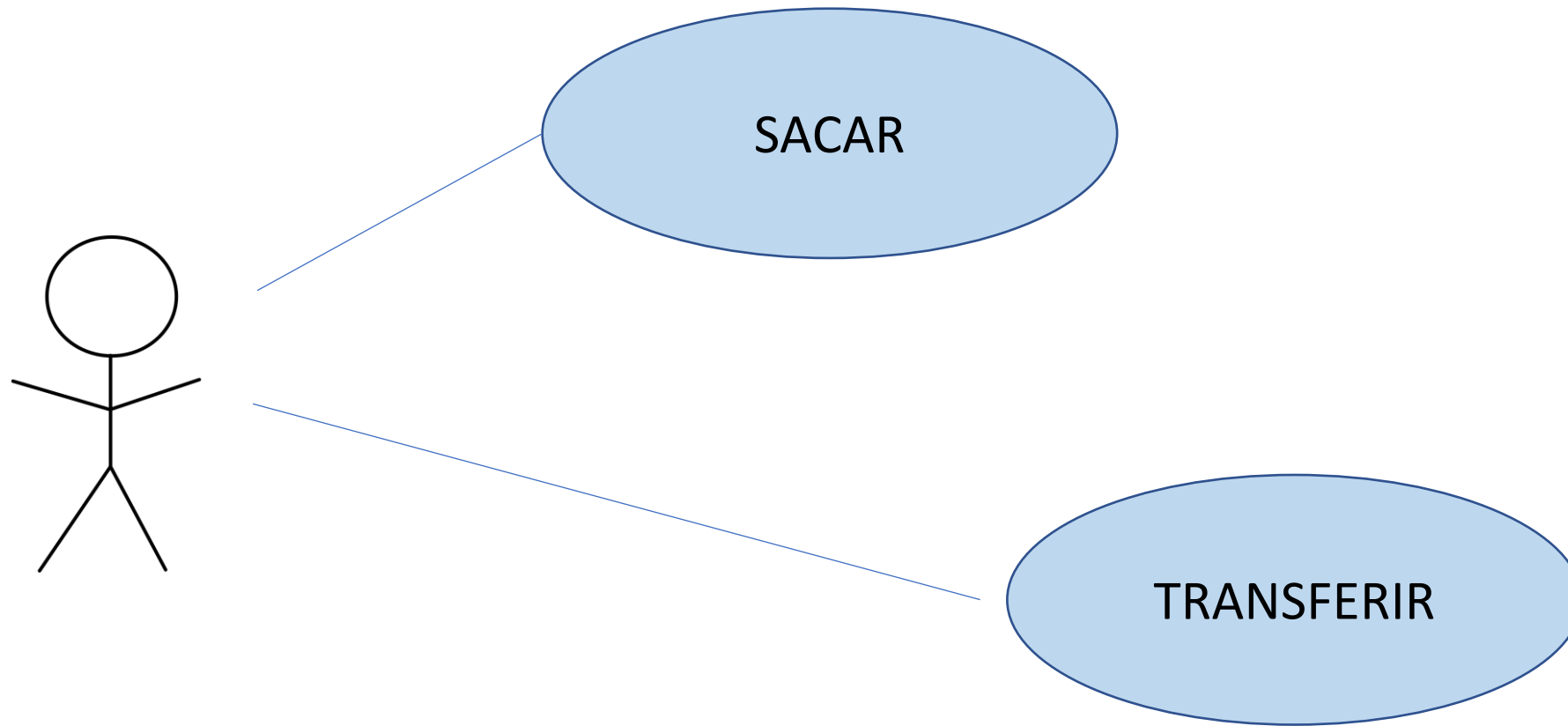
Análise gramatical baseada em

- Descrição em linguagem natural do sistema
- Descrição dos cenários de uso

Como proceder

- Substantivos: objetos ou atributos
- Verbos: métodos
- Refinar e definir novos objetos usando o conhecimento do domínio do sistema

Diagrama de Caso de Uso



Exemplo de Cenário

Nome do Cenário: Sacar

Ator: Cliente

Pré-condição: Conta e senha validadas

Fluxo normal

1. Entrar com valor do saque
2. Confirmar dados e operação
3. Debitar valor da conta do cliente

Fluxos alternativo: Saldo insuficiente

- 3.1 Apresentar aviso ao cliente

Pós-condição: Valor sacado é debitado do saldo do cliente

Exemplo de Cenário

Nome do Cenário: Sacar

Ator: Cliente

Pré-condição: Conta e senha validadas

Fluxo normal

1. Entrar com valor do saque
2. Confirmar dados e operação
3. Debitar valor da conta do cliente

Potenciais
objetos do
sistema

Fluxos alternativo: Saldo insuficiente

- 3.1 Apresentar aviso ao cliente

Pós-condição: Valor sacado é debitado do saldo do cliente

Exemplo de Cenário

Nome do Cenário: Sacar

Ator: Cliente

Pré-condição: Conta e senha validadas

Fluxo normal

1. Entrar com valor do saque
2. Confirmar dados e operação
3. Debitar valor da conta do cliente

Potenciais
atributos dos
objetos

Fluxos alternativo: Saldo insuficiente

- 3.1 Apresentar aviso ao cliente

Pós-condição: Valor sacado é debitado do saldo do cliente

Exemplo de Cenário

Nome do Cenário: Sacar

Ator: Cliente

Pré-condição: Conta e senha validadas

Fluxo normal

1. Entrar com valor do saque
2. Confirmar dados e operação
3. Debitar valor da conta do cliente

Potenciais
métodos dos
objetos

Fluxos alternativo: Saldo insuficiente

- 3.1 Apresentar aviso ao cliente

Pós-condição: Valor sacado é debitado do saldo do cliente

Modelos de Projeto

Fazem a ligação entre requisitos (problema) e implementação (solução)

Mostram os objetos ou as classes de objetos e os relacionamentos entre essas entidades

Devem incluir detalhes suficientes para facilitar a programação

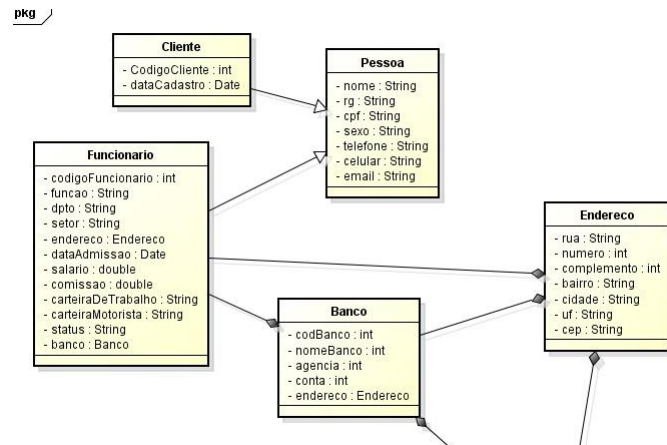
Várias Visões

Para evitar modelos complexos, eles são quebrados em diversas visões

Modelos estáticos descrevem a estrutura estática das classes

Modelos dinâmicos descrevem as interações dinâmicas entre os objetos

O modelo estático mais utilizado é o **Diagrama de Classes**





Engenharia de Software I