

Herança e Polimorfismo



Givanaldo Rocha de Souza

<http://docente.ifrn.edu.br/givanaldorochoa>
givanaldo.rocha@ifrn.edu.br

Programação Orientada a Objetos

Prof. Bruno Gomes

Projeto Orientado a Objetos

- O que desejam?
 - Produção de projetos conceitualmente simples o bastantes para serem entendidos
 - Poderosos o suficientes para resolver problemas pesados de forma eficaz
- Uma Solução:
 - Orientação a Objetos

Projeto Orientado a Objetos

- **Objetivos:**
 - Robustez
 - Sistemas confiáveis, tolerante a falhas
 - Adaptabilidade
 - Capacidade de reagir conforme o contexto
 - Reutilizabilidade
 - Reutilização do software

Projeto Orientado a Objetos

- **Princípios:**
 - Abstração
 - Encapsulamento
 - Modularidade

Abstração

- Decompor um sistema complicado em suas partes fundamentais
- Descrevê-las em uma linguagem simples e precisa
 - Atribuir-lhes um nome e descrever suas funcionalidades
- Abstrair os detalhes desnecessários
- Modelar conceitos, entidades, elementos, problemas e características do mundo real levando em conta apenas os detalhes importantes
- Vantagem:
 - Robustez

Encapsulamento

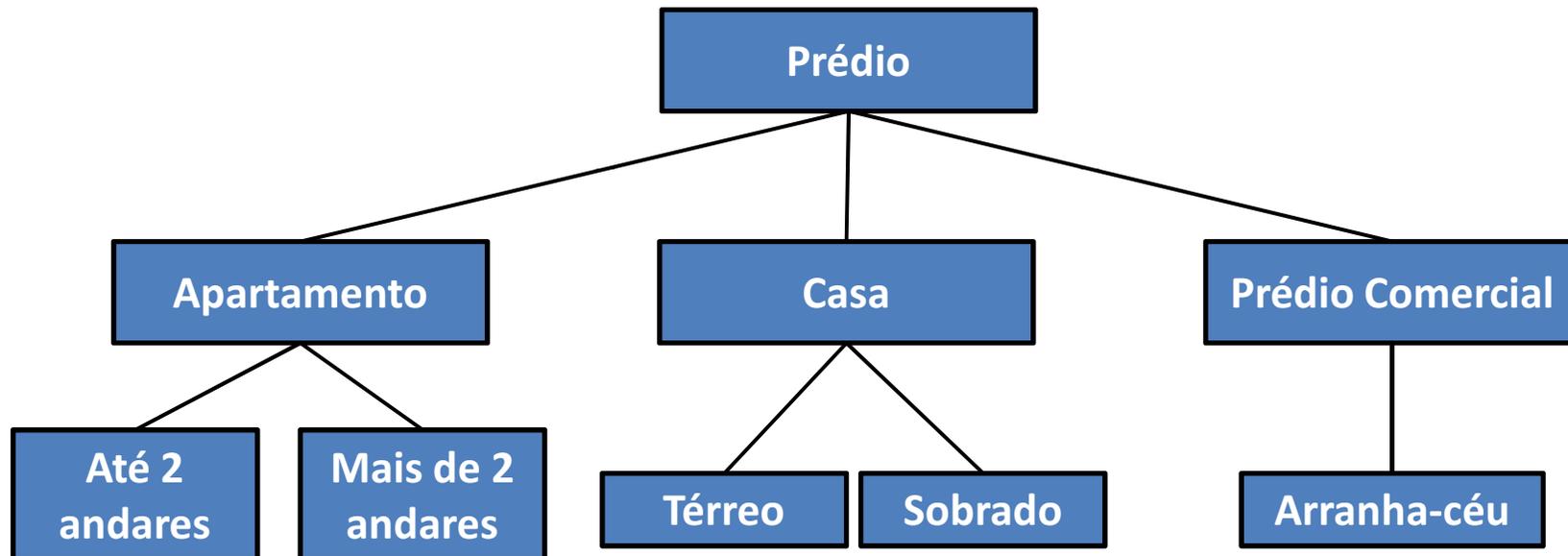
- Diferentes componentes de um sistema não devem revelar detalhes internos de suas respectivas implementações
- Vantagens:
 - Liberdade na implementação dos detalhes do sistema
 - Permite a adaptação (modificar partes de um programa sem afetar outros componentes do mesmo)

Modularidade

- Divisão dos componentes do software em unidades funcionais separadas
- Essas unidades se interagem entre si, fazendo com que o sistema funcione de forma adequada
- Vantagem:
 - Reutilização de Software

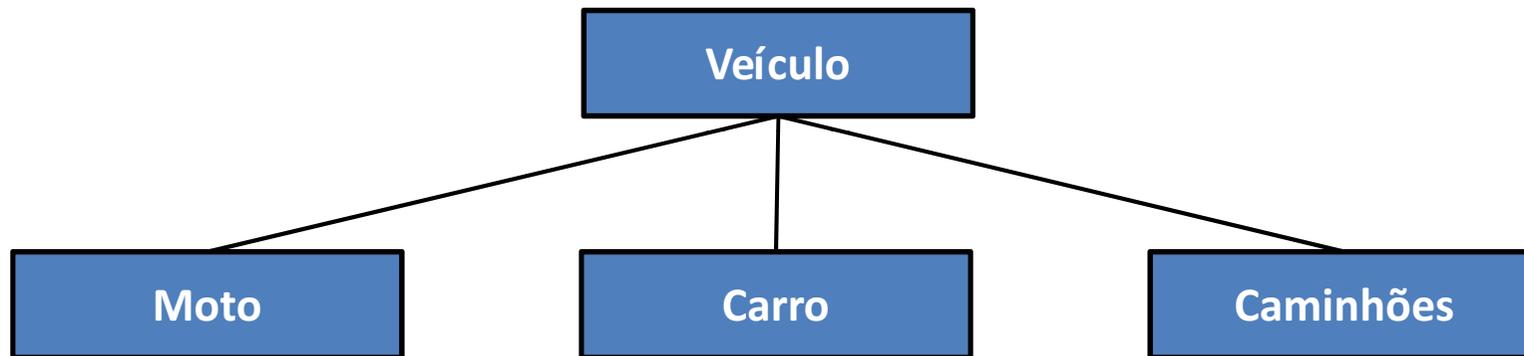
Modularidade

- Modelo Hierárquico



Modularidade

- Modelo Hierárquico

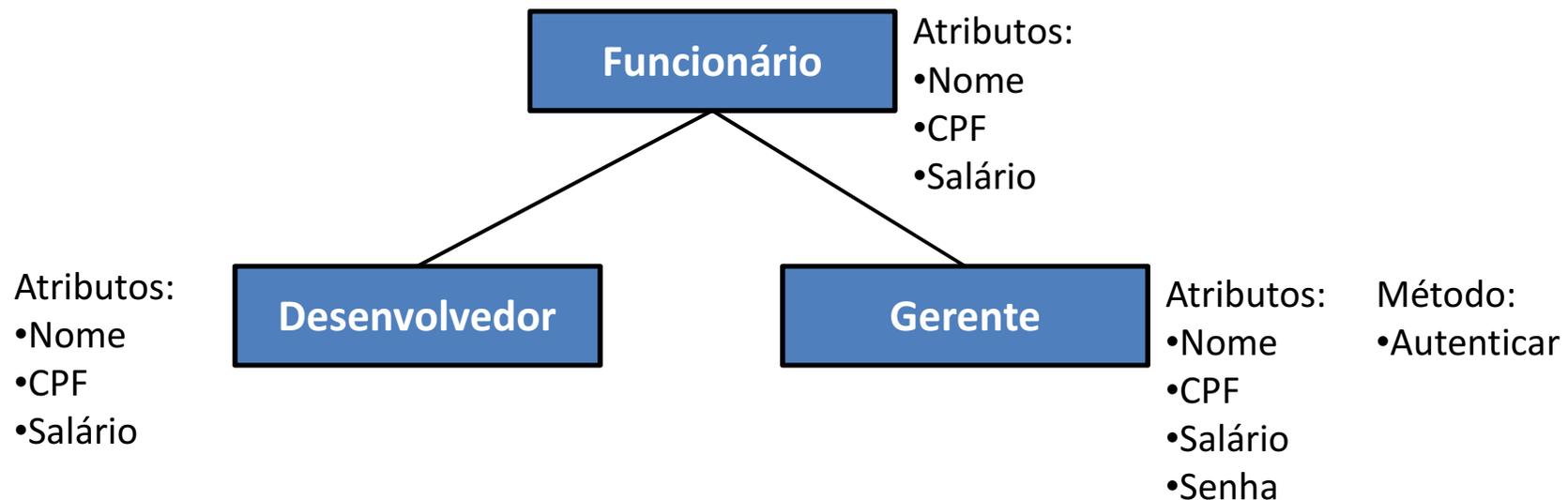


Tipos de Modelo Hierárquico

- Tipos de Modelo Hierárquico:
 - Herança
 - Polimorfismo

Herança

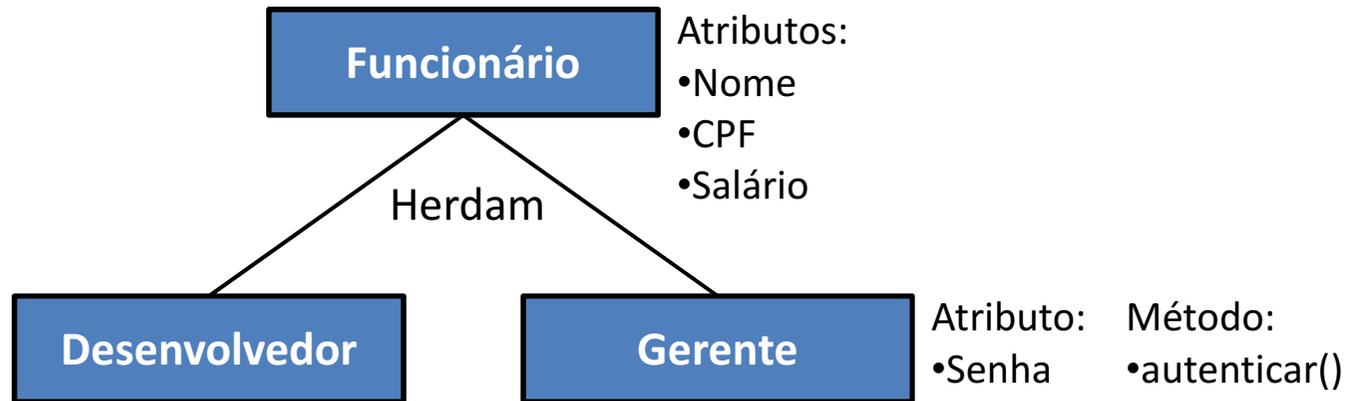
- Situação - 2 Tipos de Funcionários:



- Problema:
 - Repetição de alguns atributos para todos os funcionários
- Como Resolver?

Herança

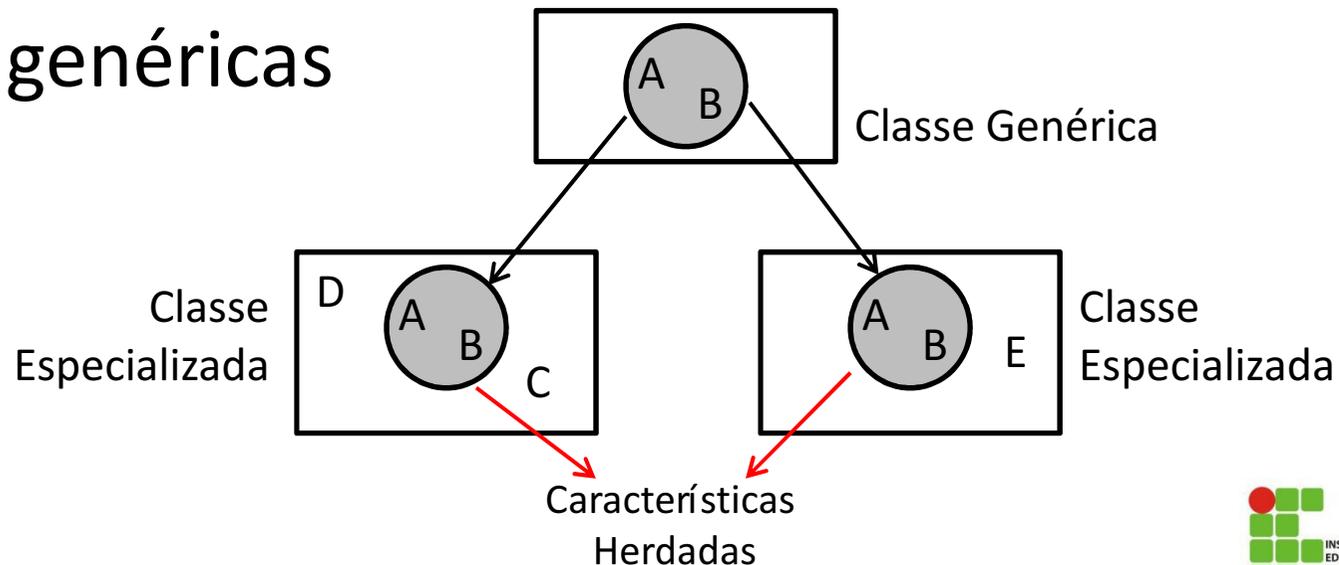
- Solução: **Utilização de Herança**



- Desenvolvedor e Gerente herdam os atributos e métodos de Funcionário

Herança

- Estrutura Hierárquica e modular
- Projeção de classes genéricas que podem ser especializadas em classes mais particulares
- Classes especializadas reutilizam o código das mais genéricas

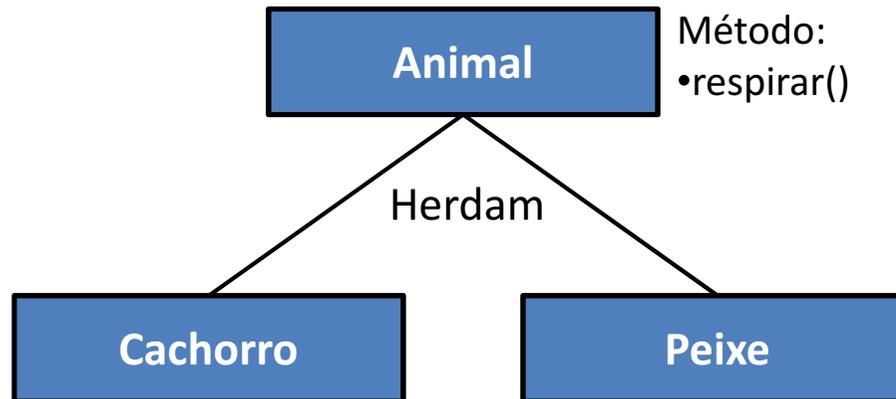


Herança

- Classe genérica, classe base, **superclasse** ou pai:
 - Define variáveis de instância “genéricas” e métodos
- Classe especializada, derivada, **subclasse** ou filha:
 - Especializa, **estende** ou herda os métodos “genéricos” de uma superclasse
 - Define apenas os métodos que são especializados

Polimorfismo

- Situação:



Cachorro e Peixe respiram da mesma forma?

Polimorfismo

- Solução:
 - Utilizar Polimorfismo para **reescrita** do método herdado na subclasse
 - Métodos com o mesmo nome, porém com implementações diferentes
 - Não confundir polimorfismo com sobrecarga de métodos.

Polimorfismo

- Significa “várias formas”
- Habilidade de um mesmo tipo de objeto poder realizar ações diferentes ao receber uma mesma mensagem
- Criação de múltiplas classes com os mesmos métodos e propriedades, mas com funcionalidades e implementações diferentes

Polimorfismo

