

INSTITUTO FEDERAL DO RIO GRANDE DO SUL

GABRIEL H. SCHAEFFER

**LIVRO DE ANÁLISE E PROJETO OO:
RESUMO DO TRECHO QUE SE ESTENDE DAS PÁGINAS 8 ATÉ 16**

**ERECHIM
2025**

1. O papel da abstração na orientação a objetos

Uma abstração permite gerenciar a complexidade e se concentrar nas características primordiais de um objeto.

2. Princípios da OO

2.1 Encapsulamento

Esse é um mecanismo que restringe acesso ao comportamento interno de um objeto. Nesse sistema, os objetos se comunicam recebendo e enviando mensagens.

Interface: representa o que um objeto conhece e o que ele sabe fazer. Define os serviços que ele pode realizar e as mensagens que pode receber. Pode ter várias formas de implementação.

Pelo princípio do encapsulamento, a implementação de um objeto requisitado não importa para o requisitante. A única coisa que este precisa conhecer é a interface do requisitado.

2.2 Polimorfismo

Capacidade de abstrair várias implementações diferentes em uma única interface.

Um objeto pode enviar a mesma mensagem para objetos semelhantes, mas com a interface em comum.

2.3 Herança

Permite que um objeto herde as propriedades de outro objeto. Classes semelhantes são agrupadas em hierarquias. Esse mecanismo facilita o compartilhamento de comportamento entre classes semelhantes.

3.0 Evolução histórica da modelagem de sistemas

O rápido crescimento da capacidade de processamento das máquinas resultou na demanda por softwares cada vez mais complexos. Dessa forma, logo percebeu-se que era necessária uma abordagem mais robusta de desenvolver software.

3.1 1950/60

Os sistemas de software eram simples e desenvolvidos de forma “ad-hoc”. Era a época dos fluxogramas e diagramas de módulos.

3.2 1970

Nessa década começaram a surgir computadores mais avançados e acessíveis. Ou seja, o mercado computacional expandiu. Sistemas mais complexos e modelos mais robustos começaram a surgir. Nesse período surgiu a programação estruturada.

3.3 1980

Aqui, os computadores se tornaram ainda mais avançados e democratizados. A Análise estruturada surge no início desse período.

3.4 1990

Aqui é onde surge o paradigma de orientação a objetos.

3.5 Fim da década de 90

O paradigma OO se consolida. Começam a surgir conceitos como padrões de projeto, componentes, frameworks. Também surge a linguagem UML.

4. Linguagem de modelagem unificada (UML)

É a linguagem padrão para modelagem de sistemas. É uma linguagem visual para modelar sistemas orientados a objetos. Ela é independente de linguagem de programação e processo de desenvolvimento, podendo ser usada com qualquer linguagem e qualquer tipo de modelo.

4.1 Visões de um sistema

Os autores da UML sugerem que o sistema pode ser analisado sob 5 visões independentes.

4.1.1 Visão de casos de uso

Descrição do sistema de um ponto de vista externo.

4.1.2 Visão de projeto

Enfatiza características que dão suporte às funcionalidades externamente visíveis do sistema

4.1.3 Visão de implementação

Gerenciamento de versões do sistema.

4.1.4 Visão de implantação

Distribuição física do sistema.

4.1.5 Visão de processo

Características de concorrência, sincronização e desempenho do sistema.

4.2 Diagramas UML

Um artefato de software são os documentos respectivos à modelagem do sistema em linguagem UML.

Os artefatos gráficos do sistema são feitos utilizando diagramas UML.

Esses diagramas são representações visuais que descrevem a estrutura, o comportamento e as interações de um sistema de software usando elementos padrão.