



Análise e Projeto Orientados a Objeto com UML e Padrões

Parte I

Análise, Projeto, e Processo

Agenda: Introdução, Casos de Usos, Diagramas de Classes

Prof. Alberto Sales

(Mestrando Ciência da Computação – PUCRS)





Aplicando UML, Padrões e APOO

□ Objetivo

- *Desenvolver habilidades práticas na utilização da ASOO para a criação de sistemas de software bem projetados, robustos, e modificáveis*

□ Linguagens OO são um primeiro passo necessário mas insuficiente

□ Outros recursos importantes

- processo de desenvolvimento
- padrões
- UML

□ Ilustrados na prática através de um estudo de caso detalhado





Atribuindo Responsabilidades

- ▮ **Saber a maneira adequada de atribuir responsabilidades a componentes de software é a habilidade mais importante na APOO**
 - Mais difícil de dominar
 - Afeta com mais profundidade a robustez, modificabilidade e reusabilidade do sistema
- ▮ **Padrões GRASP descrevem princípios fundamentais para auxiliar na atribuição de responsabilidades**
- ▮ **Saber identificar objetos ou abstrações adequados é a segunda habilidade mais importante**





O que é Análise e Projeto?

Análise — “o quê”	Projeto — “como”
<i>Investigação do problema e dos requisitos</i>	<i>Descrição de uma solução lógica</i>
Requisitos	Objetos
Casos de uso	Arquitetura
Restrições	Instalação & Operação
Vocabulário	Interface do usuário



Conflito de Terminologias

- Termos “Análise” e “Projeto” não são fixos, mas usados ao longo de um contínuo



- Significados variam de metodologia para metodologia
- Distinção é útil na prática, mas debater definições rígidas não é construtivo



O que é APOO?

- Na essência, considerar um problema e uma solução dentro da perspectiva de objetos, coisas ou conceitos

- O que é AOO?

- Investigação dos objetos de domínio e seus relacionamentos

Descritos no *Modelo de Objetos de Domínio*

- O que é POO?

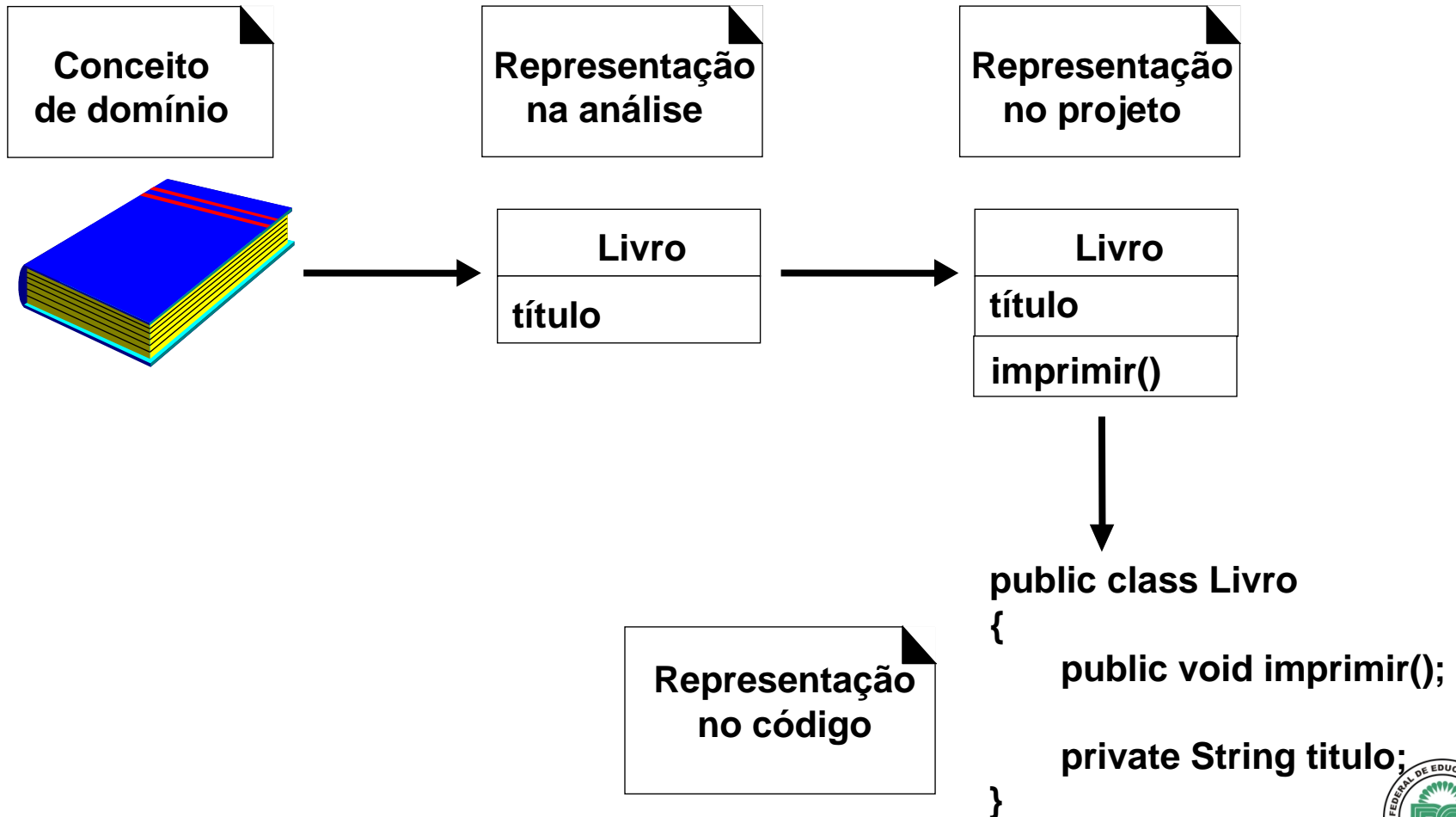
- Elaboração de uma solução lógica em termos de componentes de software e suas colaborações e responsabilidades

Descritos em *Diagramas de Classes* e *Diagramas de Interação*



Representação de um Conceito na APOO

Ex.: O conceito “Livro” em um sistema de biblioteca





Uma Analogia — Organizando os Negócios de uma Empresa

Analogia	APOO	Documentos Associados
Quais são os processos de negócio?	Análise de requisitos	Casos de uso
Quais são os papéis dos empregados?	Análise do domínio	Modelo conceitual
Quem é responsável por o quê? Como eles Interagem?	Atribuição de responsabilidades, projeto das Interações	Diagramas de classes de projeto, diagramas de colaboração



Um Exemplo — Jogo de Dados

- Objetivo: ganha o jogo o jogador que rolar dois dados e tirar sete



- Modelagem na APOO

- Casos de uso

Descrições narrativas de processos do domínio no formato de prosa estruturada

Ex: caso de uso Jogar

Atores: Jogador

Descrição: Este caso de uso começa quando o jogador rola os dados. Se o total dos dados for sete, o jogador ganha; do contrário, ele perde.





Um Exemplo — Jogo de Dados

Modelagem na APOO (cont.)

– Modelo conceitual

Conceitos, atributos, e associações que são considerados importantes no domínio da aplicação



- Um modelo conceitual descreve conceitos do mundo real, não componentes de software!



Um Exemplo — Jogo de Dados

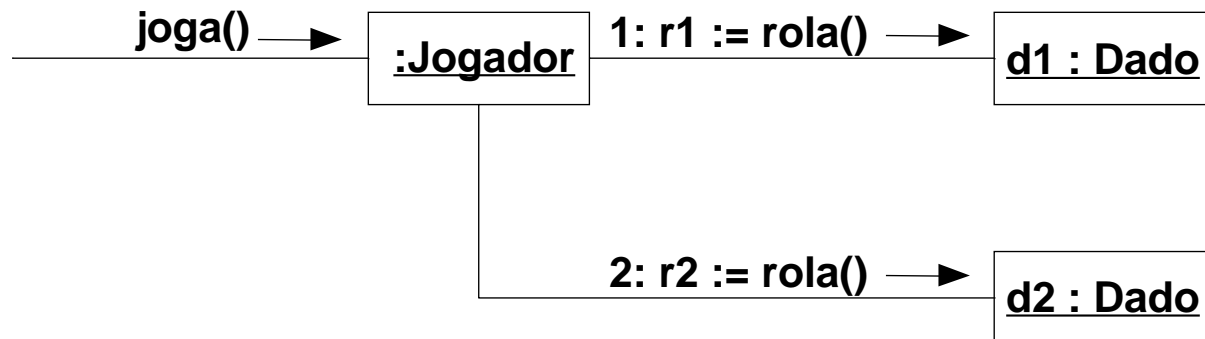
▮ Modelagem na APOO (cont.)

– Diagramas de colaboração

Alocação de responsabilidades para objetos
ilustrando como eles interagem via mensagens

Mostram o fluxo de mensagens entre instâncias e a
invocação de métodos

Ex.:



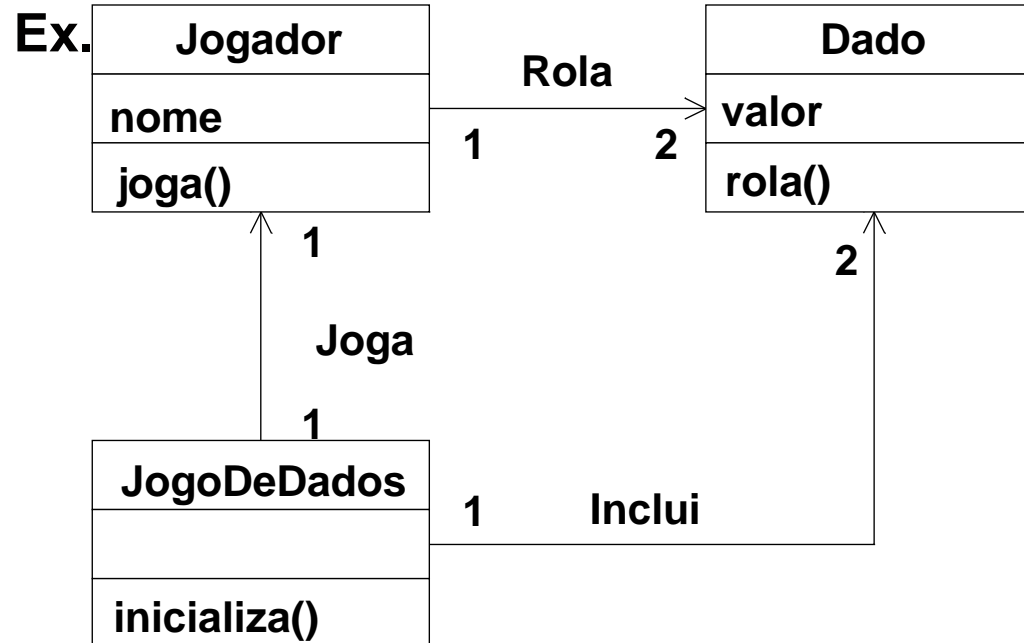


Um Exemplo — Jogo de Dados

- Modelagem na APOO (cont.)
 - Diagramas de classes de projeto

Como os objetos (de software) se conectam?

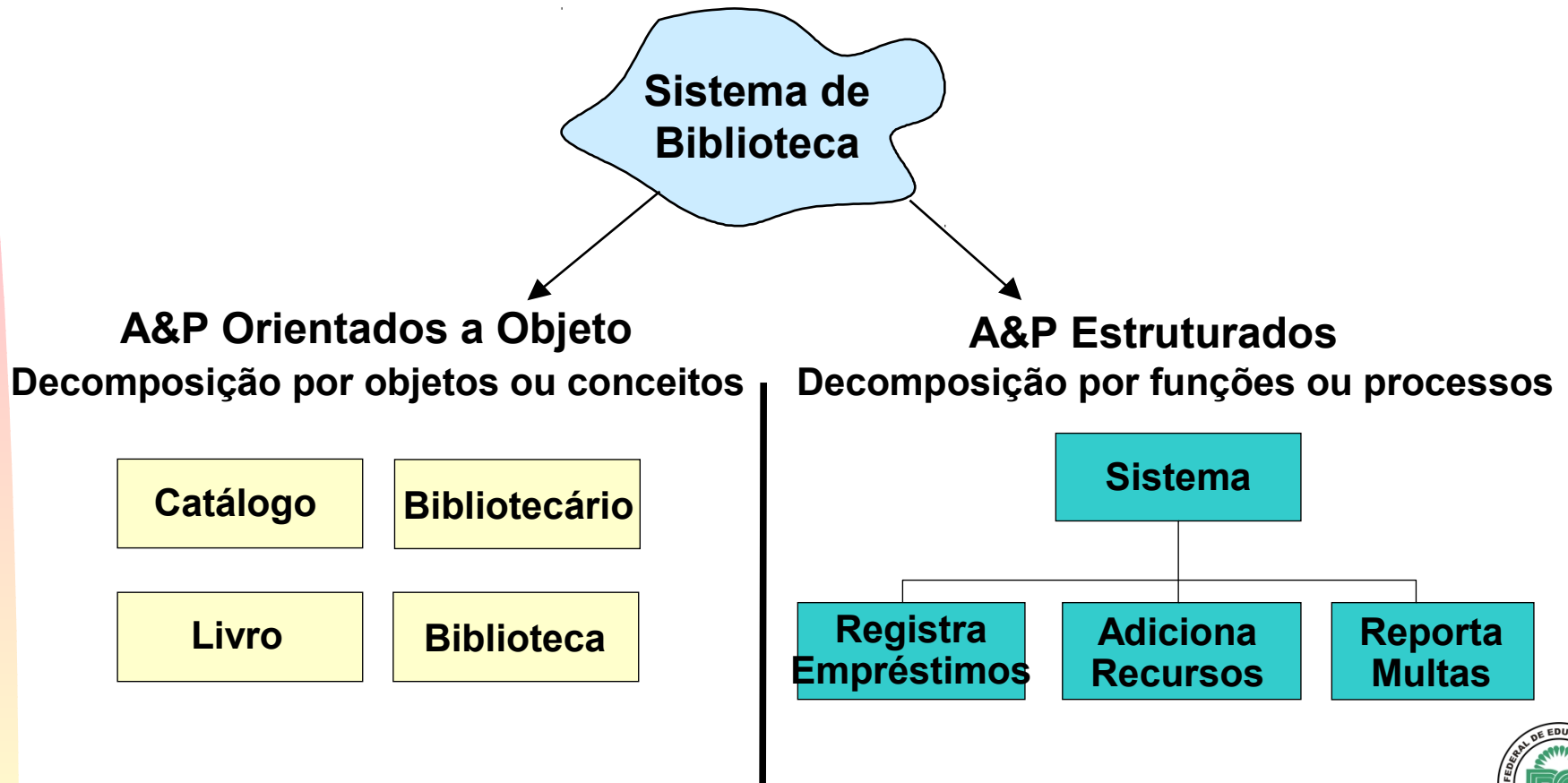
Quais são os métodos de uma classe?





APOO X APE

- Metodologias mais antigas, como Análise e Projeto Estruturados, baseiam-se em outras dimensões de decomposição





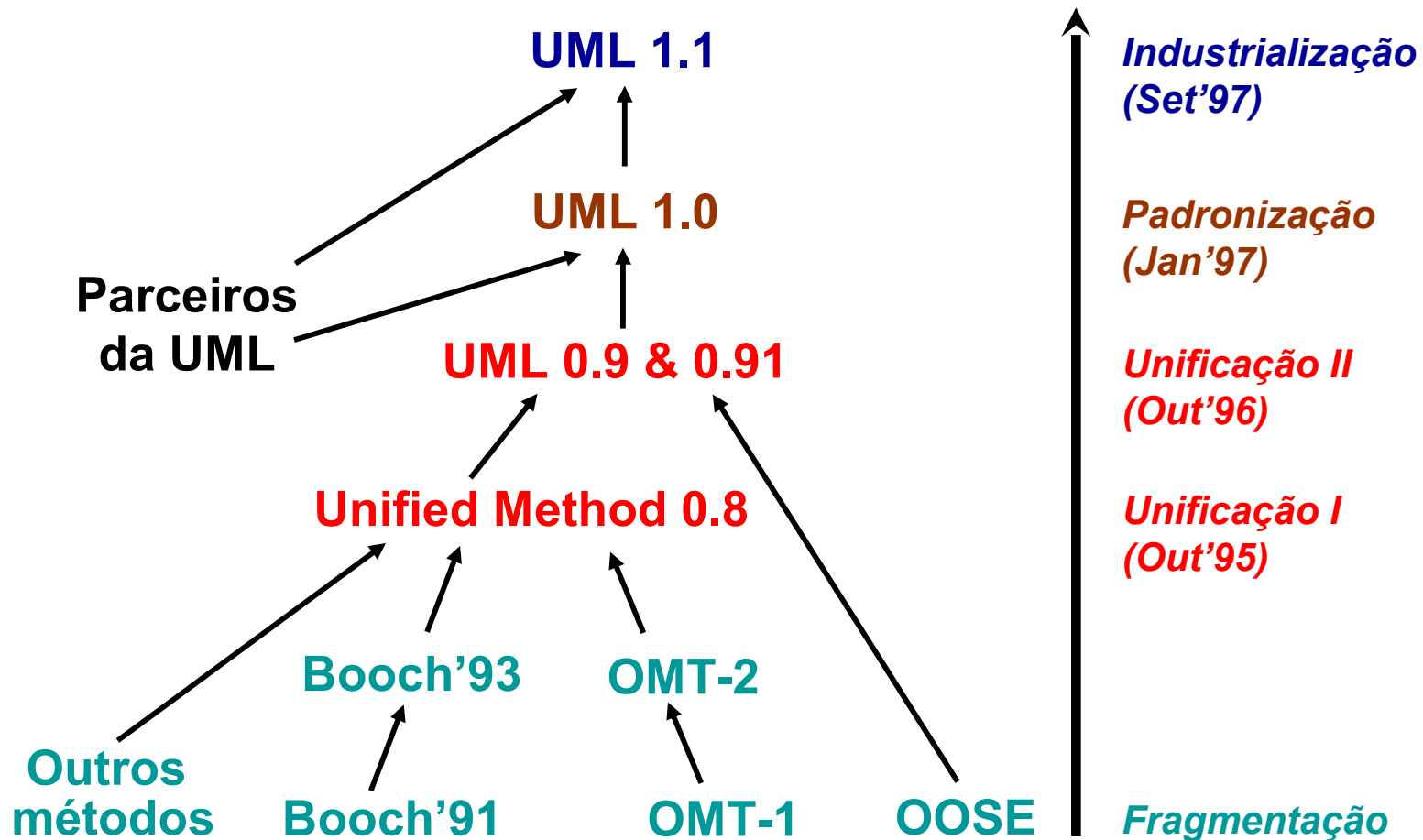
A Linguagem de Modelagem Unificada — UML

- ▮ A UML é a linguagem padrão de diagramação para visualizar os resultados da análise e projeto
- ▮ A notação (a própria UML) é relativamente trivial
- ▮ Muito mais importante: habilidade para modelar com objetos
 - Só aprender a notação UML não ajuda
- ▮ A UML não é
 - um processo ou metodologia
 - APOO
 - regras de projeto





Origem e Evolução da UML





Processo de Desenvolvimento

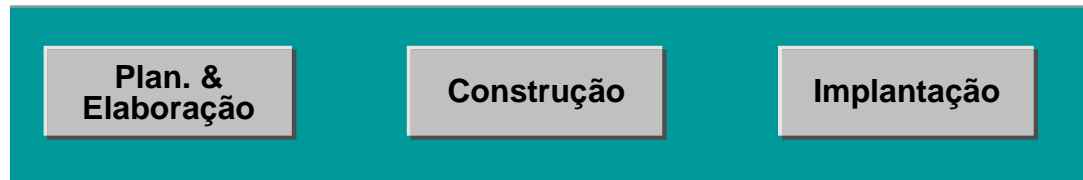
- ▮ Organização das atividades relacionadas à produção e manutenção de sistemas de software
- ▮ Útil, mas um fator de segunda ordem
 - O principal: *equipe qualificada*
- ▮ Boa equipe + bom processo = menor risco
- ▮ O processo racional unificado (RUP), baseado no modelo iterativo, é o processo padrão na indústria





Um Processo Iterativo Simplificado

▮ Simplificação do processo iterativo unificado



▮ Fácil extensão e customização

▮ Não inclui atividades importantes como

- Verificação & validação
- Divisão do trabalho
- Gerência de projeto
- Documentação





Fases

▮ Planejamento e Elaboração

- Concepção inicial, investigação de alternativas, definição de requisitos, etc.

▮ Construção

- Construção do sistema através de múltiplos ciclos de análise, projeto, implementação e teste

▮ Implantação

- Instalação e operação do sistema





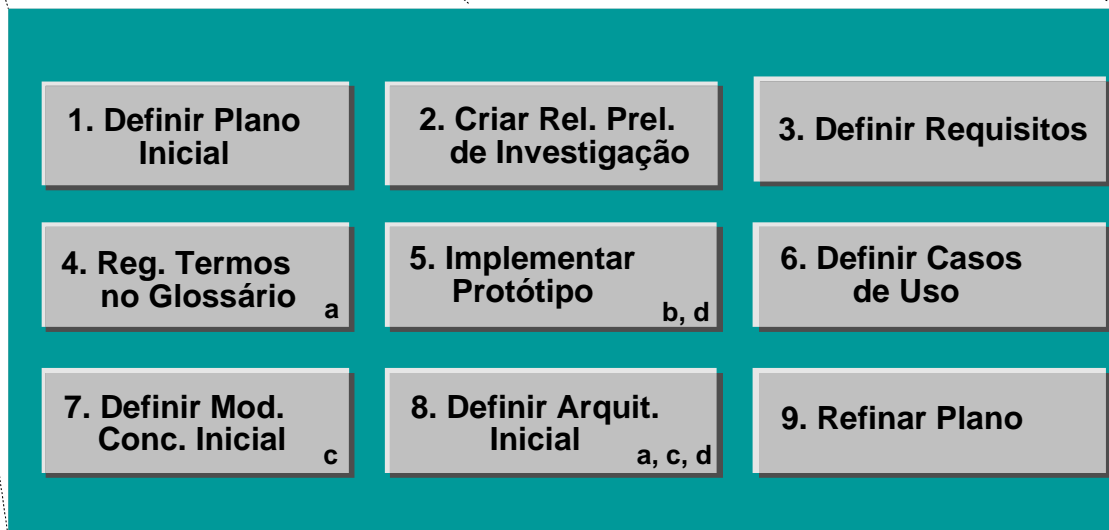
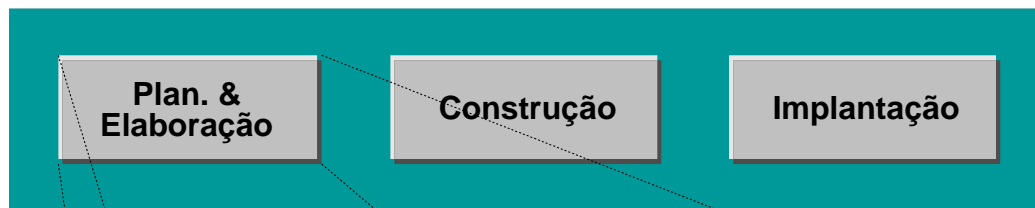
Modelos e Artefatos

- ▮ Um modelo descreve e abstrai aspectos essenciais de um sistema
 - **Modelo estático** (estrutura)
 - **Modelo dinâmico** (comportamento)
- ▮ Modelos são compostos por *artefatos* — diagramas e documentos que descrevem os elementos do modelo
- ▮ Na APOO, a UML é usada para descrever e visualizar os modelos e artefatos produzidos em cada fase do processo de desenvolvimento





Fase de Planejamento e Elaboração



Notas

- a. contínua
- b. opcional
- c. adiável
- d. ordem variada





Fase de Planejamento e Elaboração

▮ Atividades:

1. Definir plano inicial

Prazos, recursos, orçamento

2. Criar relatório preliminar de investigação

Motivação, alternativas, necessidades de negócio

3. Definir requisitos

Especificação declarativa dos requisitos

4. Registrar termos no glossário

Dicionário de termos, regras, restrições

5. Implementar protótipo

Protótipo do sistema para ajudar na definição dos requisitos





Fase de Planejamento e Elaboração

▮ Atividades:

6. Definir casos de uso

Descrição em prosa estruturada dos processos de negócio

7. Definir modelo conceitual inicial

Objetos de domínio e seus relacionamentos

8. Definir arquitetura inicial

Principais subsistemas e suas dependências

9. Refinar plano

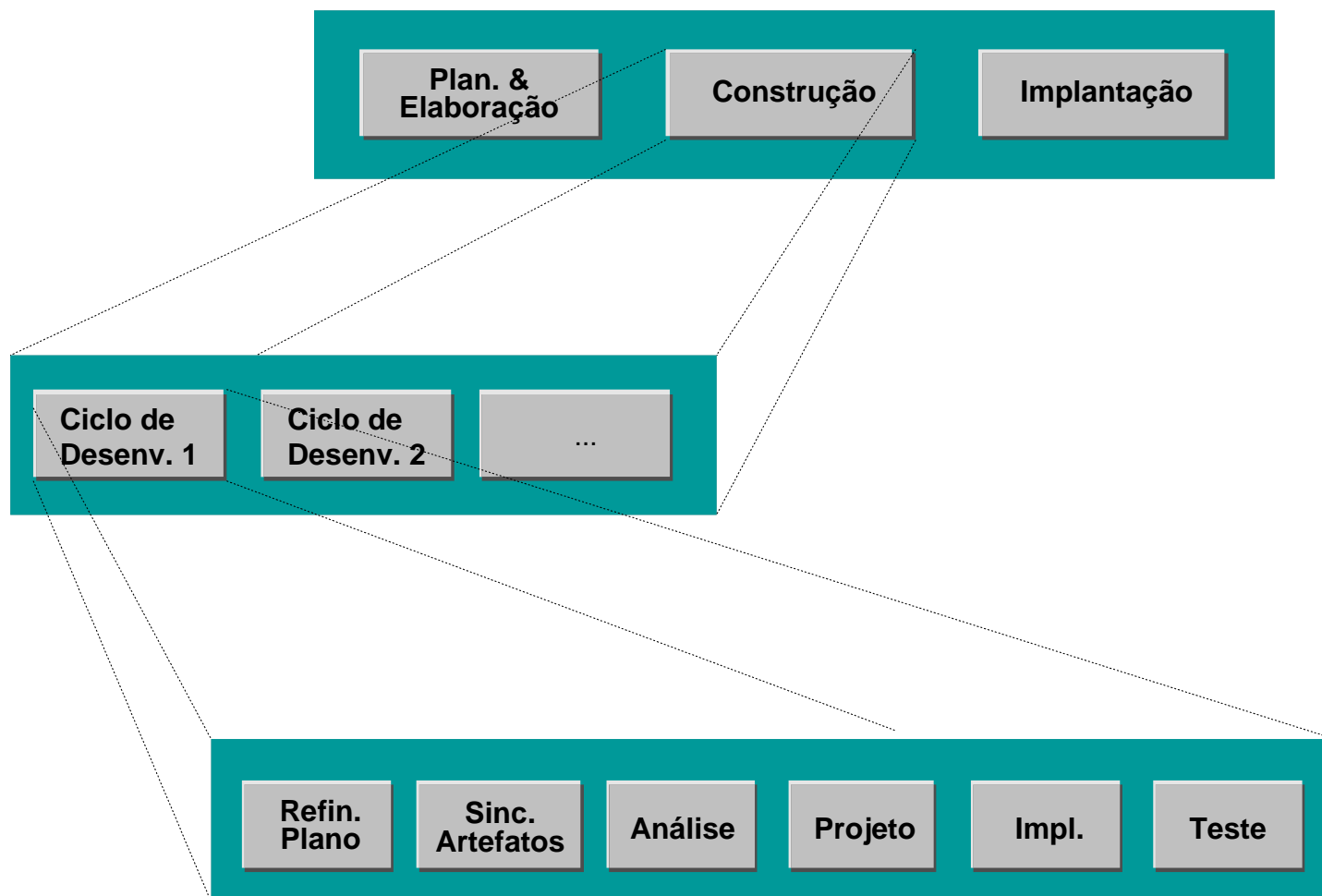
▮ Atividades não lineares

- Ex.: 7, 4, 6 em paralelo





Fase de Construção





Fase de Construção

- ▮ **Repetição de ciclos de desenvolvimento**
 - **Construção progressiva do sistema até atingir uma versão que satisfaça corretamente os requisitos**

- ▮ **Atividades típicas de cada ciclo:**
 - 1. Refinar plano**
 - 2. Sincronizar artefatos**
 - 3. Análise**
 - 4. Projeto**
 - 5. Implementação**
 - 6. Teste**





Ciclos de Desenvolvimento

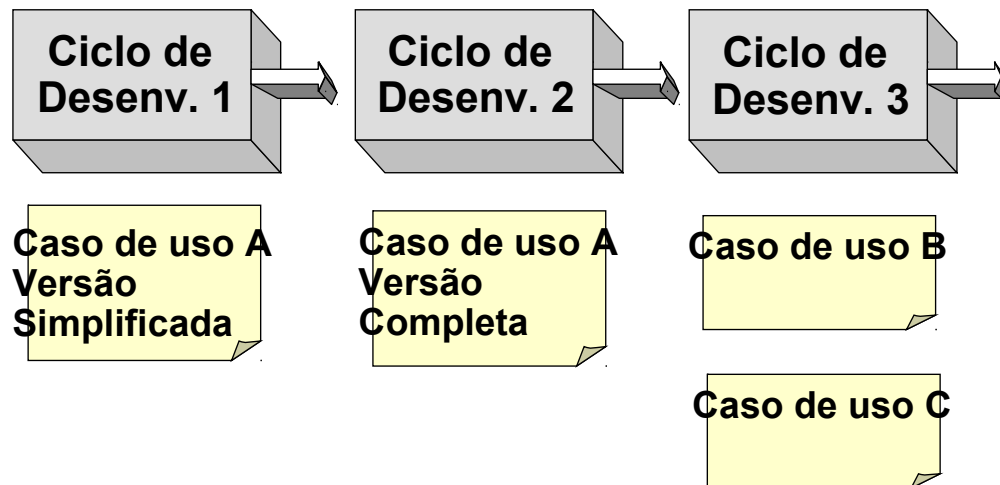
- ▮ Cada ciclo implementa um conjunto reduzido de requisitos, adicionando novas funções ao sistema
 - Crescimento incremental, através de expansões e refinamentos sucessivos
- ▮ Ciclos com tempo fixo de duas a oito semanas
- ▮ Vantagens:
 - Evita complexidade excessiva
 - Antecipa *feedback* dos usuários



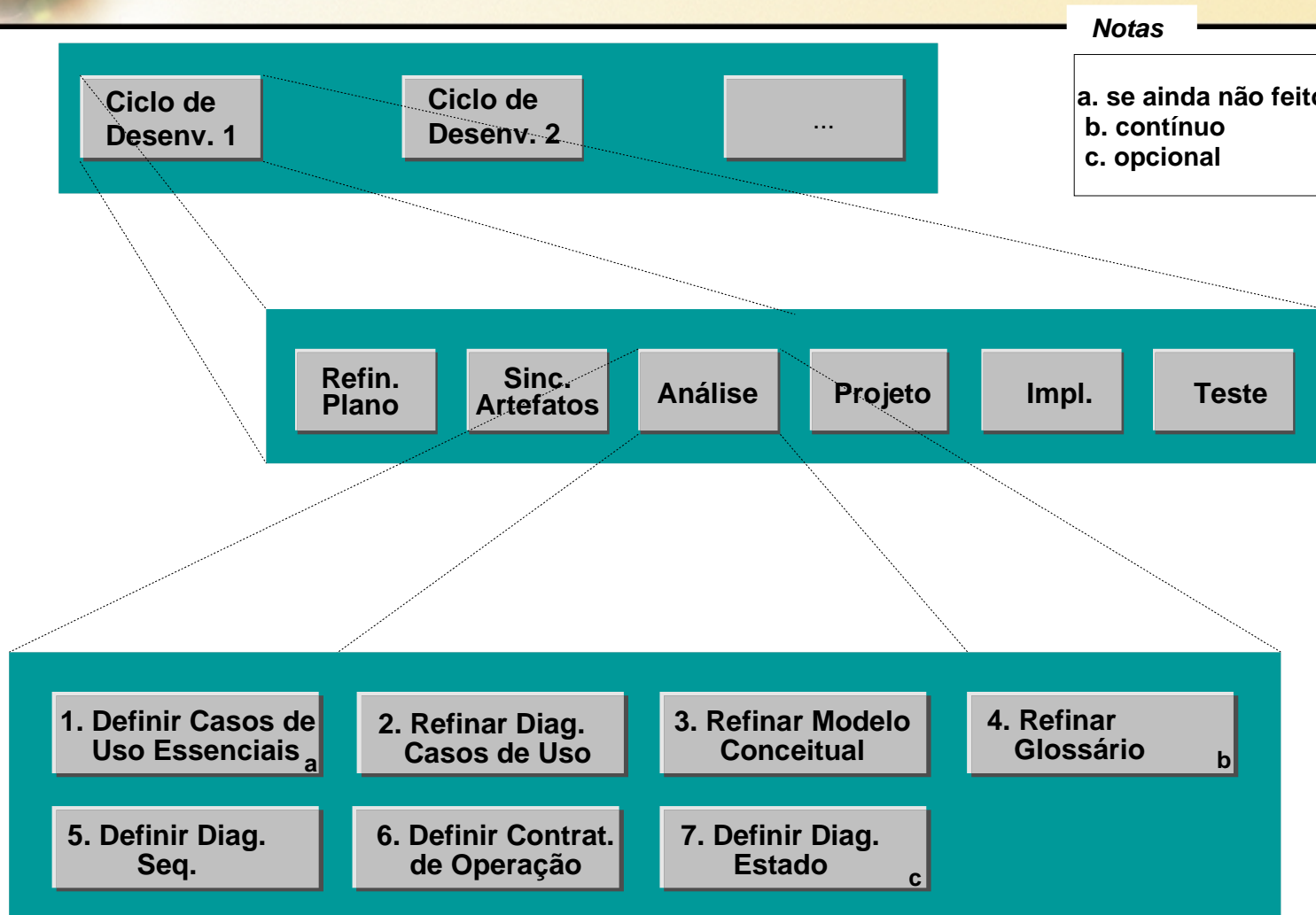


Ciclos de Desenvolvimento e Casos de Uso

- Um ciclo deve atacar um ou mais casos de uso, ou versões simplificadas de casos de uso
- Casos de uso mais relevantes devem ser atacados nos primeiros ciclos
 - Prioridade para serviços com grande influência na arquitetura do sistema ou de alto risco



Análise



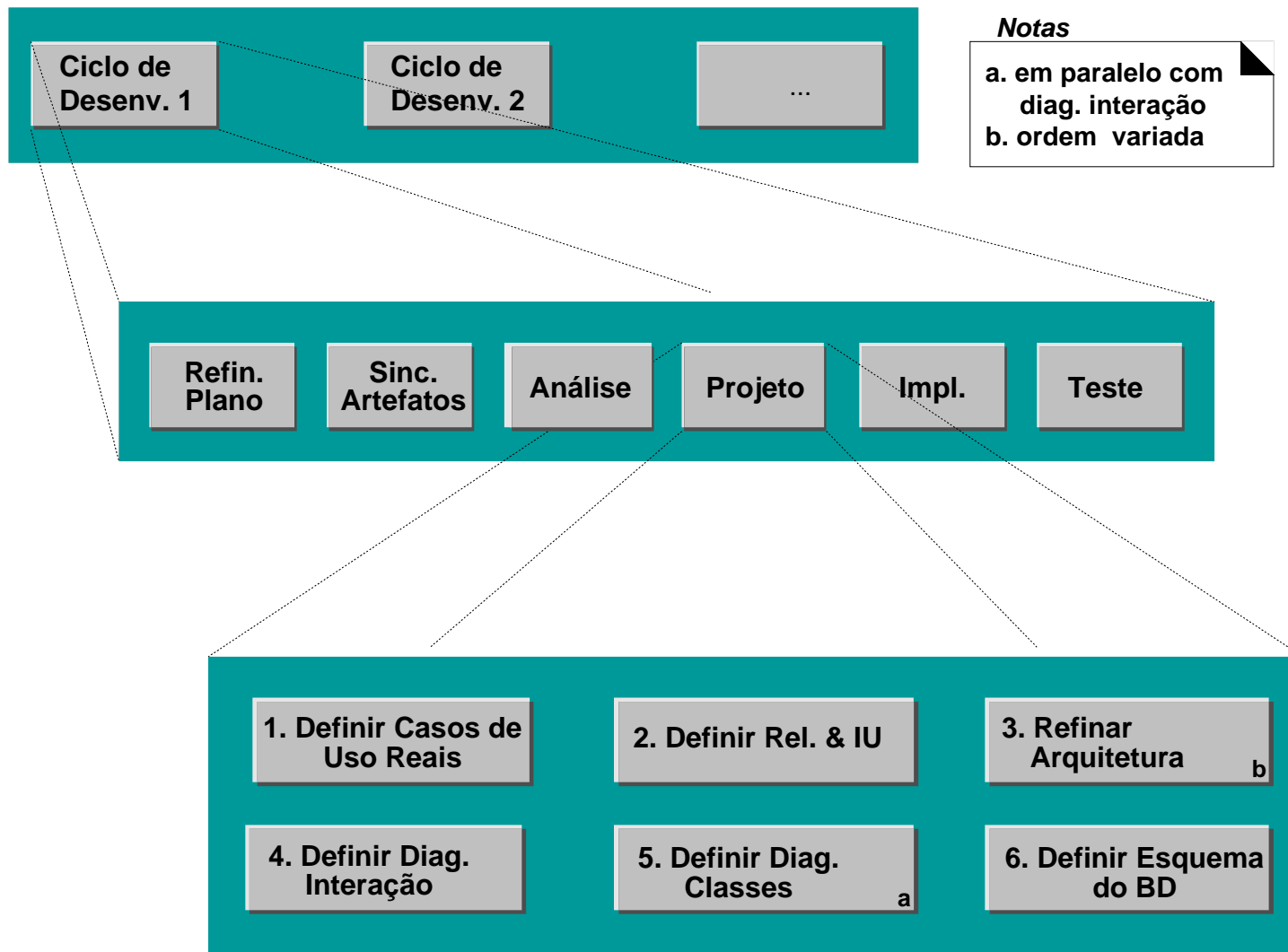


Análise

- ▮ **Subatividades:**
 - 1. Definir casos de uso *essenciais***
 - 2. Refinar diagramas de casos de uso**
 - 3. Refinar modelo conceitual**
 - 4. Refinar glossário**
 - 5. Definir diagramas de seqüência do sistema**
 - 6. Definir contratos de operação**
 - 7. Definir diagramas de estado**



Projeto





Projeto

▮ Subatividades:

1. Definir casos de uso *reais*
2. Definir relatórios e interfaces com o usuário
3. Refinar arquitetura do sistema
4. Definir diagramas de interação
5. Definir diagramas de classes de projeto
6. Definir esquema do banco de dados