Projeto da Lógica de Negócio

Instituto Federal do Espirito Santo – Campus Colatina
Disciplina: Projeto de Sistemas
Professor: Giovany Frossard Teixeira

Introdução

- Todas as outras camadas são derivadas ou dependentes dela.
- Os modelos de caso de uso e de classe da fase de análise são o ponto de partida para sua construção.
- A versão inicial é uma cópia do modelo de classe da análise, durante a fase do projeto este modelo é refinado incorporando informações para a implementação (métodos, visibilidade e navegabilidade), além de mudanças para tratar requisitos não funcionais como usabilidade e desempenho.

O projeto da Lógica de Negócio

- O projeto da Lógica de Negócio é composto da lógica do negócio vinda da camada de domínio do problema e das funcionalidades descritas pelos casos de uso.
- Dessa forma o desafio é saber: Quais classes vão incorporar as funcionalidades descritas nos casos de uso? Há 2 formas de distribuir a responsabilidade de execução dos casos.
  - Ao longo dos objetos da camada de domínio do problema.
  - Considerar que a CDP (camada do domínio do problema) que tem haver com as classes identificadas na analise e a camada de gerencia de tarefas cuida das funcionalidades descritas nos casos de uso, tratando a lógica da aplicação.

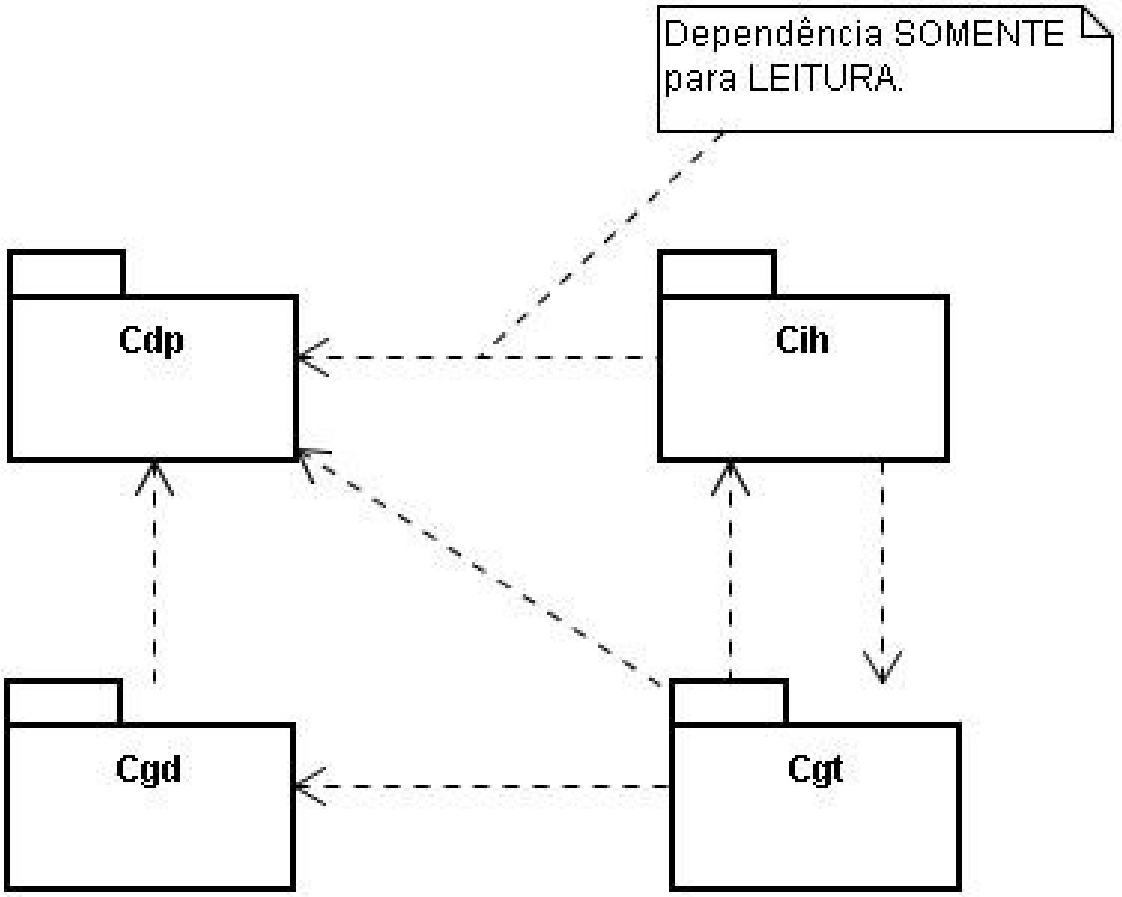
O projeto da Lógica de Negócio

- Os **casos de uso** deverão dar origem a operações e consultas do sistema, que vão estar disponíveis a partir da interface do sistema com o mundo externo.
- Temos a lógica do domínio do problema e a lógica da aplicação que se refere a lógica presente nos casos de uso.
- É importante definir onde posicionar os métodos que vão cumprir esses diferentes tipos de responsabilidade.

O projeto da Lógica de Negócio

- Baseado na estratégia de uso de Componente/camada de Domínio do Problema e componente/camada de Gerência de Tarefas chegamos ao modelo proposto por Coad e Yourdon (1991):
  - Componente de Domínio do Problema
  - Componente de Interação Humana
  - Componente de Gerência de Tarefas
  - Componente de Gerência de Dados
- Pode-se usar quantas camadas/componentes quanto necessário/desejado.

Modelo proposto por Coad e Yourdon



Componente do Domínio do Problema

- Alterações Básicas do Diagrama de Classes da Fase de Análise para o Diagrama de Classes do CDP:

1) Tipos de dados dos atributos (verificar os tipos disponíveis na linguagem a ser utilizada).
2) Navegabilidade nas associações (a navegabilidade dupla pode não ser interessante, principalmente por questões desempenho).
3) Visibilidade dos atributos e associações (atributos prioritariamente privados e não há a necessidade de apresentação de métodos gets e sets);.
4) Adição dos métodos.
5) Eliminação das classes associativas (substituição por classes normais).
6) Reutilizar classes já programadas em projetos.
7) Ajustar hierarquia de generalização e especialização (herança múltipla, por exemplo, não tem suporte em Java).
8) Ajustar modelo para melhora o desempenho.
9) Ajustar modelo para facilitar as interfaces (tipos enumerados, por exemplo).
10) Incorporar aspectos relacionados a segurança.

Componente de Gerência de Tarefas

- **Padrão Modelo do Domínio:** a lógica da aplicação (execução de casos de uso) é feita por classes do domínio o problema, eliminando assim a necessidade do Componente de Gerência de Tarefas.
  - Este modelo implica em considerar uma classe controladora do sistema de domínio do problema, relacionando-a a todos as classes do domínio.
  - O fluxo sempre inicia em uma instância desta classe controladora.
  - Esta classe recebe as requisições da interface para tratá-las invocando os métodos necessários posicionados nas classes de domínio do problema.
  - Essa classe é responsável pela delegação que consiste em capturar uma mensagem em um objeto e reenviar essa mensagem para outro objeto associado ao primeiro que seja capaz de tratar a mensagem.

Componente de Gerência de Tarefas

- **Padrão Camada de Serviço:** o Componente de Gerência de Tarefas (CGT) fica responsável por tratar a lógica da aplicação.
  - Controla o fluxo de eventos dos casos de uso.
  - Pode-se ter um gerenciador de tarefas para cada caso de uso.
  - Pode-se definir uma classe controlador para todo o sistema, neste caso, cada caso de uso da origem a uma operação nesta classe.
  - Normalmente uma solução intermediária conduz a melhores resultados.

Dúvidas?

Bibliografia

COAD, Peter; YOURDON, Edward. *Object-Oriented Design*. Upper Saddle River: Yourdon Press, 1991.