

Projeto Detalhado de Software

Coesão e Acoplamento

Referências

- “TDD e sua influência no acoplamento e coesão”

<http://blog.caelum.com.br/tdd-e-sua-influencia-no-acoplamento-e-coesao/>

- “Cohesion and Coupling”

<http://c2.com/cgi/wiki?CouplingAndCohesion>

- “Software Design (lectures of CPSC 333)”

<http://pages.cpsc.ucalgary.ca/~eberly/Courses/CPSC333/Lectures/topics.html>

- “Code complete”, Steve McConnell, Capítulo 5 e 7
(disponível na biblioteca)

Motivação

- Sistemas de software são complexos demais para serem expressos em apenas um único módulo
 - Lembre-se, aceite: seu cérebro é limitado
- Software é sujeito a muitas mudanças
 - A capacidade de um software de se adaptar a mudanças é essencial para sua própria longevidade

Coesão

- Coesão de um módulo/componente é o grau que suas responsabilidades formam uma **unidade significativa**
 - Alta coesão = responsabilidades afins
 - Baixa coesão = responsabilidades desconectadas
- Coesão é inversamente proporcional ao número de responsabilidades **distintas** que um módulo/componente possui

Coesão

- Unidade significativa de responsabilidades
 - Depende do ponto de vista, da experiência do time, do domínio
 - Adotando um processo iterativo e incremental de desenvolvimento, módulos com pouca coesão podem ser identificados

Tipos de Coesão

- Coincidente **(pior)**
 - Nenhuma correlação, apenas um agrupamento de responsabilidades

```
public Class VariasCoisas {  
    public void consertaCarro();  
    public void fazBolo();  
    public void levarCachorroPassear();  
    public void estudarParaProvaFinal();  
}
```

Tipos de Coesão

- Temporal **(evitar)**
 - As responsabilidades são acionadas em períodos ou frequências semelhantes

```
public Class TardeDaNoite {  
    public void colocarLixoFora();  
    public void soltarOCachorro();  
    public void escovarOsDentes();  
    public void jogarUmaRodadaDeWarcraft();  
}
```

Tipos de Coesão

- Procedural **(usar com moderação)**
 - As responsabilidades são acionadas em ordem, mas não compartilham dados

```
public Class OrganizarJantar {  
    public void prepararPernilForno();  
    public void ligarParaConvidados();  
    public void cortarVerduras();  
    public void arrumarAMesa();  
}
```


Tipos de Coesão

- Comunicacional (**usar**)
 - As responsabilidades são compartilham dados comuns ou a mesma estrutura de dados (que está escondida)

```
public Class Biblioteca {  
    public void buscarPorTitulo();  
    public void buscarPorAutor();  
    public void buscarPorEditora();  
    public void buscarPorMaisVendidos();  
}
```

Tipos de Coesão

- Sequencial (**usar**)
 - As responsabilidades fornecem dados de saídas para as demais, de forma sequencial

```
public Class LavarCarro {  
    public void enxaguarLataria(); // x2  
    public void passarEspumaDetergente();  
    public void secarComFlanela()  
    public void passarCera();  
    Public void polir();  
}
```

Tipos de Coesão

- Funcional (**SUPRA SUMO**)
 - Módulos possuem responsabilidades muito bem definidas e enxutas
 - Apenas um “problema”

```
public Class EnxagueDeLataria {  
    public void enxaguar(Carro carro);  
}
```

Coesão

- **Exercício (7 min):** Distribuir responsabilidades da interface
 - Identifique grupos de responsabilidades
 - Separe em novas interfaces e escolha bons nomes

```
public interface SuperMega {  
    public void iniciaPilhaComandos();  
    public void iniciaRelatorio();  
    public void iniciaDadosGlobais();  
    public void empilhaComando(Object comando);  
    public void formataRelatorio(Object relatorio);  
    public void imprimirRelatorio(Object relatorio);  
    public void desempilhaComando();  
    public void finalizaPilhaComandos();  
    public void finalizaDadosGlobais();  
}
```

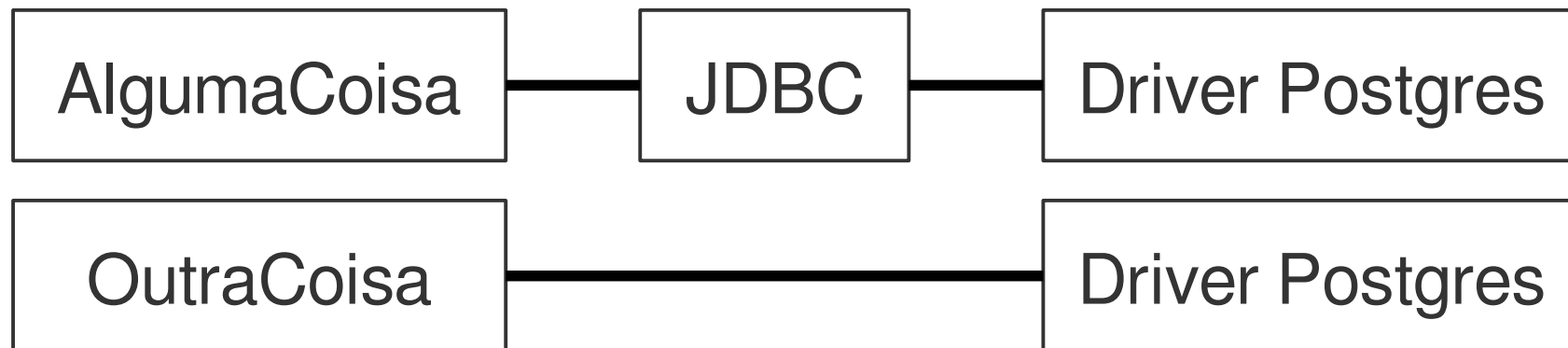
**SE O MÓDULO É ESPERTO,
O PROJETO É BURRO**

Acoplamento

- Acoplamento é o grau que um módulo depende de cada um dos outros módulos de um sistema
 - Baixo acoplamento = poucas dependências
 - Alto acoplamento = muitas dependências
- A necessidade de mudança em um módulo pode afetar os demais dependentes
- Muitas dependências aumentam a probabilidade de mudança de um módulo
- Acoplamento entre módulos é o grau de interdependência mútua

Acoplamento

- Acoplamento não deve ser avaliado apenas pela contagem de dependências
- A probabilidade de mudança também deve ser considerada



**QUAL A PROBABILIDADE DE MUDAR O SGBD DE POSTGRES PARA MYSQL?
QUAL A PROBABILIDADE DE DEIXAR DE USAR UM BD RELACIONAL?**

Critérios de Avaliação de Acoplamento

- Tamanho
 - Número de conexões entre módulos
 - Métodos e quantidade de parâmetros influencia
 - Menos é mais elegante
- Visibilidade
 - Proeminência da conexão entre módulos
 - Exemplo: Passar dados por parâmetros VS. Usar dados globais
- Flexibilidade
 - Facilidade para estabelecer conexões com módulos
 - Exemplo: Passar dados por parâmetro VS. Passar objeto por parâmetro

Tipos de Acoplamento

- Conteúdo **(PIOR)**
 - Módulo A conhece a estrutura interna do Módulo B
 - Quebra do encapsulamento da informação
- Dados/recursos comuns concorrentes **(EVITAR)**
 - Dados globais são MAUS
 - Exemplo: acesso a periféricos
- Controle **(USAR COM MODERAÇÃO)**
 - Módulo A usa mensagens para controlar os dados/comportamento do Módulo B

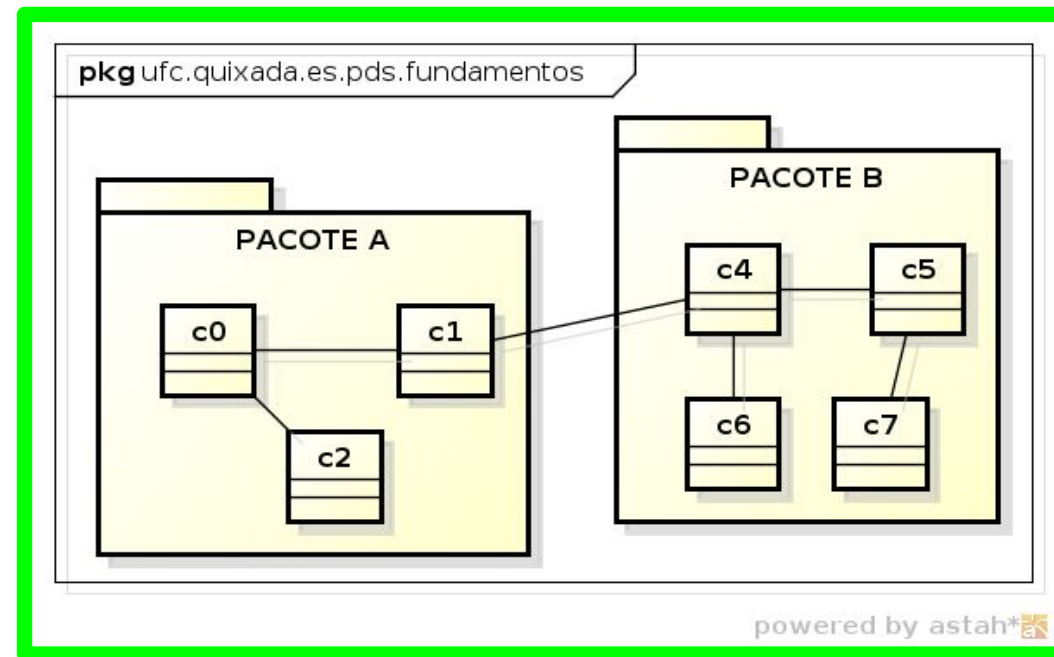
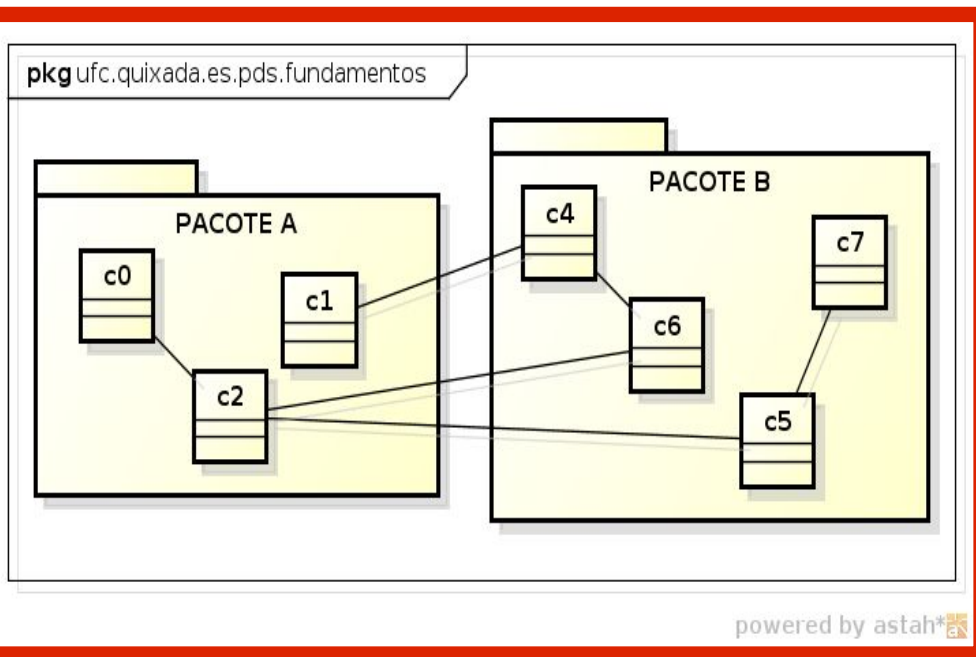
Tipos de Acoplamento

- Estrutura de Dados (**USAR**)
 - Módulo A passa uma estrutura de dados para o Módulo B, que precisa apenas de uma parte
 - Pode ocorrer impacto se a forma de acessar a parte (dentro da estrutura completa) for alterada
- Parâmetros e Troca de Mensagens (**SUPRA SUMO**)
 - Módulos A e B trocam mensagens. Dados são passados por parâmetro, de preferência dados de tipos básicos (primitivos ou built-in da linguagem)

**MÓDULOS POSSUEM POUCOS
MAS BONS AMIGOS**

Coesão e Acoplamento

- Interrelações entre pacotes/componentes do sistema
 - Maximizar COESÃO de cada pacote
 - Minimizar ACOPLAMENTO entre pacotes



Conclusão

- O desenho do software deve focar em
APROXIMAR RESPONSABILIDADES SIMILARES
SEPARAR RESPONSABILIDADES DISTINTAS
- Princípio fundamental de projeto de software é
INDEPENDÊNCIA FUNCIONAL DE MÓDULOS
buscado através de constante (iterativo e incremental)
AUMENTO DE COESÃO
DIMINUIÇÃO DE ACOPLAMENTO