

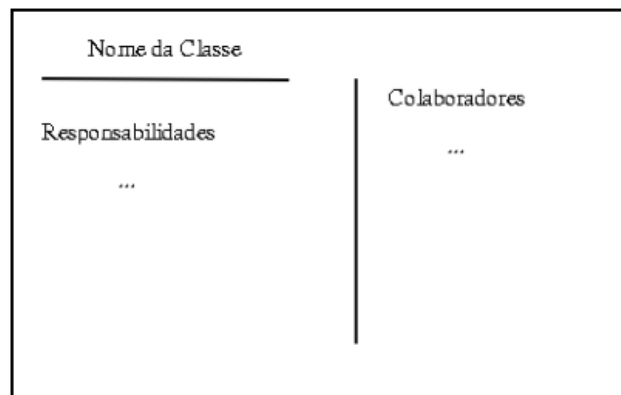
# Aula 18 - Engenharia de Requisitos

## Cartões CRC (Classe-Responsabilidades-Colaboradores)

É baseada nos três atributos principais de um objeto na fase de projeto: nome da classe, suas responsabilidades e seus colaboradores:

- O nome da classe deve descrever o objeto no contexto geral do ambiente. Deve-se tomar um certo cuidado e gastar um pouco de tempo para escolher o conjunto certo de palavras para descrever o objeto.
- As responsabilidades devem identificar os problemas a serem resolvidos e não as soluções. Identificando-se os problemas fica fácil escolher entre as soluções possíveis. Novamente deve-se tomar cuidado com o conjunto de palavras usadas. As frases devem ser curtas e concisas.
- Os colaboradores de um objeto são os objetos para os quais este irá enviar mensagens a fim de satisfazer as suas responsabilidades.

Para facilitar a modelagem, Cunningham inventou os cartões Classe-Responsabilidades-Colaboradores(CRC). Esses cartões servem, também, como meio de documentação do projeto. Os cartões CRC são pedaços de papel grosso medindo 10x15cm e contém o nome da classe, as responsabilidades e os colaboradores.



Posição dos cartões:

- Quando duas classes são mútuas colaboradoras, seus cartões devem ser colocados levemente sobrepostos.
- Quando uma classe é colaboradora de outras classes (a classe é usada pelas outras), seu cartão deve ficar abaixo dos cartões das outras classes.
- Se há uma relação de herança, deverá haver uma sobreposição quase completa dos cartões.

Para achar as classes, suas responsabilidades e seus colaboradores deve-se seguir as seguintes etapas:

1. Definição das classes e das estruturas de dados de cada classe
2. Definição das responsabilidades de cada classe

### 3. Definição dos colaboradores de cada classe

#### Classes

- Num problema a ser resolvido, as diversas classes podem ser identificadas como as entidades que existem no problema, por exemplo, aluno, turma, professor.
- São classes, também, as estruturas de dados utilizadas para resolver o problema, bem como os dispositivos (físicos e virtuais) a serem acessados. Por exemplo: interface, arquivo, controlador de tempo.

#### Responsabilidades

- As responsabilidades são definidas como sendo os problemas que as classes devem resolver.
- São, a grosso modo, as funções das classes.
- As responsabilidades são os problemas a serem resolvidos.

#### Colaboradores

- Os colaboradores são as classes que ajudam uma classe a resolver as suas responsabilidades.
- São as classes que vão prestar serviços à classe.
- A classe que está sendo analisada solicita serviços a outras classes e estas, por sua vez, prestam estes serviços, ajudando a resolver os problemas.

A Sessão de cartões CRC , inclui uma simulação física do sistema e a execução de cenários. Para se começar a sessão, deve-se considerar que todas as ideias são boas em potencial, se ter uma flexibilidade e dinâmica de grupo.

Para se fazer a sessão, é necessário de 5 a 6 pessoas, compostas por 1/2 especialistas, 2 analistas, 1 designer e 1 líder. Durante a sessão, todos os membros do grupo são responsáveis por segurar, mover e anotar uma ou mais cartas conforme as mensagens voam pelo sistema. Os membros do grupo criam, complementam e empilham os cartões durante a análise de cenários, e um escriba de sessão escreve os cenários.

O processo de fazer a sessão, é iniciar com um Brainstorming, após isso identificar as classes coerentes com o sistema e finalizar com uma execução do cenário (dramatização).

#### Pontos positivos

- Os cartões e o exercício não são ameaçadores e informais
- Proporcionar um bom ambiente para trabalhar e aprender.
- Barato, portátil, flexível e rapidamente acessível
- Permita que os participantes experimentem em primeira mão como o sistema funcionará
- Ferramenta útil para ensinar às pessoas o paradigma da orientação a objetos

#### Pontos negativos:

- Fornece apenas ajuda limitada nos aspectos de design.
- Não tem poder de notação suficiente para documentar todos os componentes necessários de um sistema.
- Não especifica detalhes de implementação.
- Não fornece uma visão dos estados através dos quais a transição de objetos durante seu ciclo de vida.