Análise e Desenvolvimento de Sistemas
Engenharia de Software I

Revisão de Análise Orientada a Objetos

Parte I

Slides da disciplina Laboratório de Engenharia de Software
Curso de Análise e Desenvolvimento de Sistemas
Faculdade de Tecnologia de São Caetano do Sul – FATEC SCS
Prof. Marco Antonio Furlan de Souza
marco.furlandesouza@fatec.sp.gov.br

Agenda

- Análise de Sistemas
- Análise Orientada a Objetos
- Métodos para identificar classes
- Identificação de classes com fichas CRC

Agenda

- Análise de Sistemas
- Análise Orientada a Objetos
- Métodos para identificar classes
- Identificação de classes com fichas CRC

Análise de Sistemas

- **Conceitos**
  - **Lembrando**, o **processo** de **desenvolvimento** de **software inicia-se** com a **elaboração** dos **requisitos** do software;
  - Os **requisitos de** um **software** podem ser **classificados** como **requisitos funcionais** e **requisitos não funcionais**;
  - **Esta elaboração** dos **requisitos envolve** "**passar** a **limpo**" os **conceitos** existentes nas **mentes** das **pessoas que**, por natureza, **não são formais ou organizados**;
  - O **resultado final** da **engenharia de requisitos** deve ser um **documento formal**, contendo as **especificações dos requisitos do software**;
  - Esta se **inicia** com o **entendimento** dos **requisitos funcionais** do sistema utilizando-se alguma **técnica**, **como**, por exemplo, **Casos de Uso**.

Análise de Sistemas

- **Conceitos**
  - Em **seguida**, com o **entendimento claro** dos **requisitos funcionais** do sistema, é **necessário analisar** o **problema** em **questão** e **propor** uma **solução** com a **estruturação** de um **software** que **resolva** o **problema**;
  - Este **processo** de **análise** do **sistema** consiste em **adotar** um **método** que permita **estudar** um **sistema complexo** pela sua **decomposição** em **partes** mais **simples**, **de acordo com** algum **paradigma**;
  - Um **paradigma** representa uma **forma** como os **métodos** e **ferramentas** são **empregados** na análise. **Exemplos**:
    - Paradigma da **orientação a objetos**;
    - Paradigma **clássico** (**estruturado**);
    - Paradigma **funcional**;
    - Paradigma **lógico**.

Agenda

- Análise de Sistemas
- Análise Orientada a Objetos
- Métodos para identificar classes
- Identificação de classes com fichas CRC

Análise Orientada a Objetos

FATEC
Faculdade de Tecnologia do Estado de São Paulo

- **Conceitos**
  - No **paradigma** da **orientação** a **objetos**, um **sistema** ou **software** é **modelado** e **construído** a **partir** de **partes menores** denominadas **objetos**;
  - **Objetos** são **elementos** que **possuem** em seu **interior**:
    - Um **conjunto** de **dados** cujos **valores** descrevem o **objeto**;
    - Um **conjunto** de **operações denominadas** de **métodos** que permitem **manipular** o **objeto** e também **invocar operações de outros objetos**.
  - A **execução** de um **sistema** ou **software orientado** a **objetos** é o **resultado** da **colaboração** entre os **objetos** que **implementaram** esse sistema, **por meio** da **execução** de **métodos** entre si;
  - Quando se **executa** um **método** de um **objeto**, diz-se que se "**enviou** uma **mensagem** a este **objeto**".

Análise Orientada a Objetos

FATEC
Faculdade de Tecnologia do Estado de São Paulo

- **Conceitos**
  - Assim, **neste paradigma**, a **complexidade** de um **sistema** é **lidada** com a **decomposição** do **sistema** em **partes menores** – os **objetos**;
  - **Cada objeto** possui um **propósito dentro** do **sistema** que, **depois**, por um **processo de composição**, **acaba** por **formar** o **sistema final**.;
  - O **objetivo** da **análise orientada** a **objetos** é **identificar** os **objetos relevantes** no **domínio** do **problema** em estudo e **identificar** também **padrões** e **relacionamentos entre eles**, **criando agrupamentos** que podem se **tornar arquiteturas**.
  - Com a **arquitetura** do **sistema definida**, pode-se **proceder** à sua **implementação**.

Análise Orientada a Objetos

FATEC
Faculdade de Tecnologia do Estado de São Paulo

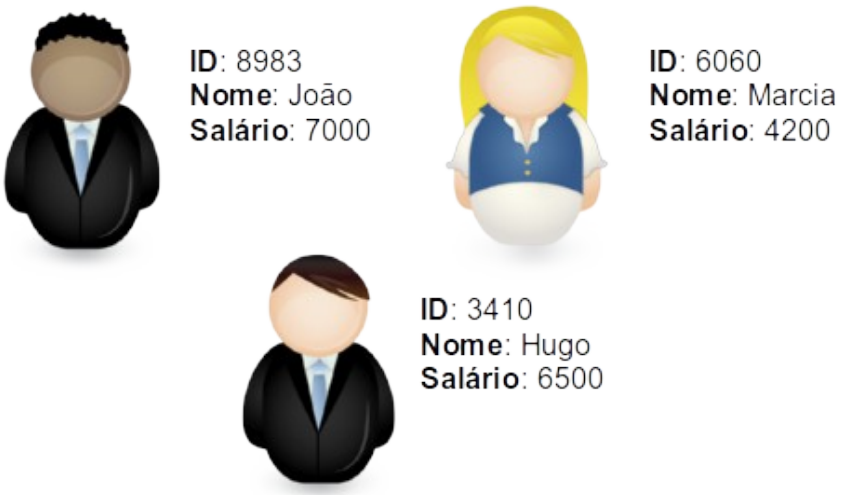
- **Conceitos**
  - **Evolução** das técnicas de análise e projeto orientadas a objetos
    - Década de 1990:
      - Métodos **Booch**, **OMT** (*Object Modeling Technique*) e **OOSE** (*Object-Oriented Software Engineering*)
    - De 1996 até hoje:
      - **UML** (*Unified Modeling Language*) é hoje uma **linguagem** de **modelagem** *de facto* para **sistemas orientados** a **objetos** e **padronizada** pela **OMG** (*Object Management Group*).
  - A **UML** especifica a **sintaxe** e o **significado** dos **elementos gráficos** que **descrevem** um **sistema orientado** a **objetos** - **representação pictórica** de todos os **aspectos** do **funcionamento** de um **sistema** computacional: a visão estática (de estrutura), a visão de comportamento (de execução) e a visão de situações (de estado) de um sistema.

Análise Orientada a Objetos

- **Elementos do paradigma da orientação a objetos**
  - Um **objeto do mundo real** é **alguma** "**coisa**" **tangível** e/ou **visível** ou algo que **pode** ser **compreendida intelectualmente** ou ainda **algo** para o **qual pensamentos** ou **ações** são **direcionados** (BOOCH et al., 2007).
  - Para a **modelagem** de **sistemas computacionais orientados** a **objetos** é **imprescindível** o **emprego** da **antropomorfização**, ou seja, dar **características humanas** a **objetos inanimados**, **constituindo** o que se conhece por "**pensando como objeto**" (WEST, 2004).

Análise Orientada a Objetos

- **Elementos do paradigma da orientação a objetos**
  - **Conceito de objeto**
    - **Exemplo – Funcionários de uma empresa**
      - **Características**: nome, idade, sexo, sangue, endereço, salário, departamento …
      - **O que limita** a **seleção** dessas **características** é o **domínio do problema**.

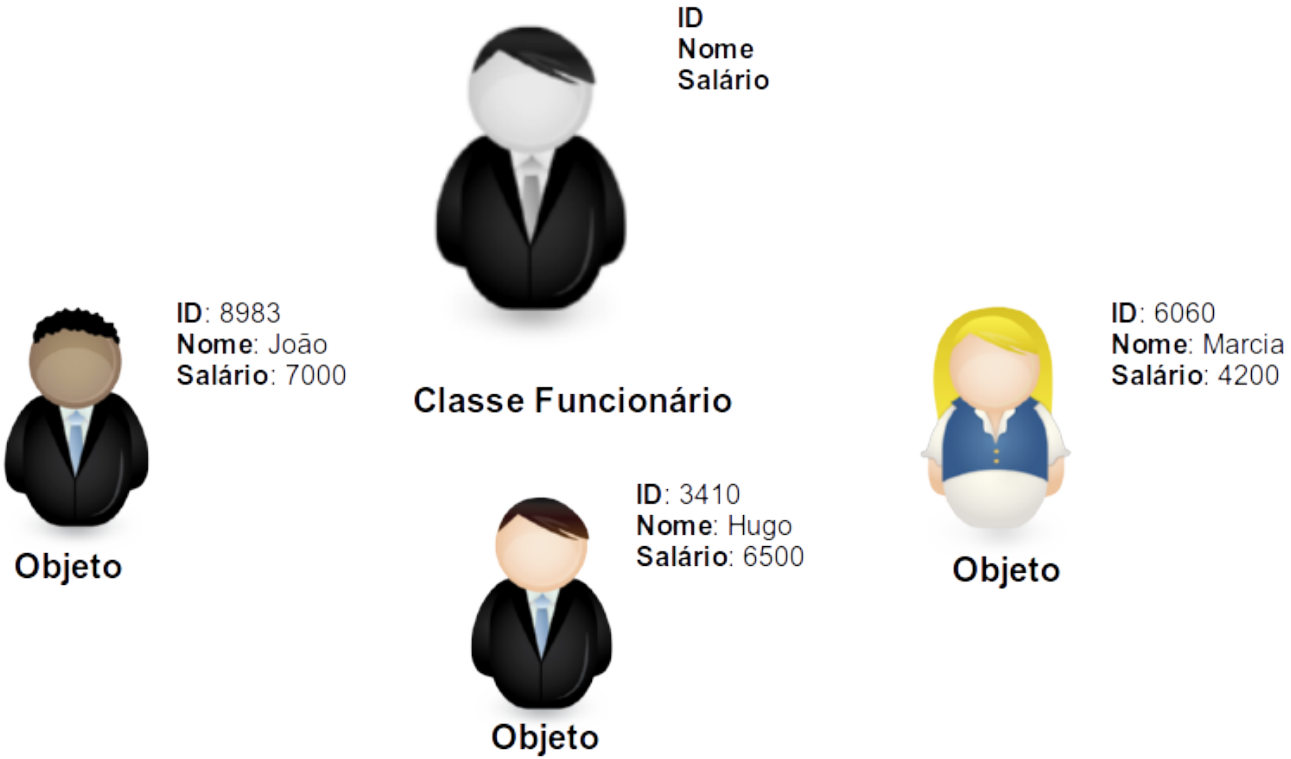


Análise Orientada a Objetos

- **Elementos do paradigma da orientação a objetos**
  - **Conceito de classe**
    - **Exemplo – Funcionários de uma empresa**
      - Embora os **objetos apresentados** possuam **valores particulares** de **identificação**, **nome** e **salários** eles possuem uma **estrutura** em **comum**;
      - Assim, **podemos agrupá-los** em uma **categoria**, por exemplo **Funcionários**.
      - A partir **desta categoria**, **pode-se identificar** os **indivíduos João**, **Marcia** e **Hugo** como seus **representantes** desta categoria.
      - A **categoria** é **conhecida** em **orientação a objetos** como **classe** e os **elementos** que a **representam** são **conhecidos** como **objetos** desta classe.

Análise Orientada a Objetos

- **Elementos do paradigma da orientação a objetos**
  - **Conceito de classe e objeto**

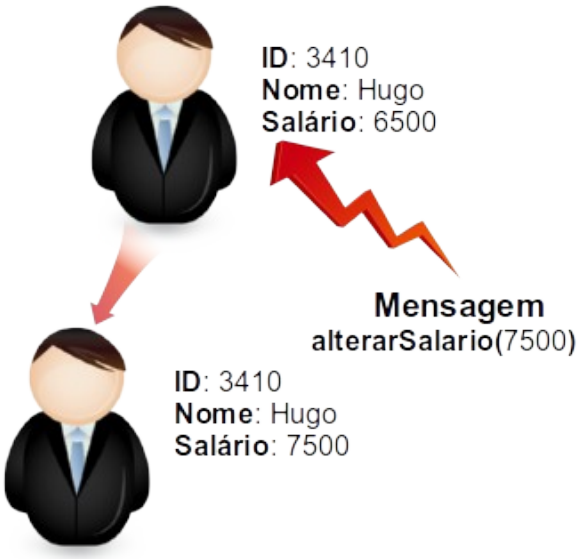


Análise Orientada a Objetos

- **Elementos do paradigma da orientação a objetos**
  - **Conceito de operação e mensagem**
    - Para que os **objetos interajam entre si** e **entre** o **ambiente**, é **necessário** que eles **troquem mensagens**, que são o **resultado** da **execução** de **operações acessíveis pelas interfaces** dos **objetos**;
    - No caso dos **funcionários**, pode-se definir o **seguinte conjunto** de **operações** (em uma sintaxe genérica, a título de exemplo):
      - obterID()
      - obterNome()
      - obterSalario()
      - alterarNome(novoNome)
      - alterarSalario(novoSalario)

Análise Orientada a Objetos

- **Elementos do paradigma da orientação a objetos**
  - **Conceito de operação e mensagem**
    - Em **orientação** a **objetos**, **enviar** uma **mensagem** a um **objeto significa executar uma operação presente** na **interface** do **objeto receptor**;
    - **Exemplo**:



Análise Orientada a Objetos

- **Elementos do paradigma da orientação a objetos**
  - **Conceito de operação e mensagem**
    - As **operações** de um **objeto podem**:
      - **Criar** e **destruir objetos** da **classe**;
      - **Alterar um** ou **mais atributos** (operações **modificadoras**);
      - **Consultar** o **conteúdo** de um **atributo** (operações **seletoras**).
    - O **conjunto** de **todos** os **valores assumidos** pelos **atributos** de um **objeto num instante qualquer** é **conhecido** como **estado** do **objeto**;
    - As **operações definidas** na **classe** é que são **responsáveis** pela **alteração ou não** do **estado** do objeto.

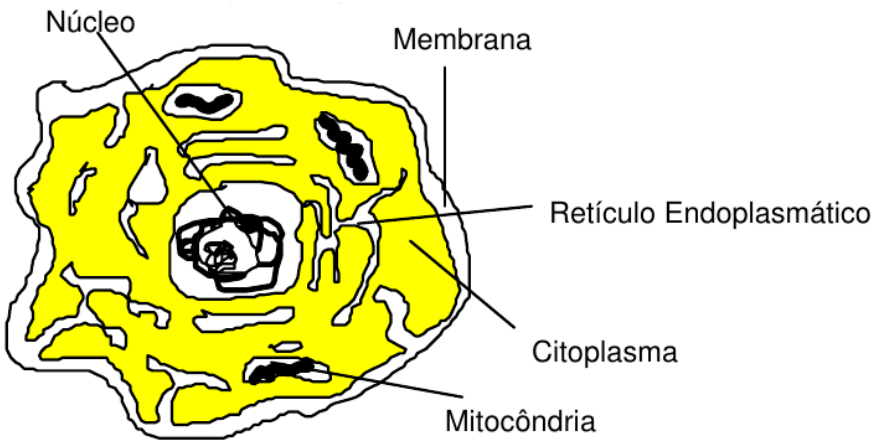
Análise Orientada a Objetos

FATEC
Faculdade de Tecnologia do Estado de São Paulo

- **Elementos do paradigma da orientação a objetos**
  - **Conceito de abstração**
    - **Abstração** é um **processo** que, a partir de um **objeto** do **mundo real**, devemos **incluí-lo** em nosso **problema**, **respeitando** o seu **domínio**;
    - Um **mesmo objeto** do mundo **real** pode ser **representado** (**abstraído**) no mundo **computacional** de **diversas maneiras**.;
    - Para a abstração interessa as **características essenciais** de um **objeto – abstrações-chave**. Essas **características distinguem** o **objeto** de **outros tipos** de **objeto**;
    - **Anteriormente** foi realizado um **rápido processo** de **abstração**: foram **selecionadas propriedades** e **operações** que se **julgou necessário** para representar um **funcionário** de uma **empresa** dentro do software.

Análise Orientada a Objetos

- **Elementos do paradigma da orientação a objetos**
  - **Conceito de encapsulação**
    - **Encapsulação** é o **processo** de se **esconder** os **detalhes** de um **objeto que não precisam** ser **expostos**;
    - Assim, as **propriedades não são diretamente acessadas**, a **não ser** que seja pelas **operações** do **próprio objeto**;
    - **Exemplo – células** que compõem um órgão:



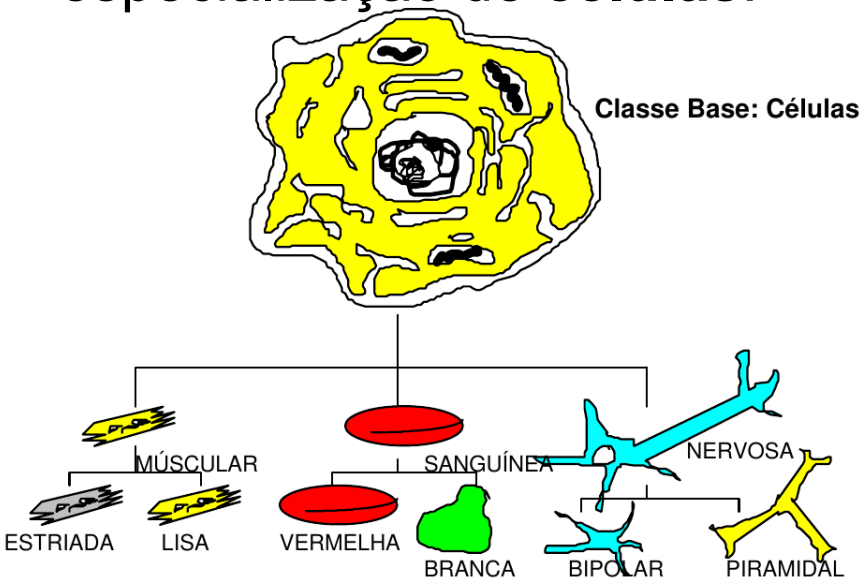
Célula Típica

Análise Orientada a Objetos

- **Elementos do paradigma da orientação a objetos**
  - **Conceito de modularidade**
    - É a **propriedade** de um **sistema** de ser **decomposto** em um **conjunto** de **módulos coesos** e **vagamente acoplados entre si**;
    - **Modularidade ideal** é aquela que, na **revisão**, **ampliação** ou **manutenção** de um **sistema**, leve à **alteração somente** dos **módulos diretamente afetados**, deixando intactos os outros módulos;
    - A modularidade **não** é uma **novidade introduzida** em **orientação objetos**: ela **existe** em **outros paradigmas**;
    - Em **orientação** a **objetos**, as **classes**, podem ser **consideradas** "**mini-módulos**" → um **sistema orientado** a **objetos bem projetado** quanto à **modularidade facilita** a sua **manutenção**, pois reduz o problema dos efeitos colaterais produzidos em consequência das mudanças.

Análise Orientada a Objetos

- **Elementos do paradigma da orientação a objetos**
  - **Conceitos de herança e polimorfismo**
    - **Herança** é um tipo de **relacionamento** que **existe** entre **classes**, onde uma **determinada classe** pode ser **derivada** a **partir** de **outra**, **herdando todas** as suas **características fundamentais** e **adicionando outras** que a **torne** mais **especializada**.
    - **Exemplo –** especialização de **células**:



Análise Orientada a Objetos

- **Elementos do paradigma da orientação a objetos**
  - **Conceitos de herança e polimorfismo**
    - **Polimorfismo** é a capacidade que objetos de classes distintas (daí "**muitas formas**") **respondam** à **mesma mensagem** mas executando ações distintas;
    - **Polimorfismo** em **orientação** a **objetos existe somente** quando há **herança**;
    - Por **exemplo**, a **classe Celula** pode **definir** uma **operação** para **oxigenação**. No entanto, **esta mesma operação**, vital para qualquer célula, **deve certamente funcionar** de **modo diferente** para **cada tipo** de **célula específica**;
    - **Princípio** da **substituição** – junto com polimorfismo é fundamental ao paradigma da orientação a objetos:
      - "*Um objeto de uma classe derivada de uma classe base pode ser utilizado no lugar de um objeto desta classe base*" (BOOCH et al., 2007).

Agenda

- Análise de Sistemas
- Análise Orientada a Objetos
- Métodos para identificar classes
- Identificação de classes com fichas CRC

Métodos para identificar classes

- **O problema da identificação**
  - A **identificação** das **classes** de **Análise** é realizada a **partir** de uma **análise** dos **requisitos** do sistema – **textos** dos **casos** de **uso**;
  - Os **casos de uso** são **estudados** e **encenados** para uma **melhor compreensão**. **Durante** este **processo** de **encenação**, tentam-se **identificar** os **objetos** que **participam** do **cenário**, suas **responsabilidades** e os **modos** como **esses objetos colaboram** com outros **objetos**, em termos das **operações** que **cada um invoca** no **outro**;
  - Isso **cria** uma clara **separação de interesses entre todas** as **abstrações**. **Conforme** o **processo** de desenvolvimento **continua**, estes **cenários iniciais** são **expandidos** para **considerar** as **condições excepcionais**, bem como **comportamentos secundários** do sistema.

Métodos para identificar classes

- **Abordagens**
  - **Abordagem clássica** (BOOCH et al., 2007)
    - **Utiliza princípios** da **categorização clássica**, teoria que tem suas raízes com **Platão**. **Consiste** em **separar objetos** em **categorias** tais como:

"Coisas"	Objetos ou grupo de objetos tangíveis como, por exemplo, automóveis, geladeira, caixa.
Papéis	Por exemplo, mãe, professor, político.
Eventos	Por exemplo, aterrissagem, interrupção, requisição.
Interações	Por exemplo, empréstimo, encontro, intersecção.
Pessoas	Humanos que executam alguma função.
Lugares	Áreas utilizadas por humanos ou coisas.

Métodos para identificar classes

- **Abordagens**
  - **Abordagem clássica** (cont.)

Organizações	Coleções formalmente organizadas de pessoas, recursos, instalações e capacidades que têm uma missão definida, cuja existência é independente dos indivíduos.
Conceitos	Princípios ou ideias não tangíveis. Usado para organizar ou manter o controle de atividades de negócios e/ou comunicações.
Eventos	Coisas que acontecem em uma determinada data e hora, ou como etapas em uma sequência ordenada.
Estrutura	Relacionamentos do tipo “parte-de” e “é-um”.
Outros sistemas	Sistemas externos com os quais a aplicação interage.
Dispositivos	Dispositivos com os quais a aplicação interage.
Eventos relembrados	Eventos históricos que devem ser registrados.
Papéis desempenhados	Diferentes papéis que os usuários desempenham na interação com a aplicação.
Locais	Localizações físicas, escritórios e locais importantes para a aplicação.
Unidades organizacionais	Grupos aos quais pertencem os usuários.

Métodos para identificar classes

- **Abordagens**
  - **Comportamento dinâmico**
    - Essas abordagens **assemelham**-se com **agrupamento conceitual**;
    - **Formam-se classes** com **base** em **grupos** de **objetos** que **exibem comportamento semelhante**;
    - **Técnicas tais** como **Fichas CRC enfatizam** as **responsabilidades – conhecimento** que um objeto mantém e as ações que ele pode realizar.
    - Nesta abordagem, pode-se entender **objetos como sendo provedores** de **serviços contribuem** para **definir funcionalidades** de um **sistema**.

Métodos para identificar classes

- **Classificação das classes**
  - Pode-se ainda **classificar** as **classes** de **análise** em **três tipos** (JACOBSON, 1992):
    - **Classes de entidade**: seus objetos **modelam** a **informação** que o **sistema** deverá **tratar**. São **dependentes** do **domínio** do **problema** e representam seus **conceitos-chave**;
    - **Classes de fronteira**: os **objetos** de **fronteira** recebem toda **funcionalidade** dos **casos** de uso **diretamente dependente** do **ambiente** do **sistema**. Classes de fronteira são facilmente identificadas nos casos de uso por que são **diretamente operadas** pelos **atores** do **sistema**.
    - **Classes de controle**: seus **objetos** agem como "**cola**" entre **objetos** dos **outros tipos**, **evitando** que se **particularize** o **controle** do **comportamento** do sistema, **embutindo-o** tanto em **objetos** do tipo **entidade** quanto de **interface**.

Agenda

- Análise de Sistemas
- Análise Orientada a Objetos
- Métodos para identificar classes
- Identificação de classes com fichas CRC

Identificação de classes com fichas CRC

FATEC
Faculdade de Tecnologia do Estado de São Paulo

- **Conceitos**
  - **Métodos humanos** de **organização para estudar sistemas complexos** (WEST, 2004):
    - **Diferenciação**: decidir como uma “coisa” é diferente de outra.
    - **Classificação**: descobrir similaridades em um grupo de coisas e rotulá-las. Classificação requer diferenciação.
    - **Composição**: “coisas” complexas podem consistir de “coisas” mais simples.
  - **Para auxiliar** nessa **tarefa** de **identificação** de **classes** e **objetos**, pode-se **utilizar** o **método** das **fichas CRC**, **acrônimo** de “***Candidates, Responsibilities, Collaborators***” (Candidatos, Responsabilidades, Colaboradores) (WIRFS-BROCK; MCKEAN, 2003).

Identificação de classes com fichas CRC

FATEC
Faculdade de Tecnologia do Estado de São Paulo

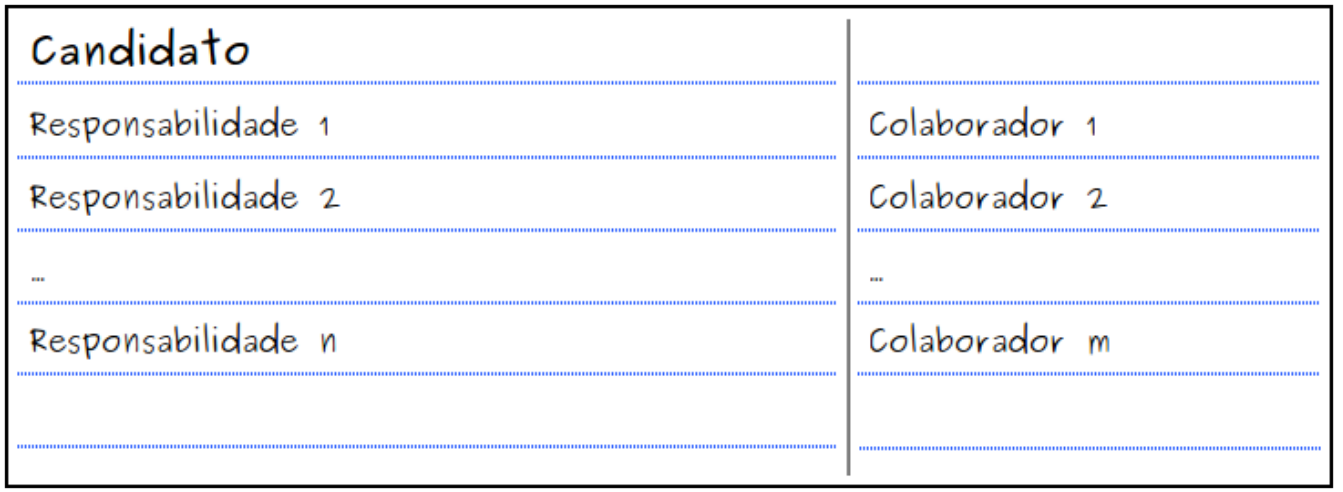
- **Conceitos**
  - Esta **técnica** foi **criada** na **década** de **1990** e naquela época denominava-se “Class, Responsibility, Collaborator” (Classe, Responsabilidade, Colaborador) (WIRFS-BROCK; WILKERSON; WIENER, 1990);
  - **É uma técnica** de **projeto dirigido** por **responsabilidade**;
  - **Responsabilidade** pode ser **entendida** como uma **caracterização abstrata** dos **serviços** um **objeto** é **capaz** de **executar** – que **responsabilidades** um **objeto** possui **frente** ao **sistema** como um **todo** (ELIËNS, 1995).
  - Uma **responsabilidade** pode ser **entendida** como um **contrato** que o **objeto honra** em **relação** a seus **clientes**.

Identificação de classes com fichas CRC

- **Conceitos**
  - **Elementos** (WIRFS-BROCK; MCKEAN, 2003)
    - **Aplicação**: é um conjunto de objetos interagentes.
    - **Objeto**: é a implementação de um ou mais papéis.
    - **Papel**: é um conjunto de responsabilidades relacionadas.
    - **Responsabilidade**: é uma obrigação de executar uma tarefa ou conhecer uma informação.
    - **Colaboração**: é uma interação de objetos ou papéis ou ambos.
    - **Contrato**: é um acordo delineando os termos de uma colaboração. Na prática, quem define os contratos é o analista/projetista do sistema.

Identificação de classes com fichas CRC

- **Aplicação**
  - O **método CRC consiste** no **preenchimento interativo** de **fichas** com **formato similar** a este:



    - **Candidato**: representa um **conceito** do **domínio** do sistema em estudo, **normalmente** uma **classe** ou **objeto** do **sistema**.
    - **Responsabilidade**: é **alguma coisa** que o **candidato conhece** ou **sabe executar**.
    - **Colaborador**: indica **nomes** de outros **candidatos** que **complementam** as **responsabilidades** do **candidato**.

Identificação de classes com fichas CRC

FATEC
Faculdade de Tecnologia do Estado de São Paulo

- **Aplicação**
  - Para **criar modelos** de **fichas CRC**, executam-se de modo iterativo os passos a seguir (AMBLER, 2004):
    - (1) **Descobrir classes**: utilizar alguma estratégia para identificar classes nos requisitos.
    - (2) **Determinar responsabilidades**: determinar o que a as classes devem fazer/saber.
    - (3) **Definir colaboradores**: determinar colaboradores para suprir conhecimentos e funcionalidades que a classe não possui.
    - (4) **Mover as fichas pelo grupo**: o método CRC, além de iterativo, torna-se mais produtivo quando executado em grupo.

Identificação de classes com fichas CRC

- **Exemplos de fichas CRC** – *Sistema de Caixa Eletrônico*

Classe: **`CaixaEletronico`**	
Responsabilidade:	**Colaborador**:
Inicializar o sistema.	**`ConsoleCliente`**, **`LeitorCartao`**, **`DispensadorDinheiro`**, **`Sessao`**, **`Banco`**.
Encerrar o sistema.	**`ConsoleCliente`**, **`LeitorCartao`**, **`DispensadorDinheiro`**, **`Sessao`**, **`Banco`**.
Iniciar uma nova sessão com o cliente.	**`Sessao`**, **`Banco`**.
Encerrar uma sessão com o cliente.	**`Sessao`**, **`Banco`**.

Identificação de classes com fichas CRC

- **Exemplos de fichas CRC** – *Sistema de Caixa Eletrônico*

Classe: `Transacao`	
Responsabilidade:	**Colaborador**:
Executar uma transação financeira.	`Sessao`, `Cartao`, `Banco`, `LogTransacao`.

Classe: `TransacaoCredito`	**Superclasse**: `Transacao`
Responsabilidade:	**Colaborador**:
Executar uma transação financeira de crédito.	`Sessao`, `Cartao`, `Banco`, `LogTransacao`.

Referências bibliográficas

- AMBLER, S. W., ,. **The object primer: agile modeling-driven development with UML 2.0**. Cambridge, UK; New York: Cambridge University Press, 2004.
- ARLOW, J.; NEUSTADT, I. **UML 2 and the unified process: practical object-oriented analysis and design**. Upper Saddle River, NJ: Addison-Wesley, 2005.
- BOOCH, G. et al. **Object-oriented analysis and design with applications**. [s.l.] Addison-Wesley, 2007.
- BOOCH, G.; JACOBSON, I.; RUMBAUGH, J. **The Unified Modeling Language User Guide**. Upper Saddle River, NJ [u.a.: Addison-Wesley, 2005.
- ELIËNS, A. **Principles of object-oriented software development**. Wokingham, England; Reading, Mass.: Addison-Wesley Pub. Co., 1995.
- JACOBSON, I. **Object-oriented software engineering: a use case driven approach**. [New York]; Wokingham, Eng.; Reading, Mass.: ACM Press?; Addison-Wesley Pub., 1992.
- KROLL, P.; KRUCHTEN, P. **The Rational Unified Process made easy: a practitioner's guide to the RUP**. Boston: Addison-Wesley, 2003.
- KRUCHTEN, P. **The Rational Unified Process: an introduction**. Boston: Addison-Wesley, 2004.
- WEST, D. **Object Thinking**. Redmond, WA: Microsoft, 2004.
- WIRFS-BROCK, R.; MCKEAN, A. **Object design: roles, responsibilities, and collaborations**. Boston [Mass.]; London: Addison-Wesley, 2003.
- WIRFS-BROCK, R.; WILKERSON, B.; WIENER, L. **Designing object-oriented software**. Englewood Cliffs, N.J.: Prentice Hall, 1990.