



Análise e Projeto de Software

Universidade Evangélica de Goiás

Curso de Engenharia de Software



Análise e projetos de Software: Introdução a Análise Orientada a Objeto

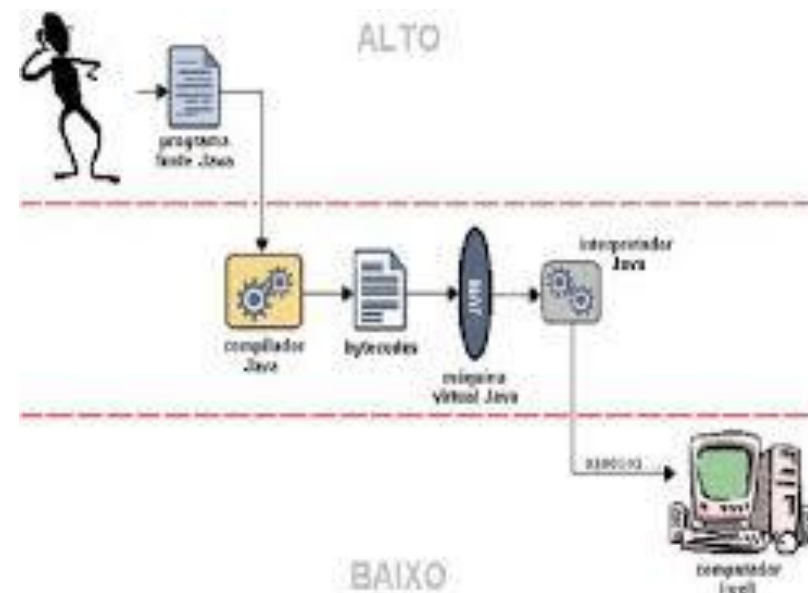
Introdução

Pois bem, outra metodologia amplamente utilizada é a Análise Orientada a Objetos. De acordo com Pressman (2006), esta metodologia aborda o domínio do problema como um conjunto de objetos que têm atributos e comportamentos específicos. Os objetos são manipulados com uma coleção de funções, denominadas métodos, operações ou serviços e comunicam-se uns com os outros através de mensagens. Assim, quando um objeto é definido, seus atributos, operações e mensagens são levadas em consideração.



Abordagem

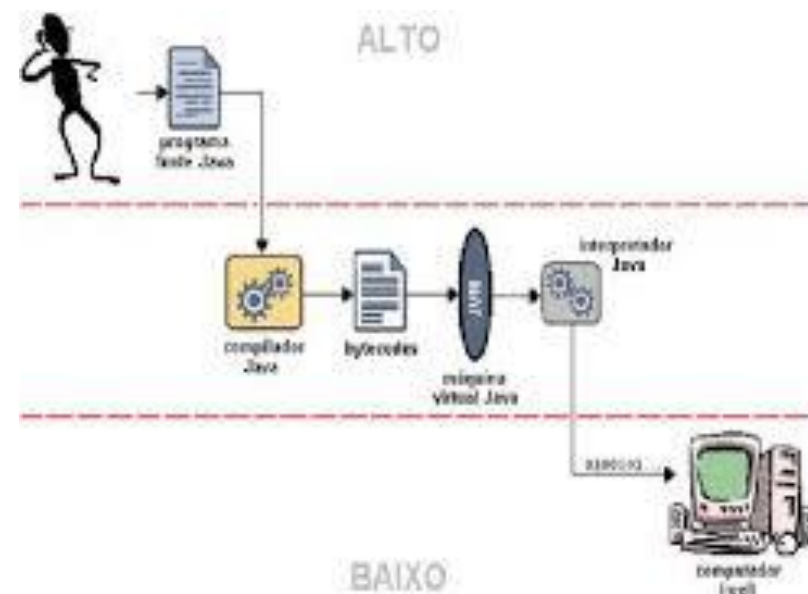
A importância desta abordagem está no fato de que os objetos encapsulam (guardam em si) não só os dados, mas o processamento aplicado a eles. Com isto, bibliotecas de classes, contendo objetos reusáveis, são desenvolvidas e, conforme já mencionado, reuso é um conceito de extrema importância, principalmente, em orientação a objetos, pois leva à criação mais rápida de software e a programas com maior qualidade.



Abordagem

Além disto, o software orientado a objetos possui estrutura desacoplada, evitando, assim, efeitos colaterais quando uma parte do software é alterada e essas alterações não afetarão outras partes, permitindo, com isso, uma maior adaptabilidade e ampliação facilitada.

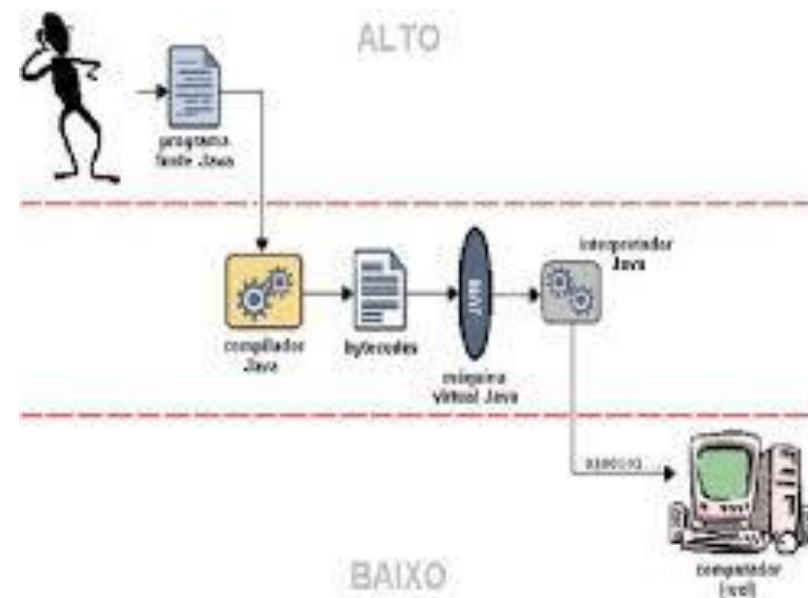
Sistemas orientados a objetos evoluem com o tempo. Desta forma, a engenharia de software orientada a objetos é mais bem descrita com uma abordagem que incentiva a criação de componentes (reuso) vinculada ao modelo de processo evolutivo. Ao longo da espiral evolutiva, que começa com as entrevistas com os usuários, as classes básicas são levantadas



Abordagem

Da mesma forma que na abordagem clássica não é possível o levantamento de todos os requisitos de software nas primeiras entrevistas, no modelo de análise e projeto orientado a objetos nem todas as classes necessárias são descobertas inicialmente, tornando-se necessárias classes adicionais ao longo do processo, demonstrando, assim, a natureza evolutiva desta abordagem.

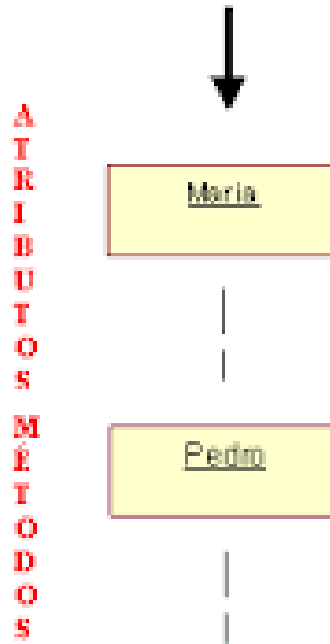
Para que a abordagem orientada a objetos seja compreendida se faz necessário que alguns conceitos sejam assimilados.



Classe



Objetos



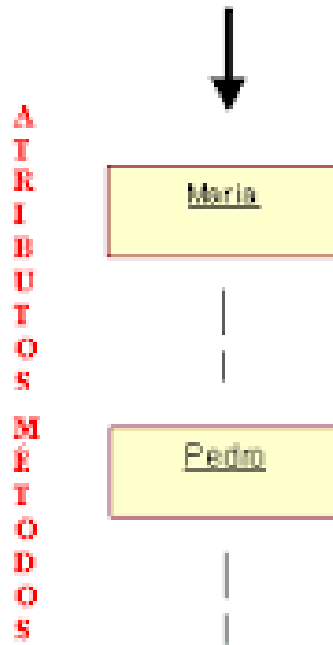
Classes e Objetos

- Uma classe é um conceito da orientação a objetos que encapsula as abstrações de dados e procedimentais necessárias para descrever o conteúdo e o comportamento de alguma entidade do mundo real.
- As abstrações de dados (atributos) que descrevem a classe são envolvidas por uma “parede” de abstrações procedimentais (operações ou métodos), que são capazes de manipular os dados de alguma forma.

Classe



Objetos



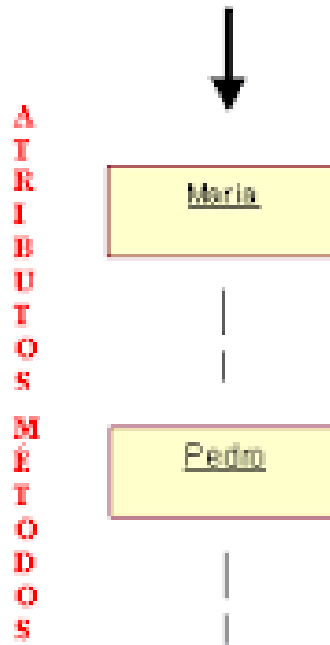
Classes e Objetos

Esta característica leva à ocultação de informação e reduz possíveis efeitos que ocorreriam em caso de modificações. Os métodos, por manipular um conjunto pequeno de atributos, diz-se que eles são coesivos e, sendo o acesso aos atributos feito somente através dos métodos, a classe tende a ser desacoplada de outros elementos do sistema.

Classe



Objetos



Classes e Objetos

Em alguns casos, classes mais genéricas são construídas e, partir destas, classes mais especializadas são derivadas ou herdadas. Isto leva a uma hierarquia de classes, na qual classes superiores são denominadas de superclasses e classes derivadas subclasses. As subclasses herdam todos os atributos e métodos das superclasses e declaram seus próprios atributos e métodos. A Figura 18 apresenta uma hierarquia de classes para mobiliário.

Classes e Objetos

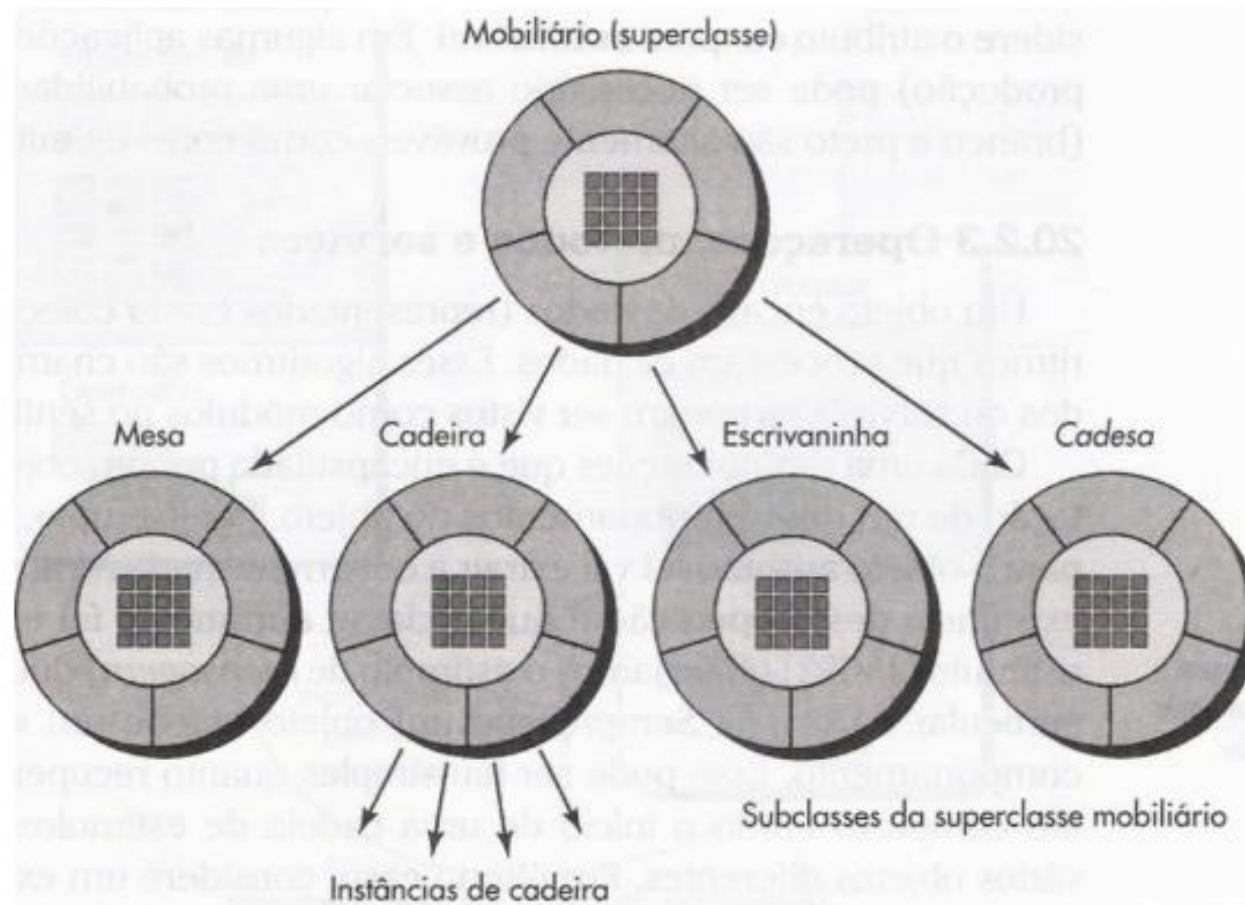
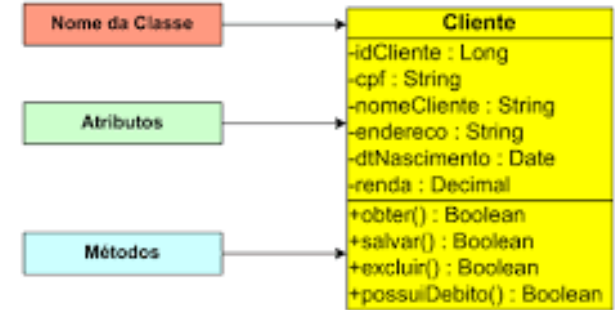


Figura 18 - Hierarquia de classes

Fonte: Pressman (2006)

Atributos

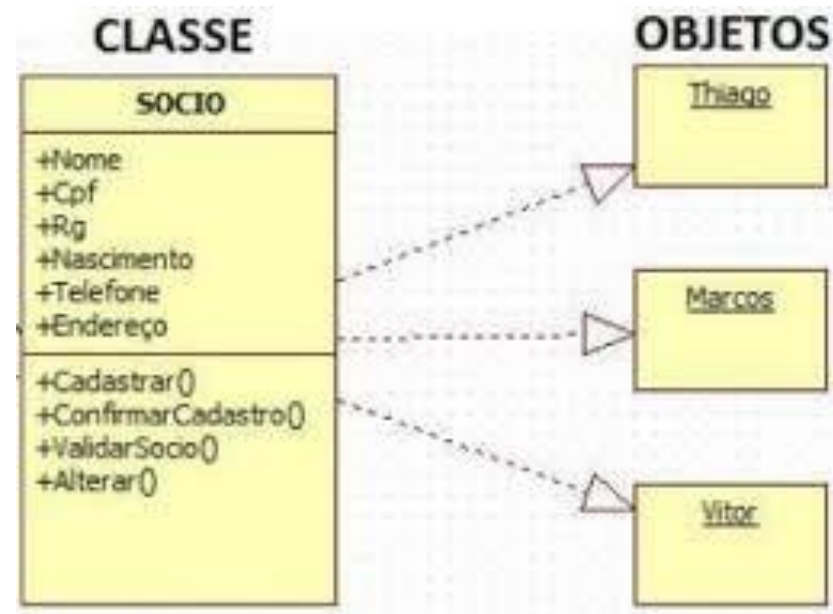
Atributos são descrições para classes de objetos. Uma pessoa, por exemplo, descrita pela classe “pessoa”, tem características, tais como o nome, data de nascimento, altura, cor dos olhos e idade, entre outros. A classe “venda”, que descreve uma nota fiscal de venda, por exemplo, é descrita por data da venda, vendedor, valor total da venda e um conjunto de itens que descreve os produtos vendidos.



Classe: composta de atributos e métodos

Operações, métodos e mensagem

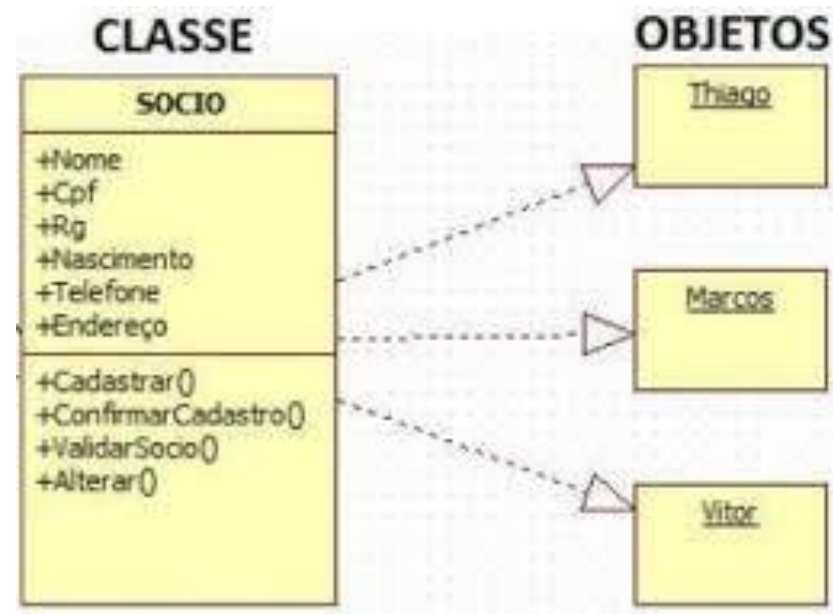
As operações (ou métodos) descrevem os algoritmos que processam os dados de um objeto. Cada operação descreve um comportamento encapsulado por um objeto. De acordo com os exemplos de classes e atributos citados anteriormente, um exemplo de método é obter total da venda, que buscará dentro do objeto o valor que representa o total da venda, possível a partir do atributo valor total da venda.



Operações, métodos e mensagem

Isso faz com que o objeto da classe “venda” receba um estímulo e este estímulo é denominado mensagem.

Uma mensagem é a forma pela qual os objetos interagem entre si. Em outras palavras, uma mensagem estimula que algum comportamento aconteça no objeto que a recebe



Encapsulamento, herança e polimorfismo



Sistemas orientados a objetos se beneficiam de características importantes por encapsularem dados e operações em uma única entidade (classe):

- A ocultação de informação permite que detalhes dos dados e da implementação das operações permaneçam escondidas do mundo externo;
- Tanto dados quanto operações que os processam são encerrados (declarados) dentro de uma entidade;
- As mensagens enviadas entre objetos não precisam conhecer detalhes das estruturas interna dos dados.

Encapsulamento, herança e polimorfismo

- A herança, conforme já visto, possibilita que novas classes possam fazer uso de classes já existentes, por meio do conceito de herança. Toda classe que herda de uma classe pré-existente possui, além de seus atributos e métodos, os atributos e métodos da classe a partir da qual se herda.



Encapsulamento, herança e polimorfismo

- A palavra polimorfismo tem suas origens no Grego e significa muitas formas (poli = muitas, morphos = formas). De forma simplista, polimorfismo é a habilidade que um tipo A pode se comportar como um tipo B. Como exemplo genérico tem-se uma classe denominada Animal, que contém o método falar. Este método apresentará um texto que representa o som que o animal faz. Porém, em Animal não é possível implementar tal método, visto que um animal por si só não emite sons. Entretanto, a classe Cachorro e Gato herdam da classe Animal, conseqüentemente, implementarão (sobrescreverão) o método falar, pois estes animais sabem como “falar”. Assim, quando uma mensagem for enviada a um objeto cachorro ordenando que ele “fale”, o método falar apresentará o texto “au-au” e quando a mesma mensagem for enviada a um objeto gato ordenando que ele “fale”, o método falar apresentará o texto “miau”.



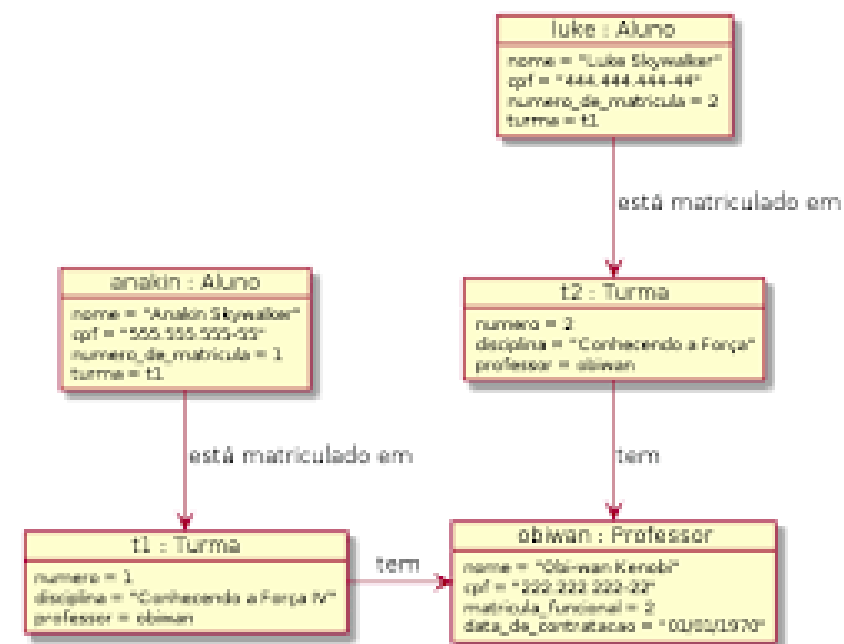
Encapsulamento, herança e polimorfismo

- Outro exemplo a considerar é o de uma aplicação que desenha elementos gráficos (círculos, quadrados e triângulos). Em uma aplicação convencional cada elemento gráfico necessitaria de um módulo de desenho, que teria uma lógica para saber qual o elemento a ser desenhado. O problema dessa abordagem reside no fato de que o acréscimo de novos elementos implicaria em novos módulos de desenho. Em orientação a objetos os elementos gráficos são descritos como classes que herdam, por exemplo, de uma superclasse “Gráfico”, contendo o método desenhar. Empregando o conceito de sobrecarga, cada subclasse define seu próprio algoritmo no método desenhar que herdou da classe “Gráfico”. Assim, o método desenhar pode ser invocado em qualquer um dos objetos que herde de “Gráfico”. O objeto que receber a mensagem invocará seu próprio método desenhar fazendo com que o elemento gráfico representado pela classe do objeto seja desenhado.

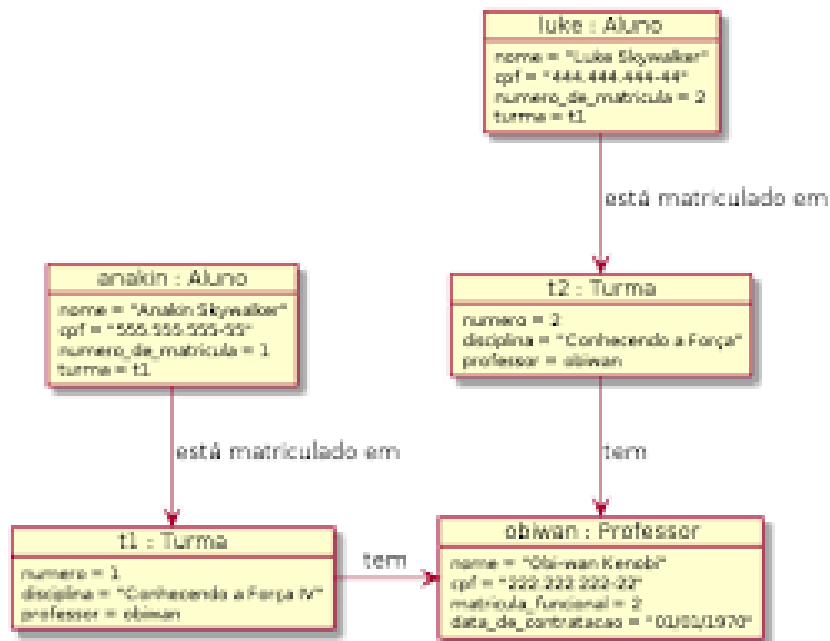


Análise orientada a objetos

- A análise orientada a objetos tem a finalidade de definir as classes que são importantes ao problema a ser resolvido, descobrindo seus atributos e operações, as relações existentes entre elas e o comportamento exibido por elas. As tarefas necessárias para alcançar tais objetivos são:
- 1. levantar os requisitos básicos junto aos usuários;
- 2. identificar as classes (atributos e operações);
- 3. especificar uma hierarquia de classes;
- 4. representar as relações entre objetos;
- 5. modelar o comportamento dos objetos;
- 6. aplicar as tarefas anteriores repetidamente (iterativamente) até que o modelo seja obtido.



Análise orientada a objetos

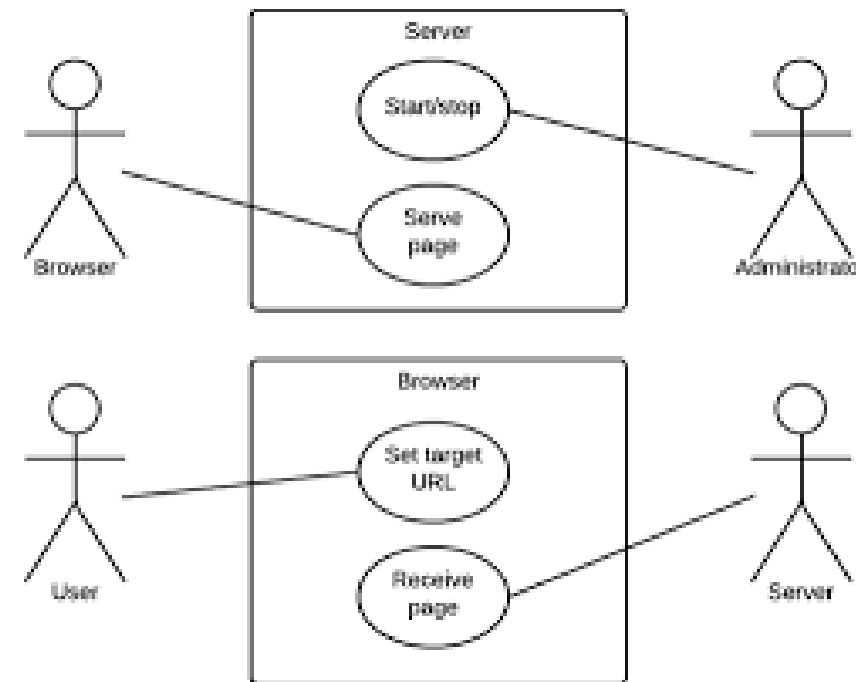


- No final dos anos 80 e durante os anos 90 vários métodos de análise orientada a objetos foram sugeridos. Estes métodos eram compostos por um processo de análise, um conjunto de diagramas e uma notação utilizada para criar o modelo de análise. Entre todos, o que mais se destacou, na verdade, foi uma combinação de 3 (três), que propunha a combinação dos métodos idealizados por Grady Booch (Método Booch), James Rumbaugh (Técnica de Modelagem de Objetos (OMT – *Object Modeling Technique*) e Ivar Jacobson (Engenharia de Software Orientada a Objetos (OOSE – *Object-Oriented Software Engineering*). Assim, surgiu a UML (*Unified Modeling Language* – Linguagem de Modelagem Unificada), cujo propósito foi o de agrupar os pontos fortes dos 3 (três) métodos.

UML

Na UML a modelagem do software é expressa mediante uma notação de modelagem, que é regida pelas seguintes regras:

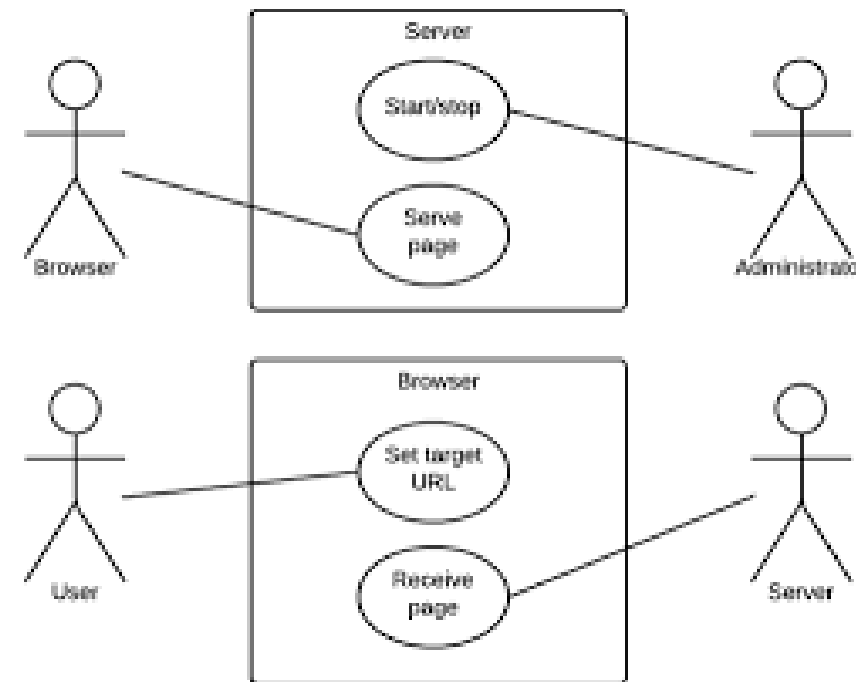
- **Sintáticas:** define quais símbolos devem existir e como são combinados para formas sentenças;
- **Semânticas:** diz respeito ao significado de cada símbolo e como ele é interpretado por si só e no contexto de outros símbolos;
- **Pragmáticas:** corresponde às regras para a criação de sentenças através da definição das intenções dos símbolos.



UML

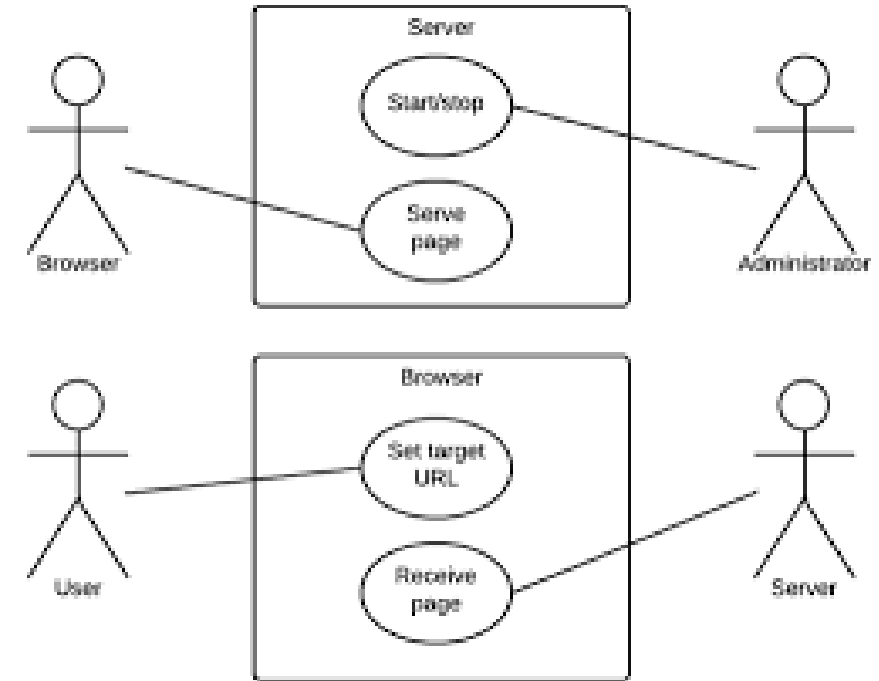
Um sistema é representado em UML por 5 (cinco) diferentes visões, cada uma definida por um conjunto de diagramas. A seguir serão apresentadas 2 (duas) das 5 (cinco) visões com as quais a análise se preocupa. As visões restantes são descritas na Unidade III – Projeto

- Na **visão do modelo do usuário** o sistema é representado sob o ponto de vista do usuário, denominado ator, e utiliza a abordagem de casos de uso, que descrevem os cenários de uso do sistema a partir da perspectiva do usuário. Em outras palavras, os casos de uso descrevem a interação dos atores com o sistema e como este é usado.
- A **visão do modelo estrutural** se preocupa com a modelagem da estrutura estática do sistema (classes, objetos e relacionamentos) visando os dados e funcionalidades do sistema.



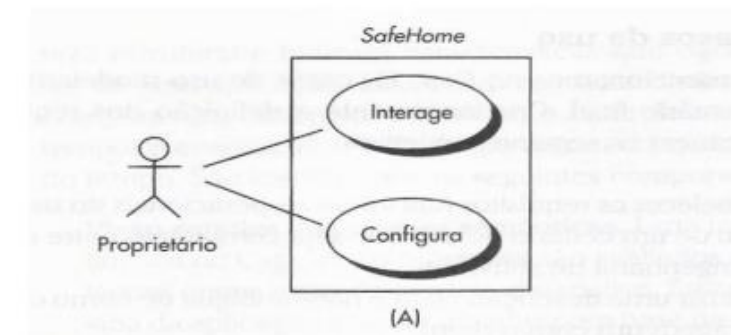
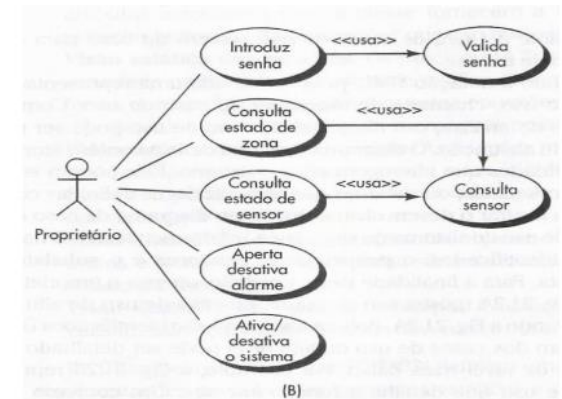
UML

- Como dito, o processo de análise orientada a objetos se preocupa em compreender como o sistema será usado. Uma vez que o cenário de uso tenha sido definido, inicia-se a modelagem do software.
- Os casos de uso modelam o sistema do ponto de vista do usuário e servem como base para o elemento inicial do modelo de análise. Com o auxílio da UML uma representação gráfica do caso de uso é criada. Conhecida como Diagrama de Caso de Uso, ele apresenta atores e casos de uso.



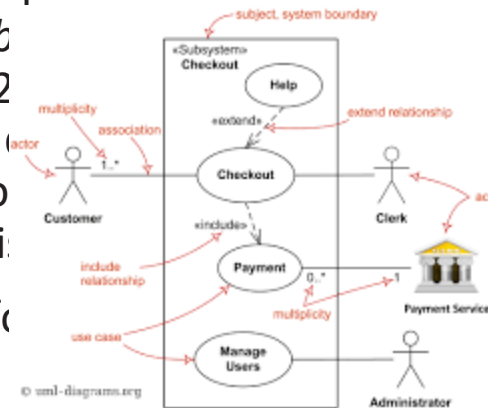
Caso de Uso

- Como exemplo, tem-se um sistema hipotético de segurança domiciliar. Entre outros, identificou-se o ator proprietário e 2 (dois) casos de uso, representados pela figura oval, apresentados na Figura A. Cada caso de uso pode ser detalhado a níveis mais baixos, semelhante ao DFD, visto anteriormente. A Figura B mostra o caso de uso para a função interage, apresentada na Figura A



Caso de Uso

- Uma vez levantados os cenários de uso, passa-se a identificar as possíveis classes e quais são suas responsabilidades e colaborações. Este processo recebe o nome de modelagem classe-responsabilidade-colaboração (*class-responsibility-collab* com exemplo apresentado na Figura 2 representação organizada de classes, do sistema. Todos os substantivos são responsabilidades, que é qualquer coisa que outras classes, que fornecem a ela informações para realizar uma responsabilidade.



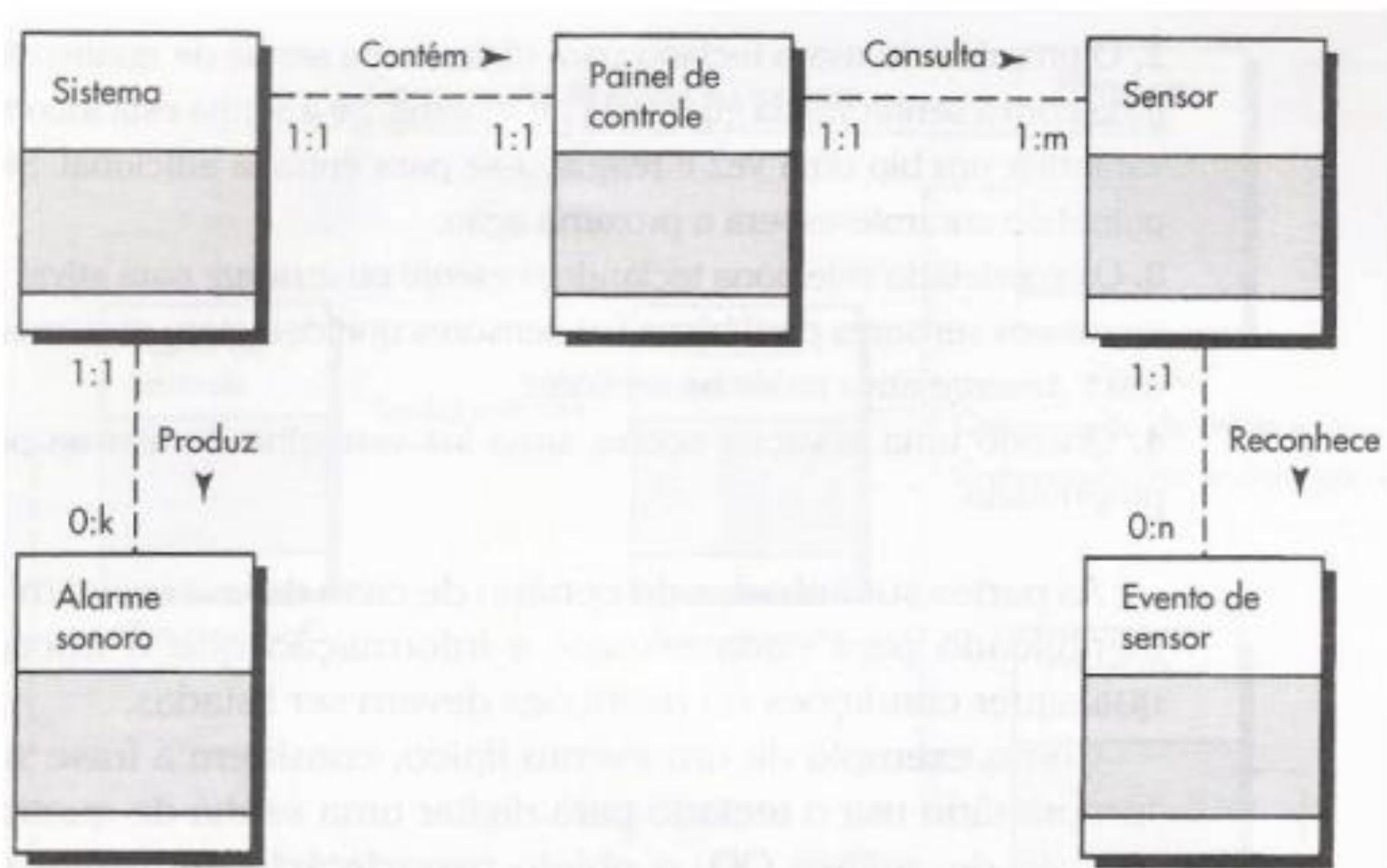
Modelo CRC

- Tendo sido estabelecidas as responsabilidades, a tarefa de modelagem objeto-relacionamento define as classes que colaboram entre si para satisfazer cada responsabilidade. Isto estabelece a associação (ou conexão) entre classes. Pressman (2006) sugere examinar os verbos na declaração dos casos de uso do sistema para que se possam encontrar os possíveis relacionamentos. A UML vale-se da notação para modelagem objeto-relacionamento a partir de uma simbologia adaptada da modelagem entidade-relacionamento. As associações entre objetos são realizada usando-se relacionamentos rotulados juntamente com a cardinalidade da associação.

Modelo CRC

- Composto de um conjunto de cartões com o seguinte formato:

Classe:	
Responsabilidade	Colaboração



Modelo CRC

Tanto a modelagem CRC quanto a modelagem objeto-relacionamento correspondem a elementos estáticos no modelo de análise orientada a objetos. Para demonstrar o comportamento dinâmico do sistema é necessário representar o comportamento do sistema como uma função de eventos e tempo, ambos específicos. Neste sentido, a modelagem objeto-comportamento descreve como o sistema orientado a objetos vai responder a eventos e estímulos externos e é criado de acordo com os seguintes passos:

1. A interação interna do sistema é avaliada utilizando-se todos os casos de uso;
2. Identificar e entender os eventos, como eles direcionam as interações e como se relacionam a objetos específicos;
3. Para cada caso de uso deve-se criar uma marcação de eventos;
4. Construir um Diagrama de Transição de Estados para todos os casos de uso;
5. Verificar a precisão e consistência do modelo objeto-comportamento.

Modelo CRC

- Composto de um conjunto de cartões com o seguinte formato:

Classe:	
Responsabilidade	Colaboração

