



Metodologias Ativas de Aprendizagem (Sala de Aula Invertida)





Metodologias Ativas de Aprendizagem

(Sala de Aula Invertida)



Modelagem e Desenvolvimento Orientado a Objeto



Modelagem e Desenvolvimento Orientado a Objeto

Plano de aula



ANÁLISE E PROJETO DE SISTEMAS ORIENTADOS A OBJETOS

- Histórico das Metodologias de Desenvolvimento de Sistemas
- Conceitos de Orientação a Objetos
- Abstração: Conceitos e Exemplos
- Classe e Objetos: Conceitos e Exemplos
- Associação: Conceitos e Exemplos
- Atributos: Conceitos e Exemplos
- Operações: Conceitos e Exemplos
- Diagrama de Casos de Uso: Conceitos, notação e aplicação
- Diagrama de Classes: Conceitos, notação e aplicação
- Diagrama de Objeto: Conceitos, notação e aplicação
- Diagrama de Sequência: Conceitos, notação e aplicação
- Diagrama de Estrutura Composta: Conceitos, notação e aplicação
- Diagrama de Comunicação: Conceitos, notação e aplicação
- Diagrama de Máquina de Estados: Conceitos e notação
- Diagrama de Atividades: Conceitos, notação e aplicação
- Diagrama de Interação Geral: Conceitos, notação e aplicação
- Diagrama de Componentes: Conceitos, notação e aplicação
- Diagrama de Pacotes: Conceitos, notação e aplicação
- Diagrama de Implantação: Conceitos, notação e aplicação
- Diagrama Temporização: Conceitos, notação e aplicação



Modelagem e Desenvolvimento Orientado a Objeto

Plano de aula

PROGRAMAÇÃO ORIENTADA A OBJETOS

- Introdução ao Java
- Introdução à Programação Orientada a Objetos com Java
- Tipos de dados em Java
- Estruturas de Controle de Fluxo e Laço em Java
- Arrays e Coleções de Dados
- Gerando e Interceptando Exceções
- Herança e Polimorfismo
- Abstração de dados
- Encapsulamento
- Organização em Pacotes
- Conexão com Bases de Dados
- Conceitos Básicos de GUI em Java
- Lidando com Eventos de GUI em Java
- Usando Gerenciadores de Layout
- Construção de GUI em Java
- Projeto de GUI conectada a Bases de Dados
- Projeto de um Applet Java



Modelagem e Desenvolvimento Orientado a Objeto

ANÁLISE E PROJETO DE SISTEMAS ORIENTADOS A OBJETOS



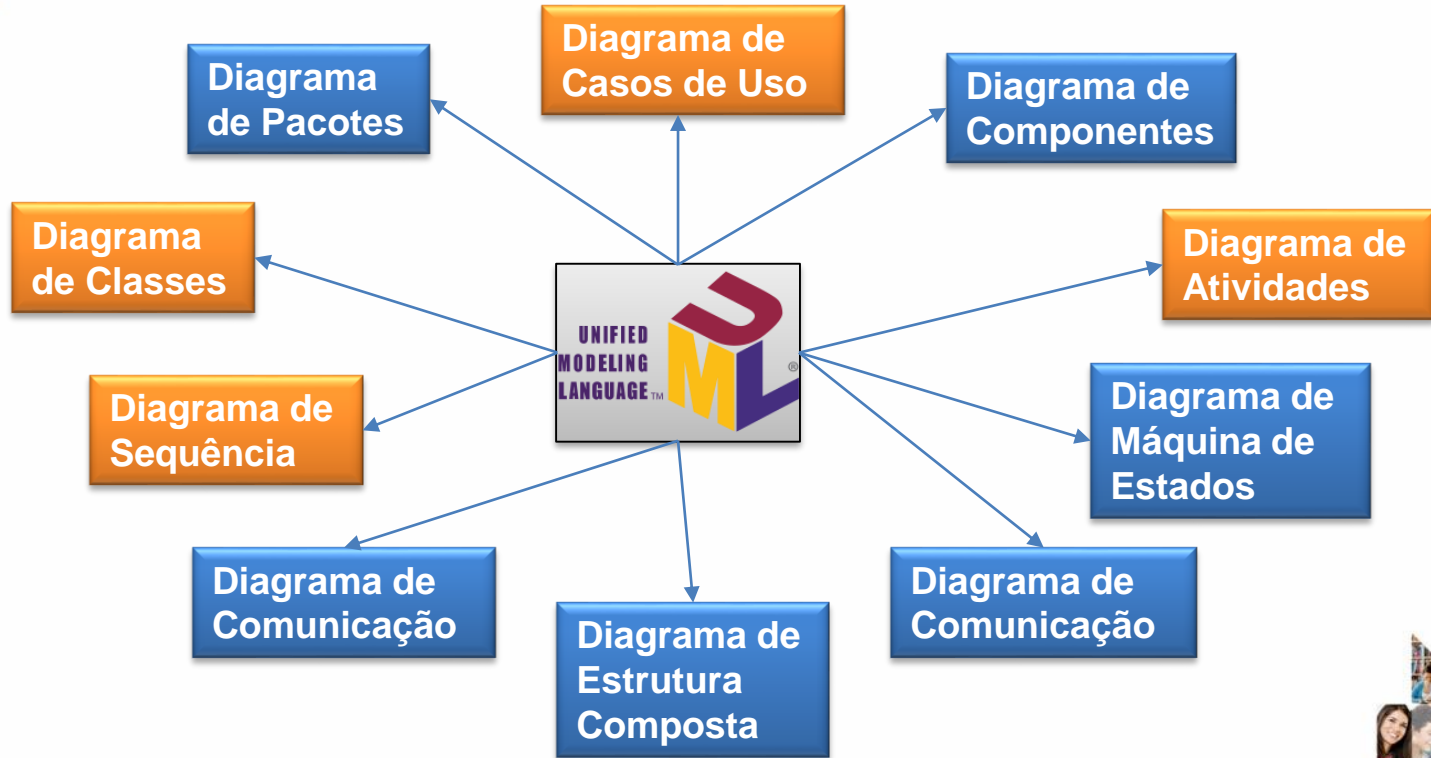
(Linguagem
Unificada para
Modelagem de
Sistemas)



<https://argouml.br.uptodown.com/windows>



Modelos



Modelagem e Desenvolvimento Orientado a Objeto

PROGRAMAÇÃO ORIENTADA A OBJETOS



Integrated Development Environment – IDE
(Ambiente de Desenvolvimento Integrado)



netbeans.org



Quais as 5 Atividades Fundamentais no Desenvolvimento de Software ?



Modelagem e Desenvolvimento Orientado a Objeto

ATIVIDADES FUNDAMENTAIS NO DESENVOLVIMENTO DE SOFTWARE

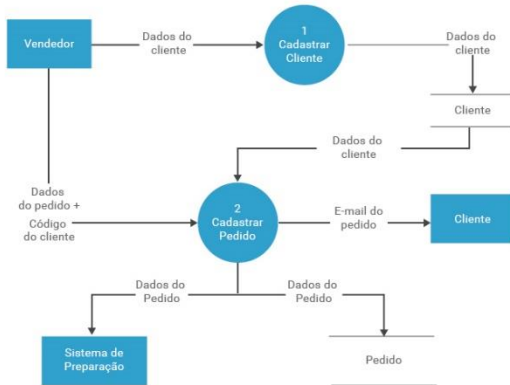


Conforme a evolução das técnicas de análise, quais as 3 formas existentes ?

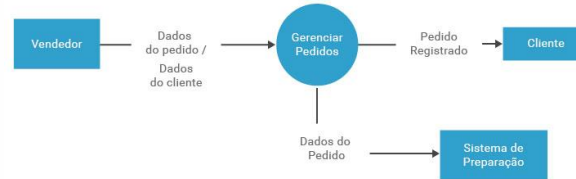
Modelagem e Desenvolvimento Orientado a Objeto

Técnicas de Análise

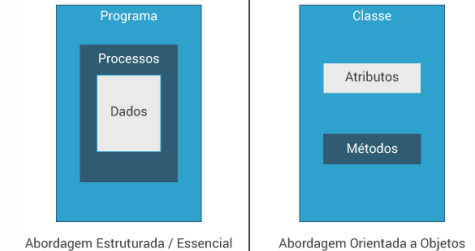
Análise Estruturada



Análise Essencial



Análise Orientada a Objetos



Objeto



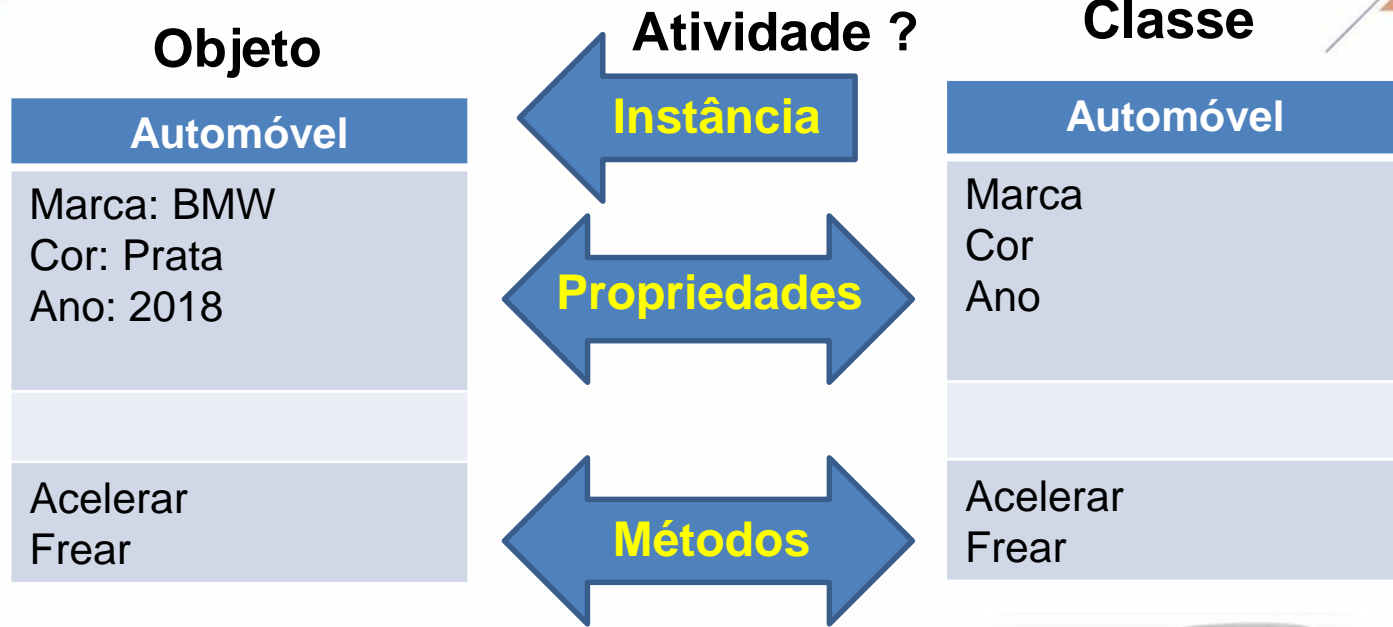
Conceito: Objeto

A **Análise Orientada a Objetos** baseia-se em alguns princípios que são utilizados para administrar a complexidade e facilitar a modelagem de sistemas. O entendimento desses princípios é essencial para a interpretação e uso de sistemas modelados orientados a objeto.

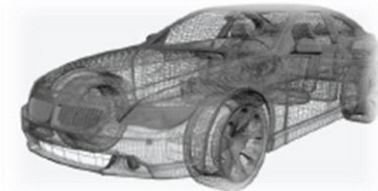
Objeto: Qualquer coisa visível ou tangível para qual a ação, pensamento ou sentimento é direcionado, ou seja, elemento do mundo real que possua atributos (**propriedades**) e comportamentos (**métodos**).

Modelagem e Desenvolvimento Orientado a Objeto

Termos OO



Atividade ? **Abstração**



Classe ? **Modelo ou Protótipo**



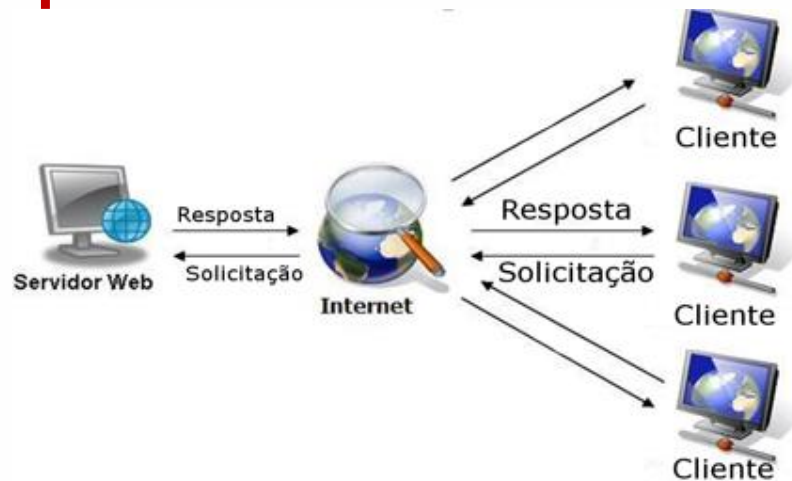
Arquitetura Cliente / Servidor

Tipos de Arquiteturas



Modelagem e Desenvolvimento Orientado a Objeto

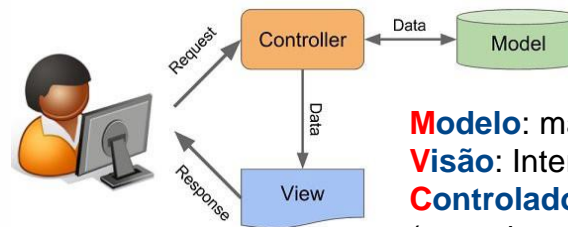
Arquitetura Cliente / Servidor



Arquitetura em Camadas



Arquitetura MVC



Modelo: manipulação de dados.
Visão: Interface gráfica do usuário.
Controlador: tratamento de dados (pesquisar, listar, calcular, etc.)
Interpretação de eventos.

CRUD (Create, Read, Update, Delete)

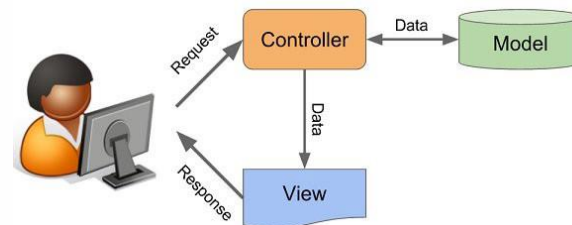
Modelagem e Desenvolvimento Orientado a Objeto

Arquitetura Cliente / Servidor

Arquitetura em Camadas



Arquitetura MVC

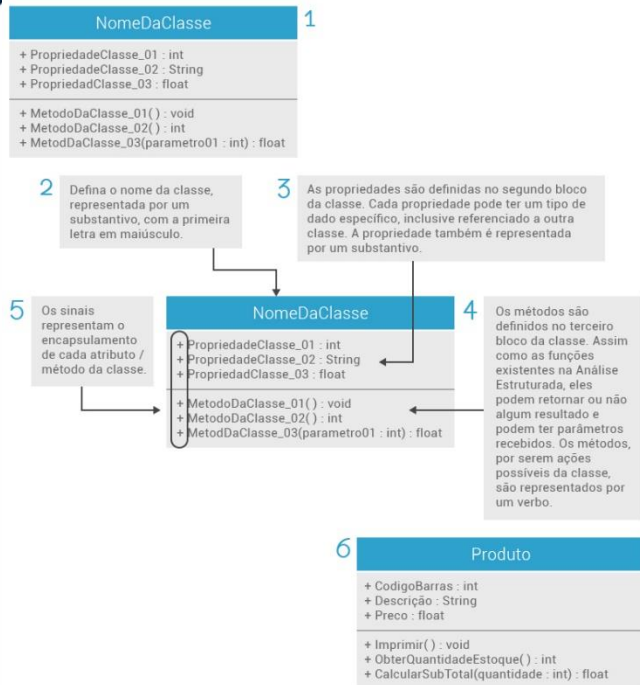


IMPLEMENTAÇÃO

Modelagem e Desenvolvimento Orientado a Objeto

Exemplo Prático

Diagrama de Classes



Codificação - Java

```
class Produto
{
    public int CodigoBarras { get; set; }

    public string Descricao { get; set; }

    public float Preco { get; set; }

    public void Imprimir ()
    {
        TODO - Escrever o método de impressão
    }

    public int ObtemQuantidadeEstoque()
    {
        TODO - Escrever o método para obter a quantidade em estoque
        return 0;
    }

    public float CalcularSubTotal(int quantidade)
    {
        return quantidade * Preco;
    }
}
```



Modelagem e Desenvolvimento Orientado a Objeto

Linguagem de Programação



- Tipos de Dados
- Operadores Aritméticos
- Operadores Relacionais e Lógicos
- Estruturas de Controle Condicional
- Estruturas de Repetição

Modelagem e Desenvolvimento Orientado a Objeto

Conceitos



Dado

**Como Dados são
transformados em
Informação**



Informação



**Princípio do
Processamento
de Dados**

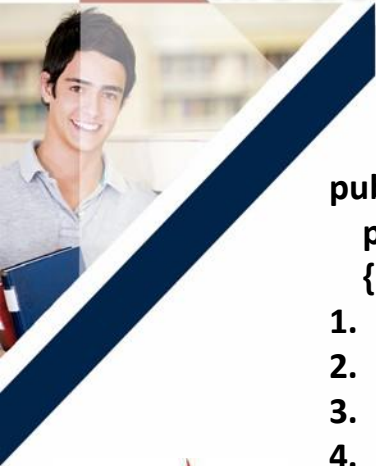


Exemplo de Aplicação



Problema		
Dados	Descrição	Valor
Vp	Valor Presente	R\$ 10.000,00
i	Taxa Juros	5% -> 0.05
n	Nº Parcelas	12
Vf	Valor Futuro	???
Fórmula		
$Vf = Vp \times (1 + i)^n$		
Exemplo		
$Vf = 10000 \times (1 + 0.05)^{12}$ $Vf = 17.958,55$		

Exemplo de Aplicação



Declaração

Entrada

Conversão

Processamento

Saída



```
public class Aplicacao_Financeira {  
    public static void main(String args[])  
    {  
1.        String vpx, tax, przx;  
2.        float vp, taxa;  
3.        int prazo;  
4.        double vf;  
5.        NumberFormat vfx = new DecimalFormat("R$ ###,##0.00");  
6.        vpx = JOptionPane.showInputDialog("Valor Presente: ");  
7.        tax = JOptionPane.showInputDialog("Taxa: ");  
8.        przx = JOptionPane.showInputDialog("Prazo: ");  
9.        vp = Float.parseFloat(vpx);  
10.       taxa = Float.parseFloat(tax);  
11.       prazo = Integer.parseInt(przx);  
12.       vf = vp * Math.pow(1+taxa, prazo);  
13.       JOptionPane.showMessageDialog(null, "Valor Futuro: " + vfx.format(vf));  
    }  
}
```



Modelagem e Desenvolvimento Orientado a Objeto

Exemplo de Aplicação



Entrada

?

Valor Presente:

10000

OK Cancelar

Entrada

?

Taxa:

0.05

OK Cancelar

Entrada

?

Prazo:

12

OK Cancelar

Mensagem

i

Valor Futuro: R\$ 17.958,55

OK



Modelos

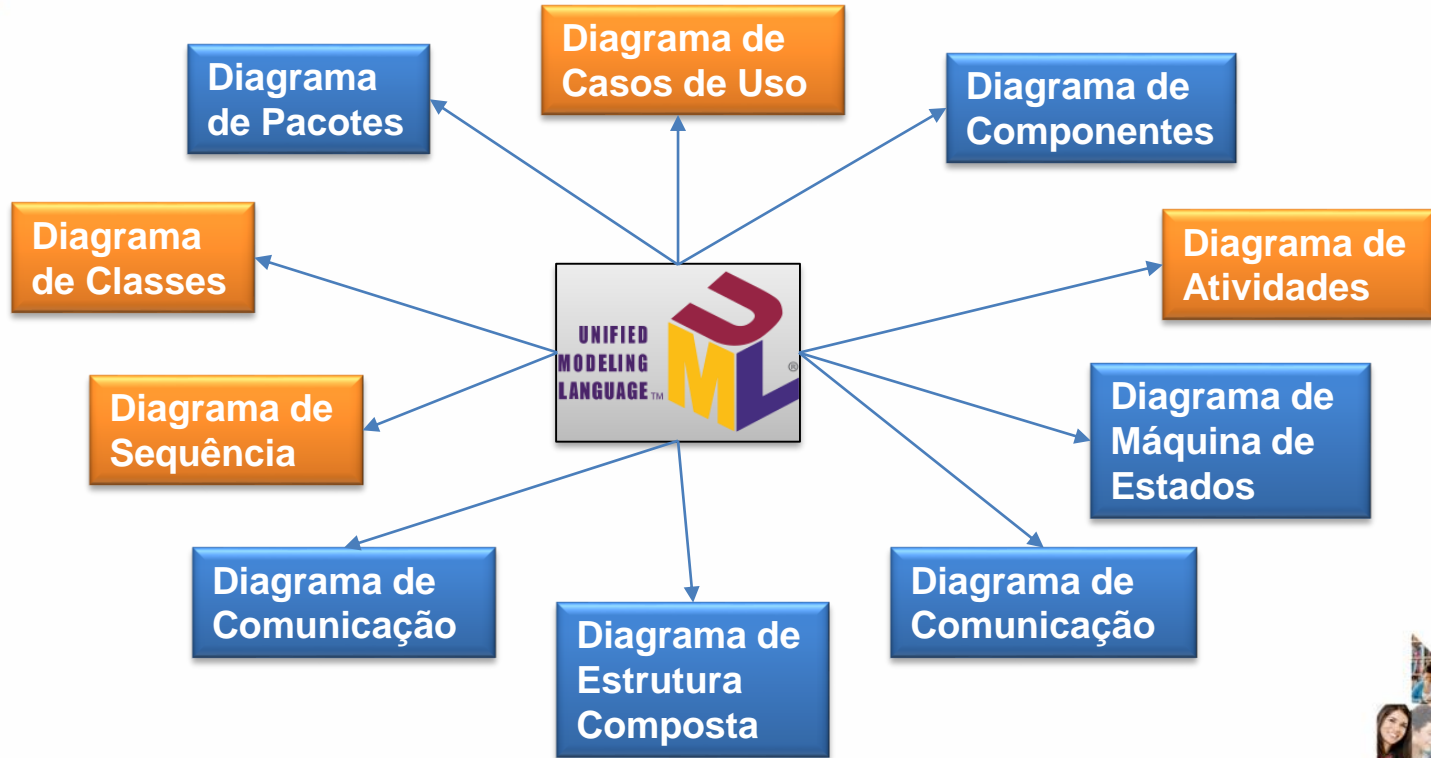


Diagrama de Casos de Uso

O diagrama de Casos de Uso é uma abordagem apresentada na UML – *Unified Modeling Language*, ou Linguagem de Modelagem Unificada para representar o comportamento de um sistema que está sendo modelado. Por tal razão, é muito comum utilizá-lo como ferramenta para substituir a especificação de requisitos funcionais.

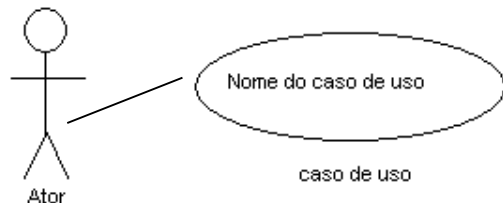
Identifica os tipos de usuários (atores) que interagem com o sistema, os papéis que eles assumem e as funções requisitadas (casos de uso).

Diagrama de Casos de Uso

Elementos

Ator - é um papel que tipicamente estimula/solicita ações/eventos do sistema e recebe reações. Cada ator pode participar de vários casos de uso.

Casos de uso - documento narrativo que descreve a sequência de eventos feitos por um ator no uso do sistema.



O relacionamento entre um Ator e um Caso de Uso em UML é ilustrado com uma linha sólida contínua.

- O nome de um caso de uso pode ser qualquer sentença, mas a UML recomenda usar uma frase ativa curta (*verbo + substantivo*), por exemplo: "comprar itens", "efetuar venda", ... (indicando uma ação). Os elementos principais do diagrama são uma elipse para representar um **Caso de Uso** e um pequeno boneco para representar um **Ator**.
- Um **Ator** pode ser entendido como:
 - Uma entidade externa ao sistema que participa de um ou mais casos de uso.
 - Fornecem eventos de entrada e recebem alguma resposta do sistema;
 - Podem ser: Papéis que pessoas representam nos casos de uso, organizações, dispositivos de hardware , outros sistemas, temporizador;

Modelagem e Desenvolvimento Orientado a Objeto

Diagrama de Casos de Uso

Elementos



Como identificar Atores ?

Para identificar os atores que vão participar do modelo pode-se fazer as seguintes perguntas :

- Quem usa o sistema ?
- Quem inicia o sistema ?
- Quem de fato interage com a atividade ?
- Quem fornece os dados ?
- Quem usa as informações ?



Diagrama de Casos de Uso

Elementos

Como Identificar Casos de Uso ?

Os casos de uso são interações entre os atores e o sistema . Temos então ações do ator e ações do sistema. Sendo que os atores sempre iniciam a ação.

Vamos dar um exemplo prático para que tudo fique mais claro. Vamos supor , por questão de simplicidade , que temos que modelar usando casos de uso a compra de item em um a loja com um atendente de vendas.

Quais são os atores ?

Quem usa o sistema é o **cliente** e ele interage com o **atendente da loja**.

Como podemos identificar o caso de uso ?

Podemos chamar este caso de uso de : **Comprar Item** (*verbo+substantivo*)

Descrição textual do caso de uso **Comprar Item** os atores **Cliente e Atendente**:

Caso de uso - Comprar Item

Atores - Cliente , Atendente

Descrição - Este caso de uso começa quando um cliente solicita ao atendente, itens que deseja comprar.

O atendente registra os itens , recebe o pagamento e emite uma nota fiscal.

O Cliente recebe os itens comprados.

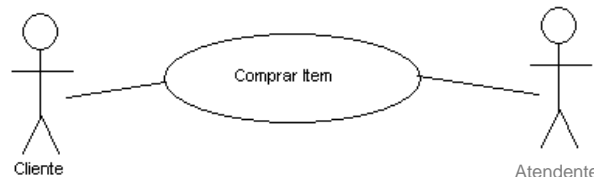


Diagrama de Casos de Uso

Relacionamento (detalhamento) Inclusão

Um Caso de Uso gera ou adiciona um novo comportamento ou outro Caso de Uso. Neste caso, um Caso de Uso obriga à execução do outro. O relacionamento de inclusão em UML é ilustrado com uma seta tracejada com o rótulo **<<include>>** na direção do Caso de Uso gerado.

----->

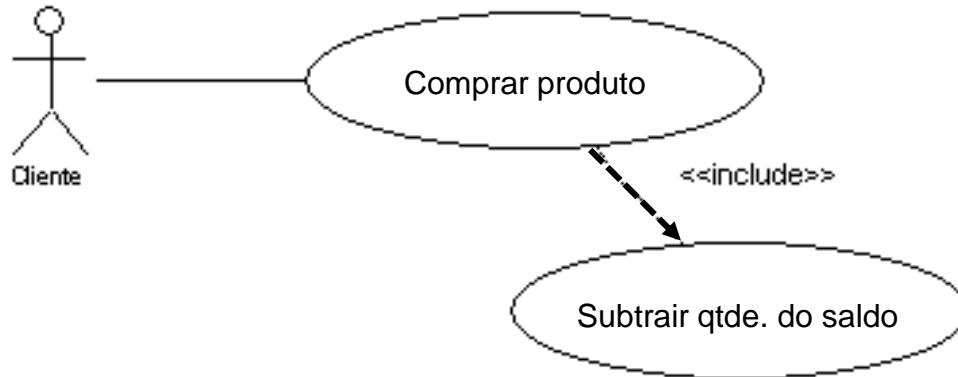


Diagrama de Casos de Uso

Relacionamento (detalhamento) Extensão

A um **Caso de Uso** é estendido uma variação de comportamento a partir de um outro **Caso de Uso**. Neste caso, o Caso de Uso base pode ser executado sem a extensão, ou seja, só ocorrerá se determinada condição for satisfeita.

O relacionamento de extensão em UML é ilustrado com uma seta tracejada com o rótulo **<<extend>>** na direção do Caso de Uso base.

----->

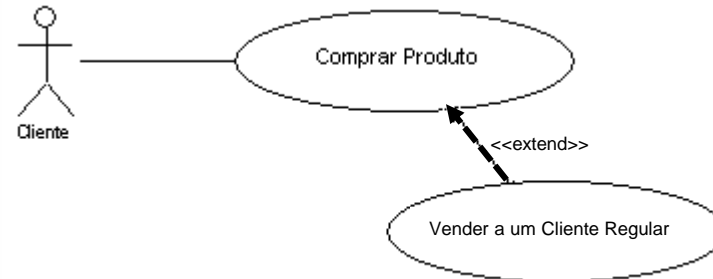


Diagrama de Casos de Uso

Relacionamento (detalhamento)

Generalização ou Herança

Indica a utilização de características e comportamento do Ator ou Caso de Uso mais genérico, ou seja, indica uma ideia de Herança.

O relacionamento de generalização em UML é ilustrado com uma seta. ➡

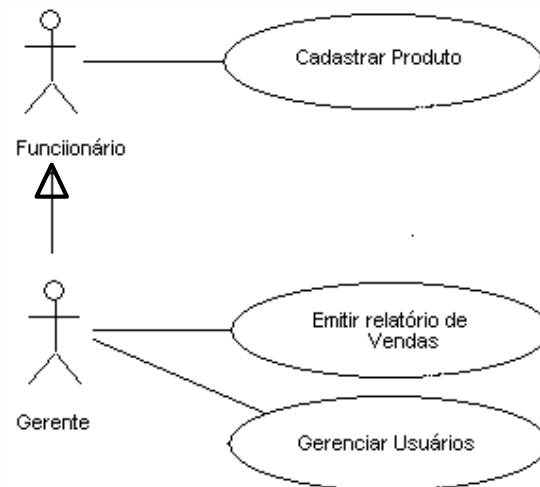
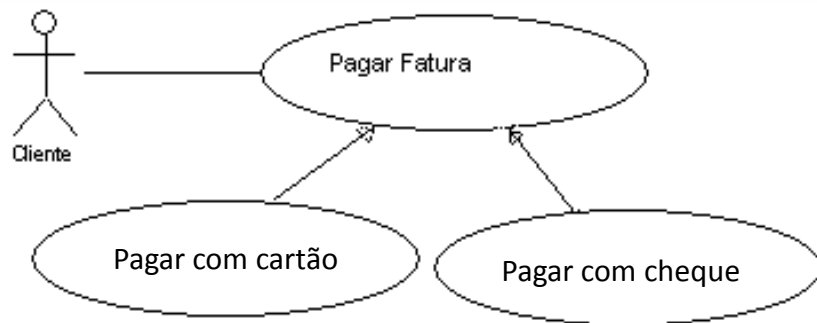


Diagrama de Casos de Uso

Características

- Mostram um conjunto de casos de uso, atores e seus relacionamentos
- Indicam como o sistema interage com as entidades externas (atores)
- Fornecem uma representação contextual do sistema

Modelagem e Desenvolvimento Orientado a Objeto

Diagrama de Casos de Uso

Notação (resumo)







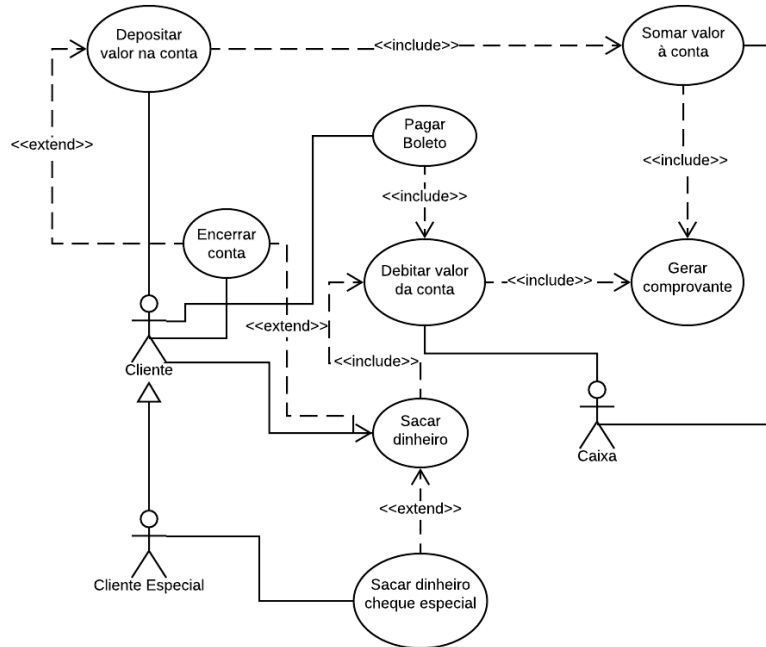
ELEMENTO	REPRESENTAÇÃO GRÁFICA	OBJETIVO
Ator		Define um conjunto coerente de papéis que os usuários do sistema podem desempenhar ao interagir com ele. Pode ser considerado tanto um indivíduo como um sistema externo.
Caso de Uso		Define uma sequência de ações realizadas por um sistema que produz um resultado de valor observável para determinado ator.
Relacionamento Simples		Linha que interliga os atores aos casos de uso que eles têm permissão de executar
Relacionamento Herança		Ideia de generalização. Pode ser aplicado tanto entre casos de uso como entre atores. Dessa maneira, é possível aproveitar características do caso de uso ou ator mais genérico.
Relacionamento Extensão		Representa que a ação do sistema que está sendo estendida pode ser executada pelo caso de uso que a estendeu. A seta é direcionada para o caso de uso que poderá executar o caso de uso opcional.
Relacionamento Inclusão		Representa que a ação do sistema será executada pelo caso de uso que a inclui. A seta é direcionada para o caso de uso que será incluído

Diagrama de Casos de Uso

Atendimento Bancário

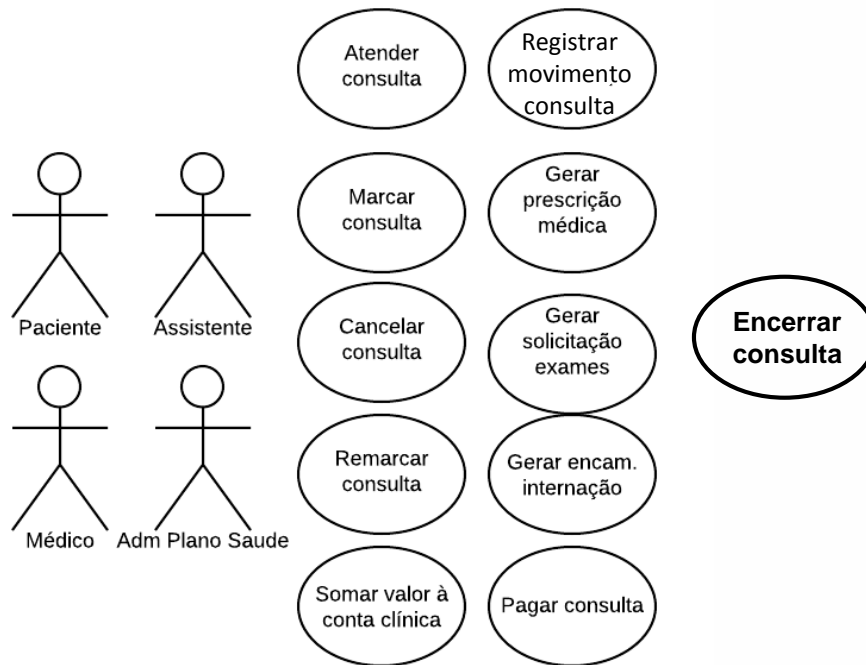


Editor UML Online

 **Lucidchart**
www.lucidchart.com

Diagrama de Casos de Uso

Clínica Médica



Modelagem e Desenvolvimento Orientado a Objeto

Diagrama de Casos de Uso

Biblioteca

Para usar os serviços de uma biblioteca, os leitores deverão estar registrados e possuir um cartão com número de identificação e foto. O sistema deve permitir que um leitor empreste um ou mais livros, por um período de tempo.

O sistema deve permitir que o leitor devolva um ou mais livros em seu poder, fazendo com que o(s) livro(s) volte(m) a ficar disponível na biblioteca. O empréstimo, devolução e a manutenção do cadastro de leitores é realizado por um atendente da biblioteca.

A manutenção (catalogação, exclusão e alteração) de livros do acervo da biblioteca é responsabilidade do bibliotecário.

Destacar Atores e Casos de Uso.

Modelagem e Desenvolvimento Orientado a Objeto

Diagrama de Casos de Uso

Locadora de Automóvel

O locatário solicita ao locador o aluguel do carro. O sistema verifica se o carro solicitado pelo locatário está disponível. Caso esteja, o processo de reserva é realizado, porém o automóvel só será liberado mediante a aprovação do gerente da loja. Após a aprovação do gerente, será gerado então uma autorização de locação onde o automóvel estará liberado para uso.

Para tanto, o inspetor de veículo anotará a data e hora da entrega e saída do automóvel. No retorno do automóvel, o inspetor, verificará as condições do veículo e anotará a data e hora de retorno para cálculo da cobrança do aluguel o qual o locador deverá realizar o pagamento.

Destacar Atores e Casos de Uso.

Modelagem e Desenvolvimento Orientado a Objeto

Diagrama de Casos de Uso

Faculdade

Um aluno, após ter sido aprovado no vestibular da faculdade, solicita sua matrícula. A secretaria da faculdade então efetua o cadastramento do aluno mediante documentação previamente solicitada. O aluno depois de matriculado, é cadastrado em uma determinada turma da faculdade de acordo com o curso escolhido. A secretaria cuida também da administração dos cursos. O professor, após a aplicação de avaliações, deve lançar as notas assim como realizar o controle da frequência dos alunos. O aluno pode, eventualmente, consultar sua frequência e notas, assim como, deve realizar o pagamento da mensalidade do curso através do boleto de pagamento gerado pelo departamento financeiro da faculdade.

Destacar Atores e Casos de Uso.

Modelagem e Desenvolvimento Orientado a Objeto

Diagramas

Para o desenvolvimento de um sistema é necessário uma perspectiva mais detalhada, assim, o diagrama de **Casos de Uso**, além do diagrama de **Classes**, deve vir acompanhado do diagrama de **Atividades** ou do diagrama de **Sequência**.

Modelagem e Desenvolvimento Orientado a Objeto

