# UML

1. Fundamentos de Análise de Sistemas orientada a objetos
2. Tipos de dados
3. Conceitos de UML

## Fundamentos de Análise de Sistemas orientada a objetos

Análise Orientação a Objetos

O que é? Para que serve?

É um seguimento para o desenvolvimento de sistemas que utiliza o conceito de objetos que se integram entre si tendo características próprias, e dessa integração realizam tarefas computacionais

Conceito de Objetos em Programação:

Objeto é um molde/classe, que passa a existir a partir de uma instancia da classe, a instancia da classe é um objeto cujo comportamento e estado são definidos pela classe, ou seja, objetos são blocos(classes) onde contêm métodos, ou atributos.

### Abstração:

É a ação intelectual de isolar fatores que normalmente estão relacionados ao objeto real, ou seja, deixar apenas características que consiga definir algo como um objeto deixando apenas características gerais sem deixar características únicas.

### Encapsulamento de Informações:

Um objeto contém informações que não são acessíveis ao outros objetos.

### Atributos:

São características que identificam o objeto, um atributo tem valor único para um objeto.

### Operações:

São serviços requisitados ao objeto que faz parte de seu comportamento em resposta a solicitações externas, então as operações são transformações realizadas ou sofridas por um objeto.

### Mensagens:

Os objetos se comunicam através de sinais enviados de um objeto ao outro requisitando um serviço ao objeto requisitado.

### Polimorfismo:

É dado pela capacidade de uma mesma operação de assumir varias formas e ter vários comportamentos e a capacidade de uma variável a se referir a diferentes objetos que preenchem as responsabilidades em função ao que é passada a elas

## Tipos de dados

### Byte:

Inteiro de 8 bits sem sinal.

### Sbyte:

Inteiro com sinal de 8 bits.

### Int:

Inteiro de 32 bits com sinal.

### Uint:

Inteiro de 32 bits sem sinal.

### Long:

Inteiro com sinal de 64 bits.

### Ulong:

Inteiro sem sinal de 64 bits.

### Short:

Inteiro com sinal de 16 bits.

### Ushort:

Inteiro sem sinal de 16 bits.

### Decimal:

Ponto flutuante decimal. esse tipo tem precisão de 28 cassas decimais.

### Double:

Ponto flutuante binário. esse tipo tem precisão de 15 cassas decimais.

### Float:

Ponto flutuante decimal. esse tipo tem precisão de 7 cassas decimais.

### Bool:

Tipos de dados bool tem apenas dois valores verdadeiro ou falso (true e false)

### Char:

Um único caractere Unicode de 16 bits.

### String:

Unicode com até 1 gigabyte de caracteres.

Conceito de Unicode:

Unicode é um padrão que permite aos computadores manipularem de forma consistente textos de qualquer sistema de escrita.

## Conceito de UML

UML – Acrônimo de Unified Modeling Language(Linguagem de Modelagem Unificada) é a a linguagem padrão para a elaboração de estruturas de projeto de software. Ela poderá ser empregada para a visualização, a Especificação, a construção e a documentação de **artefatos** que façam uso de sistemas complexos de software.

### Artefatos:

São subprodutos concretos produzidos durante um desenvolvimento de software (ex: diagramas de classes, requisitos e documentos do projeto.) tais artefatos ajudam a descrever a função e a arquitetura e o design do software.

A UML possui 14 diagramas separados em

### Diagramas de Estrutura:

Diagrama de Classes

Diagrama de componentes

Diagrama de Implantação

Diagrama de objetos

Diagrama de pacote

Diagrama de perfil

Diagrama de estrutura composta

### Diagramas Comportamentais:

Diagrama de caso

Diagrama de atividade

Diagrama de máquina do estado

Diagrama de sequência

Diagrama de comunicação

Diagrama de visão geral de interação

Diagrama de tempo

# Diagrama de caso

#### O que é? Pra que server?

#### Quais são os elementos?

#### Como recomenda-se nomear os casos de uso?

#### Tipo de Relacionamentos?

O que é? Pra que server?

Resume os detalhes dos usuários do sistema, suas interações com o sistema.

Quais são os elementos?

Os elementos são:

### Atores:

Os usuários que interagem com o sistema, pode ser uma pessoa um grupo, que interage com o sistema que está sendo desenvolvido eles devem ser objetos externos que produzam ou consumam dados (ou seja que tenha uma interação e uma relação de uso real com o sistema)

### Sistemas:

Sistema é uma sequencia especifica de ações e interações entre os atores e o sistema, o sistema é conhecido também como cenário.

### Metas:

O resultado final da maioria dos casos de uso, para ser considerado um bom diagrama de uso será necessário que tenha sido descrito as atividades e variantes usadas para atingir a meta

Como recomenda-se nomear os casos de uso?

Deve se ter um nome que seja possível a identificação e a diferenciação dos demais casos de uso, geralmente é utilizado expressões verbais que expõe comportamentos no vocabulário do sistema que está sendo projetado.

Tipo de Relacionamentos?

Os relacionamentos

### Include:

O relacionamento de (include) é o de inclusão ou agregação

### Extend:

O relacionamento de (extend) é o de Extensão ou generalização

### Herança:

Diagrama

Descrição gerada automaticamente

Exemplo modelo de caso de uso:

Diagrama

Descrição gerada automaticamente

# Diagrama de classes

1. O que é? Pra que server?

#### Quais os tipos de relacionamentos?

#### O que é multiplicidade? Quais os tipos?

#### Exemplo de modelo de Classe.

O que é? Pra que server?

É uma representação da estrutura e relação das classes que servem de modelo para objetos. É uma modelagem muito útil para o desenvolvimento do sistema (ou seja, serve para definir todas as classes que o sistema necessita, para seu desenvolvimento).

Quais os tipos de relacionamentos?

### Agregação:

### Composição:

### Generalização:

### Dependências:

O que é multiplicidade? Quais os tipos?

A multiplicidade nada mais é do que os limites inferior e superior da quantidade de objetos aos quais outro objeto está associado. Esses limites podem receber os valores: apenas um; zero ou muitos; um ou muitos; zero ou um; ou até um intervalo específico

Exemplo de modelo de Classe:

Diagrama

Descrição gerada automaticamente

# Diagrama de caso de uso