

Análise orientado a objetos

Fundamentos da UML

Ma. Vanessa Matias Leite

1

- Unidade de Ensino: 01
- Competência da Unidade: Compreender os fundamentos da UML
- Resumo: Conhecer os diagramas que compõem a UML e as fases do processo unificado.
- Palavras-chave: UML; processo unificado; paradigma orientado a objetos;
- Título da Teleaula: Fundamentos da UML
- Teleaula nº: 01

2

Características da UML

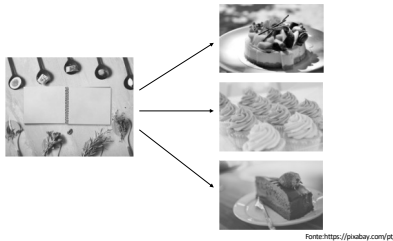
3

Paradigma Orientado a Objetos

- Linguagens como JAVA, C++, C#, PHP;
- Relacionamento entre classes e objetos e o relacionamento entre eles:
 - Herança;
 - Polimorfismo,
 - Agregação
 - composição

4

Paradigma Orientado a Objetos



5

Classe	Define o comportamento de seus objetos - através de métodos - e os estados possíveis destes objetos - através de atributos.
Objeto	Instância de um classe.
Herança	As classes compartilham seus atributos, métodos e outros membros da classe entre si.
Polimorfismo	Métodos que têm a mesma assinatura, mas comportamentos distintos.
Encapsulamento	Proibição do acesso ao direto ao estado do objeto.

6

Características da UML

- Combina os conceitos comuns de linguagens OO;
- Compatível com o desenvolvimento de software desde os requisitos até as etapas finais do desenvolvimento;
- Compatível com diversos escopos;

7

Objetivos da UML

- Modelar diferentes linguagens e situações;
- Padrão para o desenvolvimento de software;
- Simplicidade;

8

Modelos

- Capturar e definir com precisão os requisitos do software;
- Auxiliar o início do projeto do sistema;
- Solução que contenha as decisões de projeto;
- Explorar diferentes soluções;
- Permitir o fácil entendimento de projetos complexos.

9

Fluxo de desenvolvimento

- Pode ser utilizada em qualquer fluxo de desenvolvimento;
- Apresentação visual da semântica do sistema;
- Contexto;

10

Diagramas UML

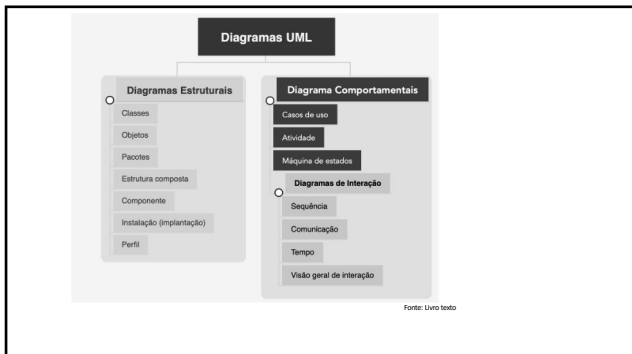
11

Nível de abstração

Nível de Abstração	Objetivo do Diagrama
ALTO	Ser claro e simples, apresentar os conceitos ao cliente para tomada de decisão
MÉDIO	Guiar o desenvolvimento apresentado, sem detalhar demais, as classes, os objetos e as interações
BAIXO	Demonstrar como deve ser desenvolvido o sistema propriamente dito. Necessita de diagramas e modelos com a especificação completa de cada módulo, interação e outras informações que possam ser necessárias

Fonte: Livro texto

12



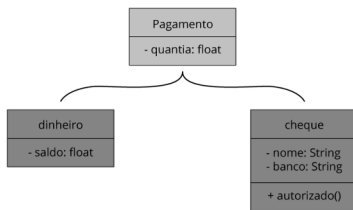
13

Diagrama de classes

- Classe:
 - Atributos;
 - Métodos;
- Relacionamento:
 - Associação;
 - Herança;
 - Polimorfismo;

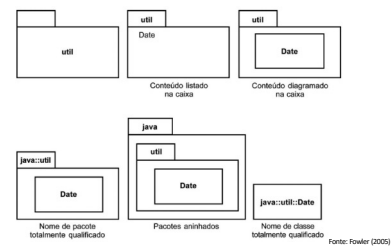
14

Diagrama de classes



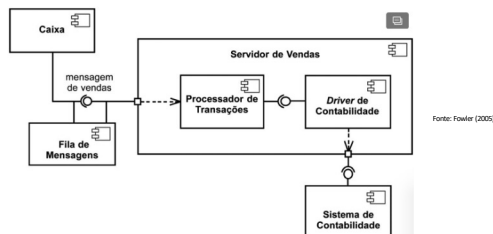
15

Diagrama de Pacotes



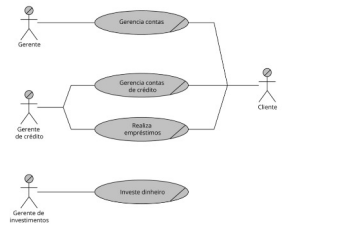
16

Diagrama de Componentes



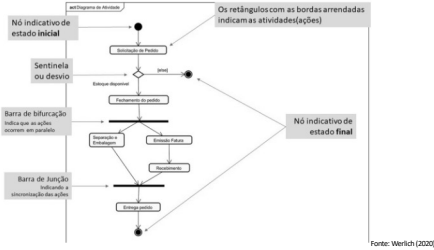
17

Diagrama de caso de uso

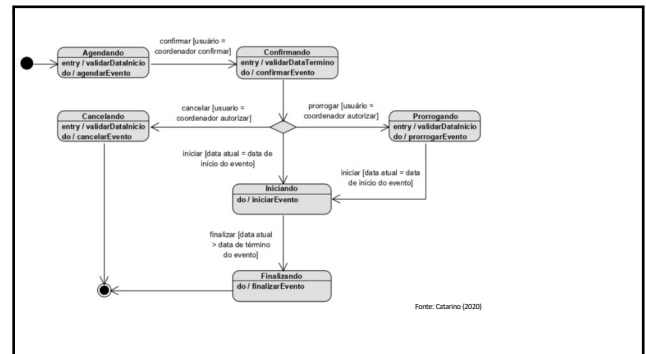


18

Diagrama de Atividades

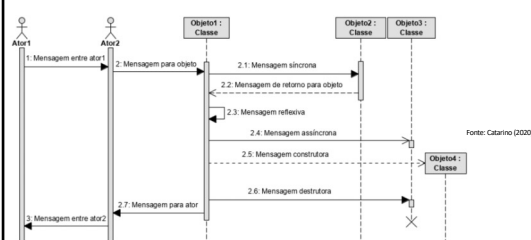


19



20

Diagrama de Sequência



21

Processo Unificado

22

Processo Unificado

- Processo — quem, o que, como e quando;
- Processo Unificado (PU);
- RUP (do inglês *Rational Unified Process*);

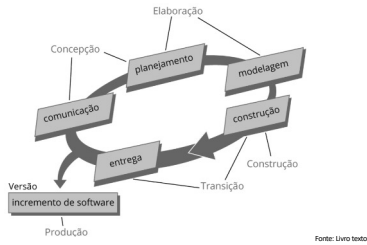
23

Processo Unificado

- Interativo e incremental;
- Dirigido por uma lista de casos de uso;
- Focado na arquitetura do sistema;
- Orientado a riscos;

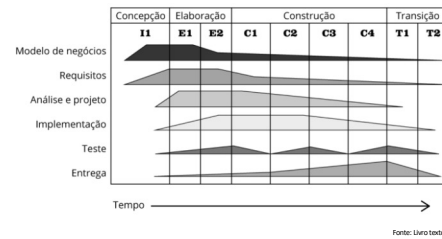
24

Fases do processo unificado



25

Fluxo de trabalho



26

Fase de Conceção

- Diagrama de Caso de Uso;
- Diagramas de sequência;
- Diagrama de colaboração;
- Diagrama de atividades;
- Diagrama de máquinas de estado;

27

Fase de elaboração

- Diagrama de classes;
 - Diagramas de sequência;
 - Diagrama de colaboração;
 - Diagrama de atividades;
 - Diagrama de máquinas de estado;
- Evolução

28

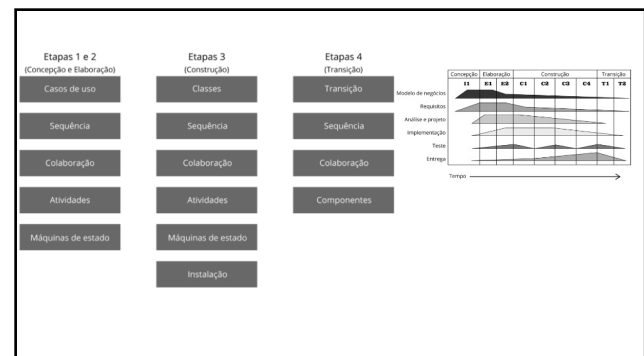
Fase de construção

- Diagrama de instalação;

Fase de Implementação

- Diagrama de classes;
- Diagramas de sequência;
- Diagrama de colaboração;
- Diagrama de atividades;

29



30

Mecanismos comuns da UML

31

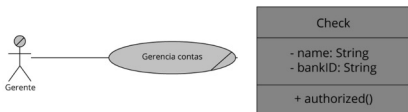
Especificação

- Descrição exata do elemento;
- No diagrama de classe existe uma especificação;
 - Atributos;
 - Operações;
 - Comportamentos;

32

Adorno

- Notação gráfica dos modelos;



Fonte: Livro texto

33

Divisões comuns

Na modelagem de sistemas orientados a objetos, costuma ser dividido pelo menos de duas maneiras:

- Divisão de classes e objetos. Uma classe é uma abstração; um objeto é uma manifestação concreta dessa abstração.
- Separação de interface e implementação. Uma interface declara um contrato e a implementação representa uma realização completa e fiel desse contrato.

34

Mecanismos de extensão

- Estereótipos: é possível, na UML, utilizar o “desenho” de um determinado bloco e modificá-lo para um propósito específico, criando um novo objeto.
- Restrições: é possível, também na UML, alterar as restrições na construção de um diagrama. Em UML, as restrições são representadas pelas strings que acompanham as ligações entre elementos.

35

Mecanismos de extensão

- Valores predefinidos: é possível predefinir valores específicos em um diagrama, para guiar a implementação do sistema ou gerenciamento de configurações do sistema.

36

Regras de consistência de diagramas UML

- O número de objetos no diagrama de sequência deve ser o mesmo do número de classes;
- Deve se atentar para as atualizações do diagrama de classes e reproduzi-las corretamente no diagrama de sequência.
- O nome dos métodos deve ser respeitado entre os diagramas de classe e sequência

37

Regras de consistência de diagramas UML

- Os diagramas de classe e sequência devem ser sincronizados;
- Cada uma das situações representadas no diagrama de casos de uso deve ter uma operação correspondente no diagrama de classes.
- Cada caso de uso deve ter um substantivo e um verbo associados.

38

Regras de consistência de diagramas UML

- Para cada caso de uso deve existir ao menos um diagrama de sequência;
- Deve haver consistência entre os atores do diagrama de casos de uso e o de sequência.

39

Recapitulando

40

Recapitulando

- Características da UML;
- Diagramas UML;
 - Estruturais e comportamentais;
- Processo Unificado;
- Mecanismos comuns da UML;

41