

Análise e Design

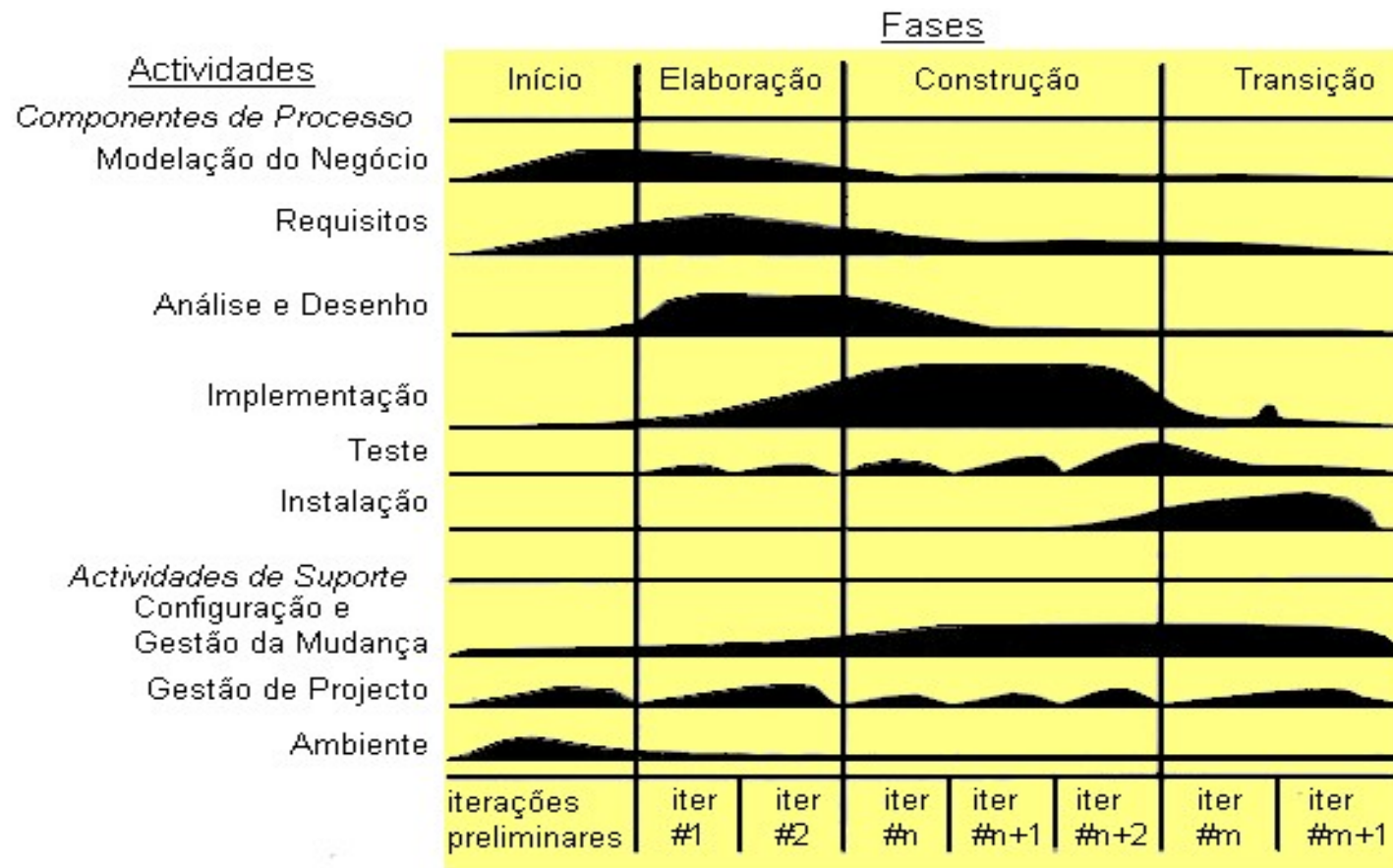
Abordagem Orientada por Objectos

Prof. Doutor Carlos J. Costa

Temas a Tratar

- **Análise e *Design***
- **Abordagem Orientada por Objectos**
 - **Conceitos**
 - **Características da Abordagem**
 - **Objectos, Classes e Instancias**

Análise e Design



Análise e Design

- **Análise é frequentemente relaciona-se com o “O Quê” do sistema**
- ***Design* (ou desenho) descreve o que se relaciona com o “Como”**

Análise

- Os modelos produzidos na análise mostram o que é o sistema, e como as várias componentes do sistema se relacionam entre si.
- Permite caracterizar o que existe no sistema actual e o que é que se pretende implementar.

***Design* (ou Desenho)**

- Os modelos que são produzidos no design mostram como as partes do sistema funcionam em conjunto.
- A actividade de *design* relaciona-se com a resposta do sistema aos requisitos.

Análise e *Design*

Ferramentas:

- Com vista a apoiar na actividade de análise e *design* é produzido o diagrama de classes de objectos.

Visão tradicional

- A visão tradicional no desenvolvimento de *software* adoptava a perspectiva de um algoritmo.
- Nessa visão, o principal bloco de construção do *software* é o procedimento ou função

Abordagens Orientadas por Objectos

- Os principais elementos de construção de todos os Sistemas é são:
 - Objecto
 - Classe de Objectos

Objecto

- **“Objecto é algo que ocupa espaço no mundo real ou conceptual e com o qual se pode fazer coisas” (Booch et al. 1999)**
- **Os objectos têm:**
 - **Nome (ou Identidade)**
 - **Estado**
 - **Operações (ou Comportamento)**

Objecto

- **Nome (Identidade)** – Todo o objecto deve ter um nome que o diferencie de outros objectos num contexto (e.g. minha máquina de calcular)
- **Estado** – Um objecto tem estado, o que envolve as propriedades do objecto conjuntamente com os valores dessas propriedades (e.g. calculadora ligada)
- **Operações (Comportamento)** – podemos fazer algo com o objecto ou o objecto poderá fazer algo com outro objecto (e.g. calculadora faz somas)

Classe e Instância

- **Classe** - É a descrição de um conjunto de objectos que compartilham os mesmos atributos, operações, relacionamentos e semântica. (e.g. máquinas de calcular).
- **Instância** - Um objecto é uma instância ou ocorrência de uma classe. Manifestação concreta de uma abstracção. (e.g. “a minha máquina de calcular” é uma instância da classe “máquinas de calcular”).

Características da Abordagem

- A abordagem orientada por objectos tem como principais características:
 - Abstracção
 - Encapsulamento
 - Hierarquia

Abstracção

- **Princípio que consiste em ignorar os aspectos de um assunto que não são relevantes para o propósito actual, de forma a nos concentrarmos nos aspectos que são realmente relevantes.**
- **Abstracção é a representação concisa dum objecto mais complexo, incidindo sobre as características essenciais do objecto.**
- **Abstracção Boa**
 - **Apropriada (Existindo uma necessidade real consegue ser satisfeita)**
 - **Nível adequado**

Encapsulamento

- Mecanismo que permite esconder a implementação do objecto, de forma que os outros componentes do sistema não tenham acesso ao que se passa dentro do objecto
- Este conceito está associado à modularidade, que consiste em decompor um sistema num conjunto de módulos coesos e ligados
- Os objectos interagem por mensagens

Herança

- **Qualquer mecanismo que permite a um objecto incorporar todos ou parte das definições de outro objecto como parte de si mesmo (e.g. “médico” e “oftalmologista”)**

Bibliografia

Bennet, S. McRobb, S & Farmer, R., *Object Oriented Systems Analysis and Design using UML*, MacGarw-Hill, 1999.

Booch, G., Rumbaugh, J. & Jacobson, I, *The Unified Modeling Language User Guide*. Addison Wesley, 1999 (tradução portuguesa brasileira ____; *UML Guia do Usuário*; Campus, 2000).

Costa, C. *Desenvolvimento para Web*, ITML Press, 2007

Nunes, M & O'Neill, H. *Fundamental de UML*, FCA, 2001

Silva, A & Videira, C., *UML, Metodologias e Ferramentas CASE*, Edições Centro Atlântico, 2001

Terry, Q. *Visual Modeling With Rational Rose 2000 and UML*, Addison-Wesley. 2000.

Oxford Dictionary of Computing, Oxford University Press.