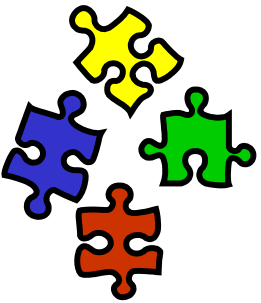
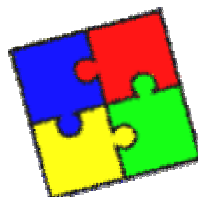
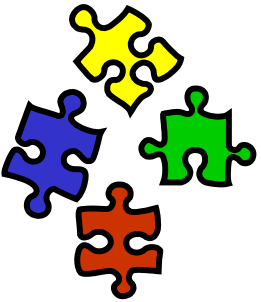


Modelagem/Arquitetura de Software

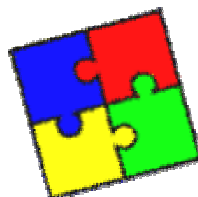
Thaís Vasconcelos Batista

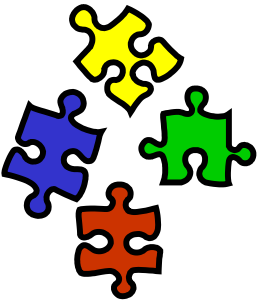




Objetivo do Curso

- *Apresentar as tendências atuais para desenvolvimento de aplicações baseadas em componentes, oferecendo uma visão conjunta das tecnologias usadas desde a modelagem até a implementação porém com o foco na* ***modelagem da aplicação****.*

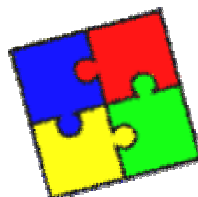


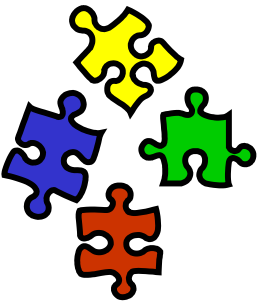


Enfoque do Curso

Modelagem de Software baseado em Componentes

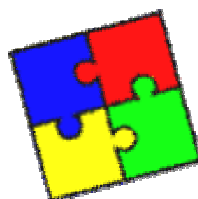
Arquitetura de Software + Componentes

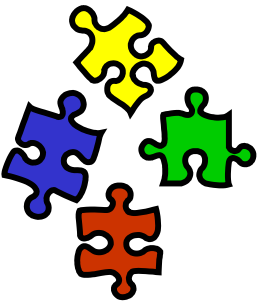




Conteúdo do Curso

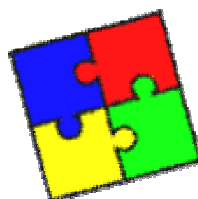
- Introdução:
  - Definições (Arquitetura de Software, Componente, Desenvolvimento baseado em Componentes, Modelos de Componentes)
- Arquitetura de Software (AS)
  - Elementos básicos de AS
  - Definição de Linguagens de Modelagem
  - Estilos Arquiteturais

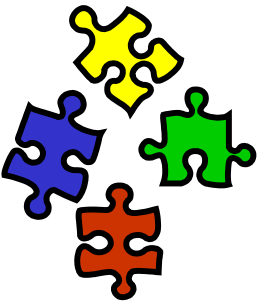




Conteúdo do Curso (cont.)

- Propostas de Padronização de Modelagem
  - UML (Unified Model Language)
  - ACME
  - MDA (Model-Driven Architecture)
- Modelos de Componentes
  - Aspectos Gerais
  - CORBA: Arquitetura, Serviços e CCM (CORBA Component Model)
  - EJB (Entreprise JavaBeans)





Bibliografia

- G.Heineman & W. Councill

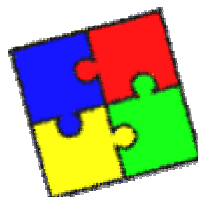
  Component-Based Software Engineering – Putting the Pieces Together – Ed. Addison Wesley.
- C. Szyperski

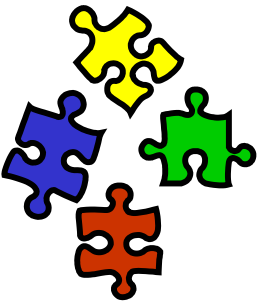
  Component Software – Beyond Object-Oriented Programming – Addison Wesley. 1998.
- C. Hofmeister, R. Nord & D. Soni

  Applied Software Architecture – Ed. Addison Wesley. 2000.
- M. Shaw & D. Garlan

  Software Architecture – Perspectives on na Emerging Discipline – Ed. Prentice-Hall. 1996.
- G. Booch, J. Rumbaugh & I. Jacobson

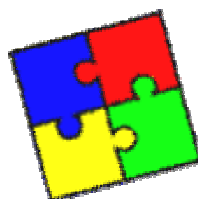
  UML – Guia do Usuário – Ed. Cammpus, 2000.

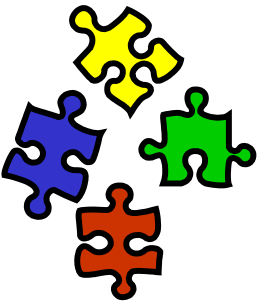




Datas Importantes

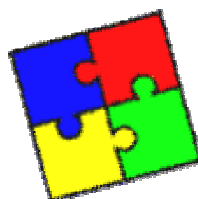
- Início das Aulas: 08/10/2002
- Fim das Aulas: 13/02/2003
- Provas: 03/12/2002 (1a. Prova)
  13/02/2003 (2a. Prova)
- Não Haverá Aula:
  - 15 e 17 de outubro
  - 23/12/2002 a 11/01/2003

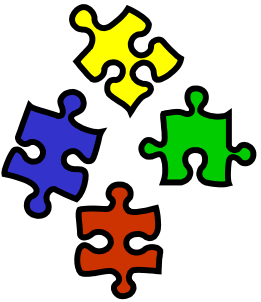




Características das Áreas

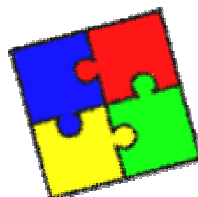
- Áreas recentes
- Variedade de Terminologias
- Diversidade de Opiniões
- Baseadas nas idéias de
  - Desenvolver software reusando partes
  - Modelar o sistema combinando componentes disponíveis e previamente testados e seguindo padrões

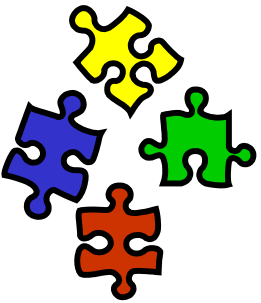




Motivação

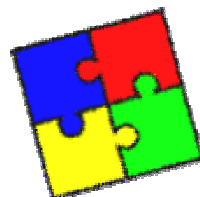
- **Reuso**: diminuir tempo e custo de desenvolvimento
- **Evitar Falhas**: componentes previamente testados são menos suscetíveis a falhas
- **Interoperabilidade**: capacidade de componentes de diferentes origens compartilharem e trocarem informações

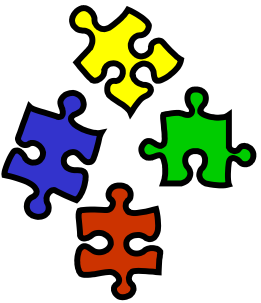




Arquitetura de Sofware

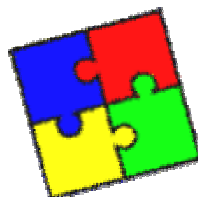
- Define *conceitos, padrões e estilos* para a composição de software formado por componentes
- *Framework* é usado muitas vezes como sinônimo de Arquitetura

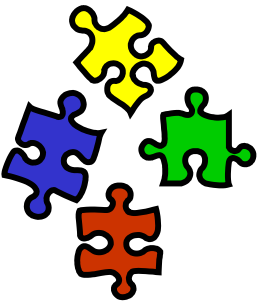




Componente

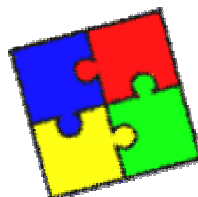
- Definições:
  - *É um elemento de software que segue um **modelo de componentes** e pode ser desenvolvido independentemente e composto através de um **padrão de composição** [B.Council and G. Heineman]*
  - *Componentes são elementos **padronizados** usados para **composição** [C. Szyperski]*

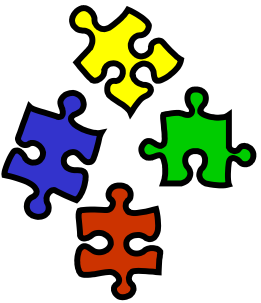




Componente

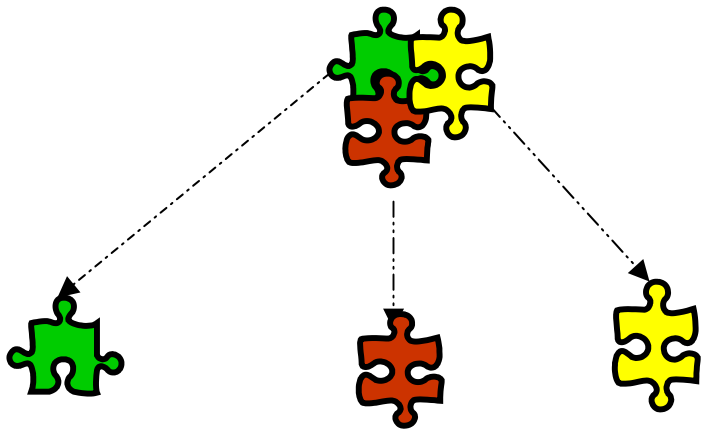
- Características:
  - Auto-contido
  - Funcionalidade bem definida
  - Definido através de interfaces que possibilita composição sem conhecimento da implementação do componente
  - Definido de acordo com um modelo de componentes

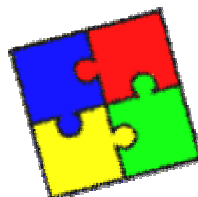


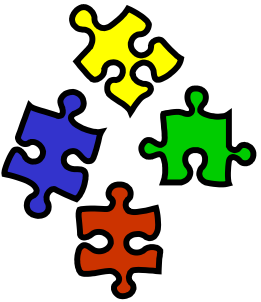


Composição

- União de porções de software “pré-fabricadas” para formar um sistema

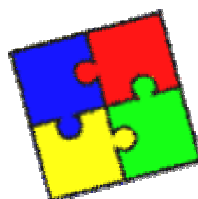


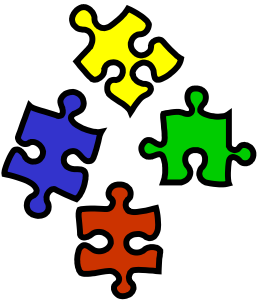




Desenvolvimento baseado em Componentes

- Mercado de Componentes
- Menos tempo de desenvolvimento
- Mais confiável (por reusar partes testadas)
- Ideal de possibilitar que o desenvolvimento de software seja uma *linha de produção em massa*

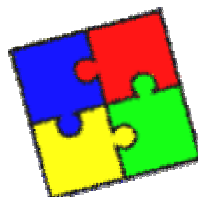


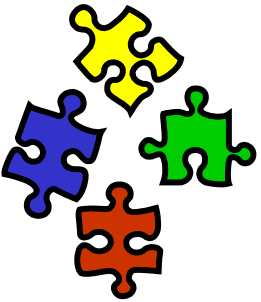


Objetos X Componentes

- A definição de objetos não inclui:
  - Noções de independência
  - Composição

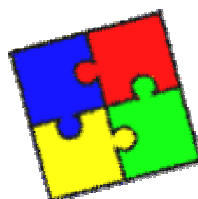
Apesar destes aspectos poderem ser adicionados, a tecnologia de objetos é mais usada para construir aplicações monolíticas [C.Szyperski]

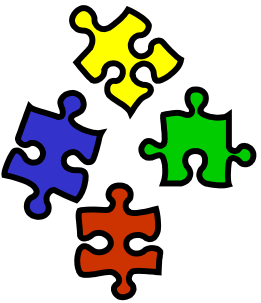




Exemplo do Uso da Idéia de Componentes

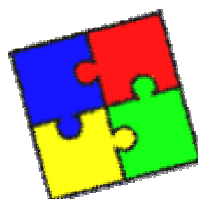
- Sistemas Operacionais: aplicações são componentes executando sobre eles (compartilhando arquivos e fazendo composição via pipe e filtros)
- Plug-in: Browsers Netscape
- Visual Basic

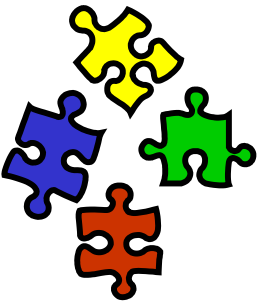




Modelo de Componentes

- Determina a forma como um componente deve ser desenvolvido
  - Exemplos
    - COM (Component Object Model) da MicroSoft determina que cada componente ofereça uma interface *IUnknown*
    - CORBA da OMG determina que o componente tenha uma interface escrita em IDL (Interface Definition Language)
- Determina um padrão de interação

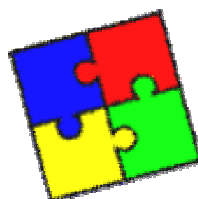


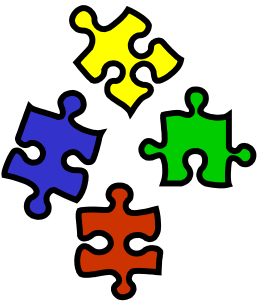


Modelo de Componentes

- Define **padrões** para:

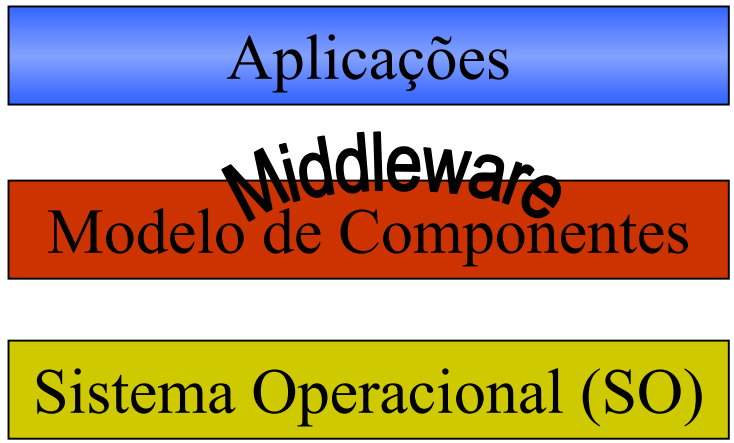
Interfaces	Especificação do componente
Identificação	Nomes únicos globais
Interoperabilidade	Comunicação e troca de dados entre componentes implementados em linguagens diferentes

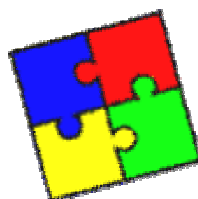


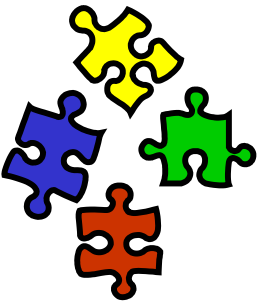


Modelo de Componentes

- A implementação do modelo de componentes:
  - executa no topo de um SO.
  - oferece suporte a execução dos componentes
- ***Middleware***: software que situa-se entre a aplicação e o sistema operacional







Arquitetura & Componentes

Arquitetura da Aplicação

Modelos de Componentes