

Arquitetura de Software

Prof. Jacson Rodrigues Barbosa

Curso: Bacharelado em Engenharia de Software

Aula 03

Transição de Requisitos para Arquitetura
A importância dos Atributos de Qualidade

Referência Bibliográfica

- *Alguns conteúdos apresentados possuem adaptações das aulas do*
 - *Prof. Dr. Valdemar V. Graciano Neto*

Roteiro

- Introdução
- Contextualização no Processo de Desenvolvimento de Software
- Atributos de Qualidade
- Transição Requisitos - Arquitetura
- Proposta de Atividade

Máxima

"Os Atributos de Qualidade (AQ) priorizados vão delinear a arquitetura do software."

Introdução

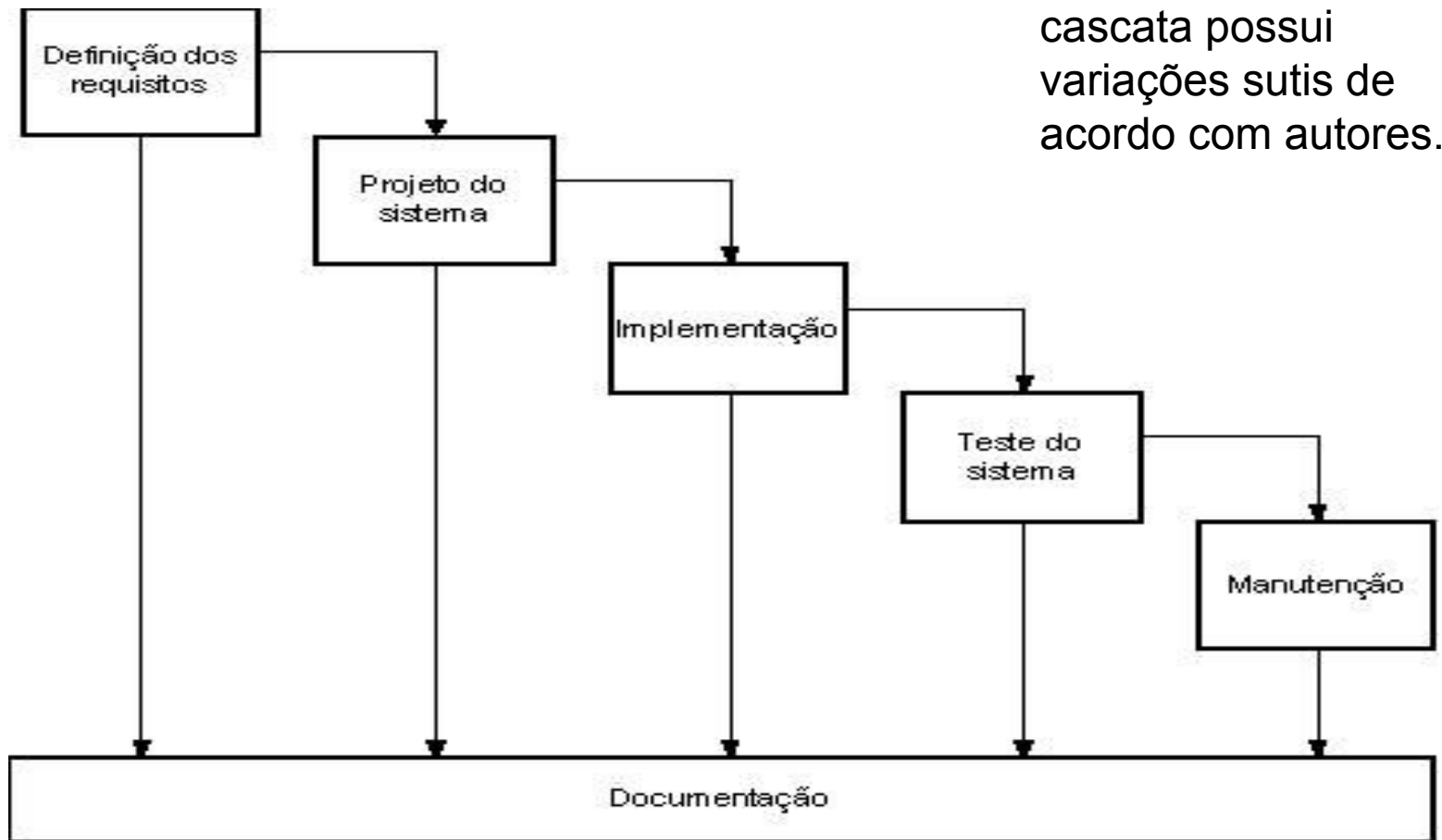
- A definição da arquitetura de software ocorre dentro da Engenharia de Software;
- Por engenharia, entende-se a atividade de²:
 - *"Criar soluções com custo plausível para problemas práticos pela aplicação de conhecimento científico para construir coisas a serviço do ser humano".*

Contextualização

- Logo, projetar a arquitetura de software é uma atividade inerente a isso.
- O **projeto da arquitetura não costuma ser explícito** dentro do processo de desenvolvimento.

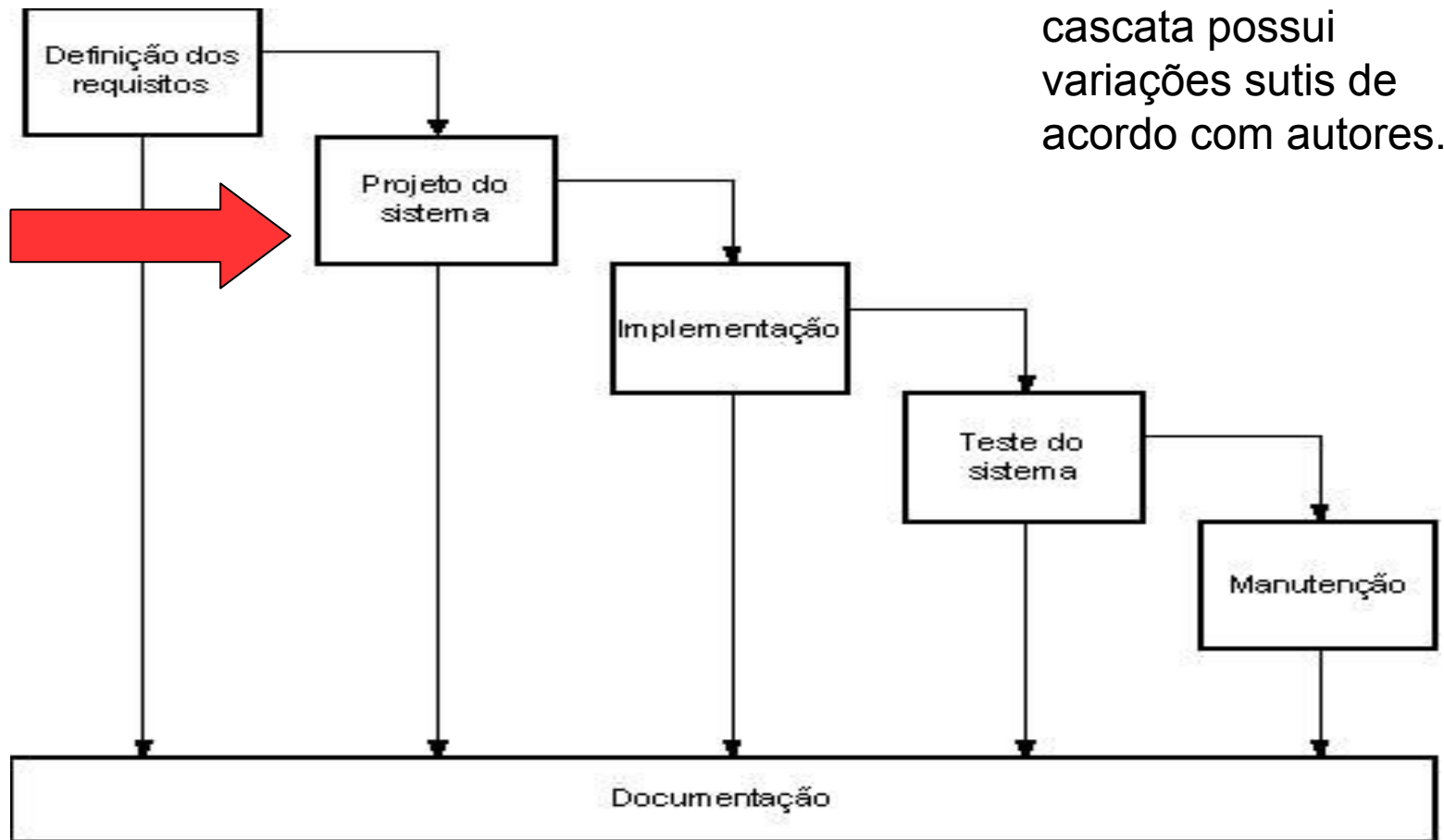
Contextualização

- Localização no processo



Contextualização

- Localização no processo



Contextualização

- A etapa de Projeto (Design) é separada em¹:
 - Projeto Arquitetural; e,
 - Projeto Detalhado.
- O Projeto Arquitetural (que possui um Processo associado, visto posteriormente) é a etapa onde ocorre a definição da arquitetura.

Transição

- Antes do Projeto da Arquitetura, uma etapa inteira (seguindo o modelo cascata) do processo de desenvolvimento é concluída: a **Engenharia de Requisitos**;
- Logo, a concepção da Arquitetura de um Software consiste na **transformação** de um modelo de requisitos utilizado como **insumo**. O produto é um **modelo** completo de Arquitetura de Software para o produto que está sendo desenvolvido.

Transição

- Sobre Requisitos
 - Taxonomia varia entre autores:
 - Requisitos Funcionais (RF) e Não-Funcionais (RNF) – Pressman
 - Requisitos de Usuário (RU) e Requisitos de Sistema (RS)⁴. Neste contexto, RS podem ser RF, RNF, ou Requisitos de Domínio (RD) - Sommerville;
 - Outras:
 - RF, RNF, Regras de Negócio;
 - Necessidades, Requisitos, Regras de Negócio.

Requisito Funcional

- Comportamentos (Funções ou Serviços) ofertados pelo sistema que dão suporte a objetivos, tarefas e atividades [Malan].
- Propósitos do sistema em desenvolvimento
- Responsabilidades a serem cumpridas pelo sistema
- O QUE o sistema deve fazer
- Ex.: Cadastrar usuário, recuperar músicas, diagnosticar doença, aumentar volume, etc.

Atributo de Qualidade

- Uma propriedade testável e mensurável de um sistema que é usada para indicar quão bem o sistema satisfaz as necessidades do usuário
- Qualificação das funcionalidades
- Adjetivo: fácil, seguro, rápido...
- Diz respeito ao COMO
- *-ilities* (mas também tem performance, security, reutilização)
- Qualidade
- Ex.: portabilidade, modificabilidade, manutenibilidade, etc.

Atributo de Qualidade

- ISO 25010



Catálogo de Atributos de Qualidade

- Adequação Funcional
 - Grau com que um produto ou sistema fornece funções que atendem às necessidades declaradas e implícitas quando usadas sob condições especificadas.
 - Métricas?

Catálogo de Atributos de Qualidade

- Eficiência de desempenho
 - Desempenho em relação à quantidade de recursos usados nas condições declaradas.
 - Envolve: tempo, taxas de processamento (quantidade de consultas, por exemplo), limites...
 - Métricas?

Catálogo de Atributos de Qualidade

- Compatibilidade
 - Grau com o qual um produto, sistema ou componente pode trocar informações com outros produtos, sistemas ou componentes e / ou executar suas funções necessárias, enquanto compartilha o mesmo ambiente de hardware ou software;
 - Envolve interoperabilidade
 - Métrica?

Catálogo de Atributos de Qualidade

- Usabilidade
 - Grau com que um produto ou sistema pode ser usado por usuários específicos para atingir metas especificadas com eficácia, eficiência e satisfação em um contexto específico de uso.
 - Métrica?

Catálogo de Atributos de Qualidade

- Confiabilidade (Reliability)
 - Grau com que um sistema executa funções especificadas sob condições determinadas por um período de tempo especificado. Envolve, dentre outras:
 - **Disponibilidade.** Grau em que um sistema, produto ou componente está operacional e acessível quando necessário para uso.
 - **Tolerância a falhas.** Grau para o qual um sistema, produto ou componente opera conforme pretendido, apesar da presença de falhas de hardware ou software.
 - **Recuperabilidade.** Grau com que, no caso de uma interrupção ou falha, o sistema pode recuperar os dados diretamente afetados e restabelecer o estado desejado do sistema.
 - Métricas?

Catálogo de Atributos de Qualidade

- Segurança
 - Grau com que um produto ou sistema protege as informações e os dados para que as pessoas ou outros produtos ou sistemas tenham o grau de acesso aos dados adequado aos seus tipos e níveis de autorização.
 - Métrica?

Catálogo de Atributos de Qualidade

- Outros
 - Manutenibilidade
 - Portabilidade
 - Longevidade
 - Sustentabilidade
 - ...



Reflexão

- Nem sempre um requisito considerado não-funcional o é.
- Ex.: Segurança
 - Em um sistema de informação, é não-funcional (a forma como funcionalidades são entregues);
 - Para um framework de segurança, é funcional.

Transição

- Independentemente:
 - RNF são chamados de Atributos de Qualidade (AQ);
 - AQ são os principais considerados para escolher o padrão arquitetural adotado;
 - RF são agrupados por similaridade e modularizados em partes específicas para aumentar a manutenibilidade;

Transição

- Convenção (taxonomia) para esta discussão:
 - Necessidades (especificações de anseios genuínos do cliente de modo mais amplo e abstrato);
 - RF;
 - Atributos de Qualidade;
 - Regras de Negócio (RN);

Transição

- Logo:
 - RNF auxiliam na escolha do Padrão Arquitetural;
 - RF são agrupados de acordo com o padrão arquitetural escolhido;
 - Para constar, exemplos de Padrões Arquiteturais:
 - Camadas (layers, tiers) (Sistemas de Informação Empresariais);
 - MVC (aplicações web);
 - Pipes-And-Filters (Compiladores, Processos UNIX);
 - Invocação Baseada em Eventos (Swing);
 - Módulos (Sistema Operacional);

Transição

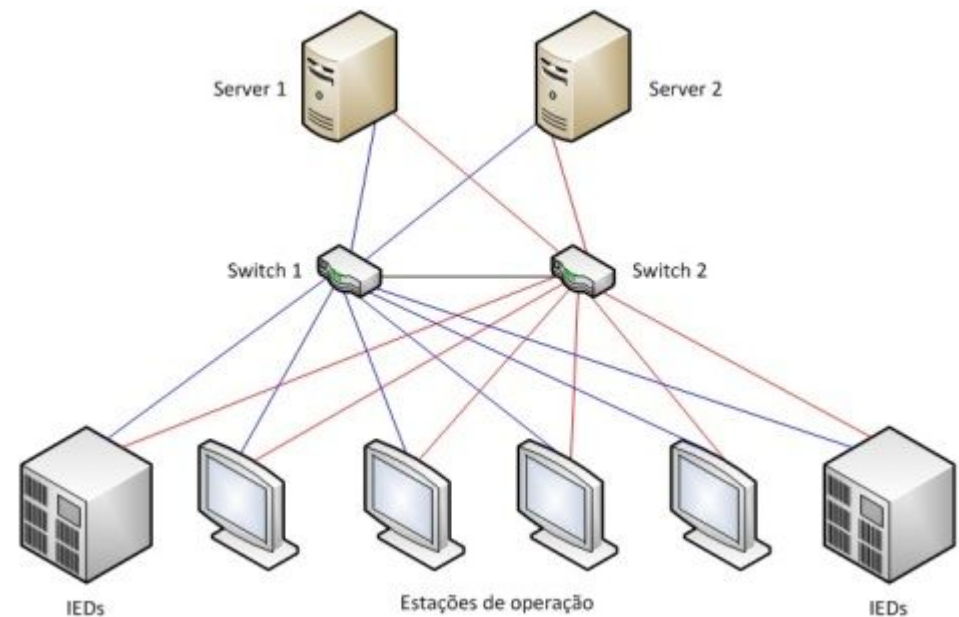
- Observação importante:
 - Em geral, a não ser que haja uma política de modularização⁵, RN (regras de negócio) ficam espalhadas na arquitetura.

Transição

- Uma forma de auxiliar na manutenção da arquitetura e documentar a transição, é através do uso de MATRIZ DE RASTREABILIDADE.
- Rastreabilidade pode ser entre:
 - RF e RF;
 - RF e RNF;
 - RF e Módulos da Arquitetura;
 - RNF e Módulos da Arquitetura;

Transição

- Como AQ são usados como insumo para a definição do padrão arquitetural?
- Exemplos:
 - AQ: Tolerância a Falhas
 - Solução: Arquitetura Distribuída com Redundância



Fonte:

<http://kb.elipse.com.br/pt-br/images/ID4727/Figura%20Rede.jpg>

Transição

- Como AQ são usados como insumo para a definição do padrão arquitetural?
 - Exemplos:
 - AQ: Segurança
 - Solução: A Arquitetura pode ser em camadas, mas deve haver elementos de gerência de segurança como um gerenciador central com verificações espalhadas no arquitetura, ou trechos de login em pontos específicos e manutenção de sessão (no caso de software web);

Transição

- Neste contexto, cabe falar sobre
 - Coesão;
 - Acoplamento.

Transição

- Observação Importante:
 - Em geral, atributos de qualidade devem ter métricas associadas a eles;
 - Exemplo:
 - Tempo de resposta em segundos;
 - Processamento de Consultas em consultas por segundo;
 - Usabilidade em quantidade de passos até chegar em um determinado ponto;
 - Etc.

Transição

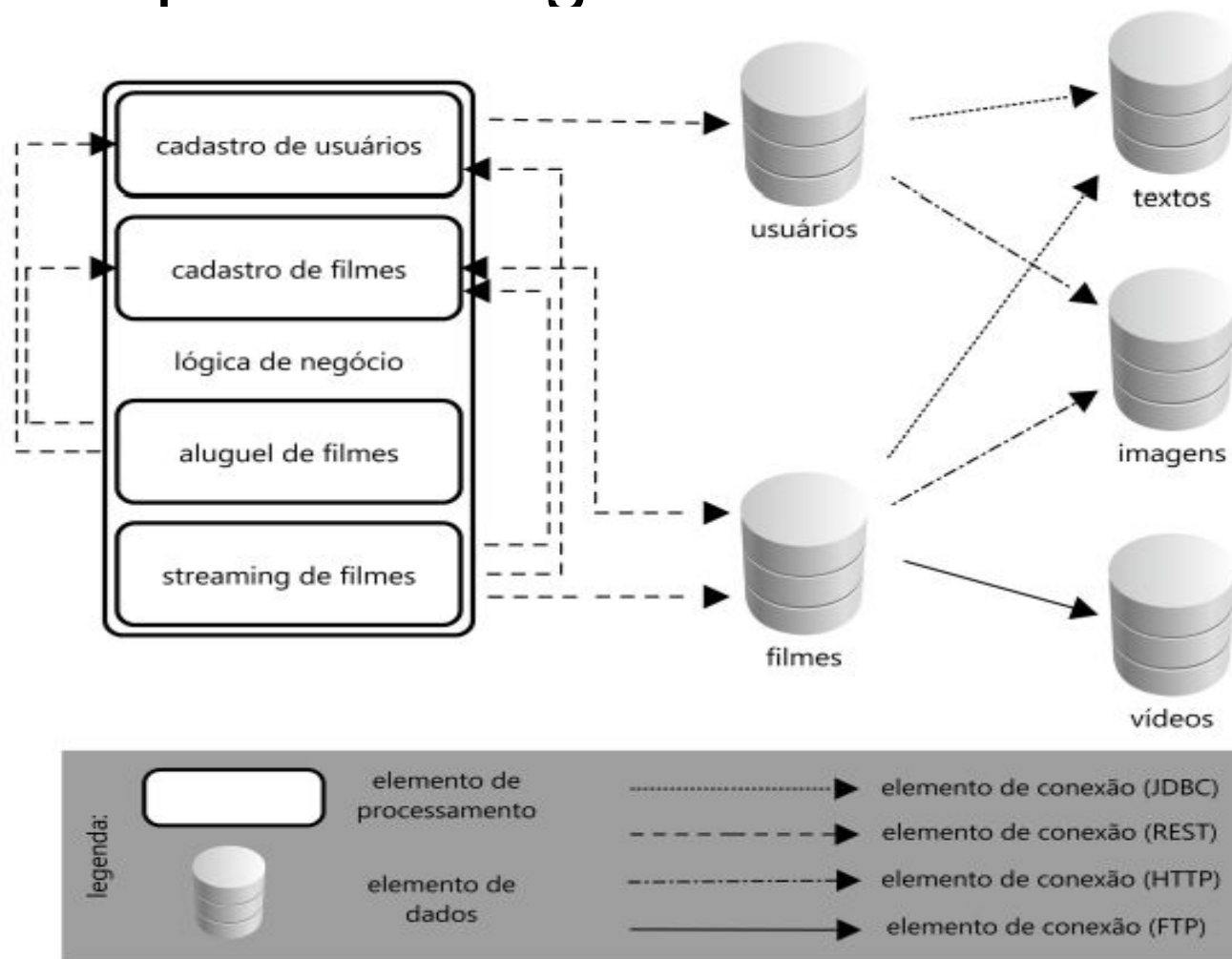
- Segundo Perry e Wolf⁶, uma arquitetura possui¹:
 - Elementos;
 - Organização;
 - Decisões.

Transição

- Segundo Perry e Wolf⁶, uma arquitetura possui¹:
 - Elementos podem ser:
 - Processamento (Camada de Lógica, etc.);
 - Dados (Banco de Dados, Fontes de Dados como Dispositivos de Armazenamento);
 - Conexão (Elementos que ligam como protocolos ou invocações).

Transição

- Exemplo de Arquitetura segundo Perry e Wolf⁶, apresentado por Germoglio¹:



Transição

- O problema é:
 - Qual abordagem utilizar para identificar possíveis padrões arquiteturais a partir dos requisitos existentes e atributos de qualidade que devem ser atendidos?
 - Como agrupar requisitos para delinear a arquitetura do software?
 - Como garantir que apenas um padrão arquitetural será o suficiente para atender à construção adequada do software e não uma arquitetura híbrida?

Transição

- O problema é:
 - Qual abordagem utilizar para identificar possíveis padrões arquiteturais a partir dos requisitos existentes e atributos de qualidade que devem ser atendidos?
 - Como agrupar requisitos para delinear a arquitetura do software?
 - **Resposta: Não há métodos pré-estabelecidos definitivos. Esta é uma decisão que depende de experiência, do domínio, e de algumas heurísticas.**

Transição

- Propostas Iniciais
 - As seções do documento de Requisitos costumam dar boas suspeitas sobre como modularizar (se o documento for bem feito e organizado, é claro);
 - O domínio para o qual o software é projetado ajuda a escolher padrões previamente utilizados de forma bem sucedida:
 - Pipes and Filters para projeto de Compiladores;
 - Módulos para Projeto de SO;
 - Camadas para Projeto de software de SI;
 - MVC para projeto de Aplicações Web;

Transição

- Propostas Iniciais

- Outra dica:

- O livro de Martin Fowler⁸ indica alguns padrões para arquiteturas corporativas.
 - Em geral, ele mostra em que situação cada estilo arquitetural deve ser utilizado.
 - Logo, atender às necessidades isoladas listadas no livro leva à construção de uma arquitetura híbrida de modo natural.

Exercício

- Fazer um mapa conceitual que resume as informações da aula de hoje.

Referências

1. Germoglio, Guilherme. Arquitetura de Software. Connexions, Houston, Texas. 2010. 254 p.
2. Shaw, Mary; Garlan, David. Software Architecture: Perspectives on an Emerging Discipline. 1996. Prentice-Hall. 242 p.
3. Silveira, Paulo; et. al. Introdução à Arquitetura e Design de Software: Uma visão sobre a plataforma Java. 2012. Editora Campus-Elsevier. 268 p.

Referências

4. Sommerville, Ian. Engenharia de Software. 8a. Edição. Addison-Wesley. 2007.
5. Boff, Glauber; de Oliveira, Juliano Lopes. Modeling, Implementation and Management of Business Rules in Information Systems. INFOCOMP; 2:1. pp 17-28. 2010.
6. Perry, Dewayne; Wolf, Alexander. Foundations for the Study of Software Architecture. ACM SIGSOFT, Oct. 1992.
7. Fowler, Martin. Padrões de Arquitetura de Aplicações Corporativas. Bookman, 2008. 493 p.