Engenharia de Software

Aula 09 – Projeto de Software: Arquiteturas

Profa. Rafaella Nascimento
rafaellalnascimento@fac.pe.senac.br

Objetivos

- Apresentar a importância do projeto de arquitetura de software;
- Apresentar os padrões de arquitetura e modelos de referência.

Arquitetura

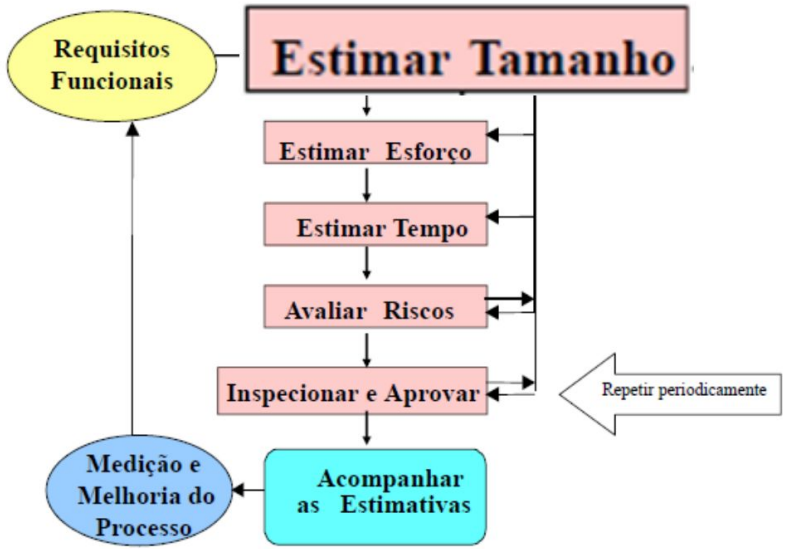
Fases genéricas do projeto de software:

1. Especificação de software. A funcionalidade do software e as restrições a seu funcionamento devem ser definidas.

2. Projeto e implementação de software. O software deve ser produzido para atender às especificações.

3. Validação de software. O software deve ser validado para garantir que atenda às demandas do cliente.

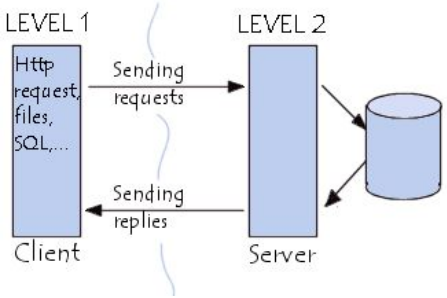
4. Evolução de software. O software deve evoluir para atender às necessidades de mudança dos clientes.



Arquitetura

- Toda obra da humanidade apresenta um projeto arquitetural.
- O projeto arquitetural precede a etapa de construção da obra.
- O projeto arquitetural determina as partes de uma construção e como estas devem interagir.
- A arquitetura garante a unidade da obra, ou seja, a consistência entre as suas partes.

Arquitetura



Categorias arquiteturais	Representações de projeto	Restrições
Arquitetura Clássica	Modelos, desenhos, planos, elevações e perspectivas	Padrão de circulação, acústica, iluminação e ventilação
Arquitetura de Software	Modelos para diferentes papéis, múltiplas visões	Desempenho, confiabilidade, escalonamento e manutenibilidade

Projeto da Arquitetura de Software

Em ES, o Projeto da Arquitetura do Software é …

- Arranjo do sistema para fazer corresponder os requisitos – tanto funcionais quanto não funcionais - aos subsistemas e componentes.

Arquitetura = {Elementos, Organização, Decisões}

É um conjunto de elementos arquiteturais (de dados, de processamento, de conexão) que possuem alguma organização. Os elementos e sua organização são definidos por decisões tomadas para satisfazer objetivos e restrições do projeto.

Arquitetura de Software

Além da escolha dos algoritmos e estruturas de dados, a arquitetura envolve:

- Decisões sobre as estruturas que formarão o sistema;
- Controle;
- Protocolos de comunicação;
- Sincronização e acesso a dados;
- Atribuição de funcionalidade a elementos do sistema;
- Distribuição física dos elementos escalabilidade e desempenho entre outros atributos de qualidade e
- Seleção de alternativas de projeto.

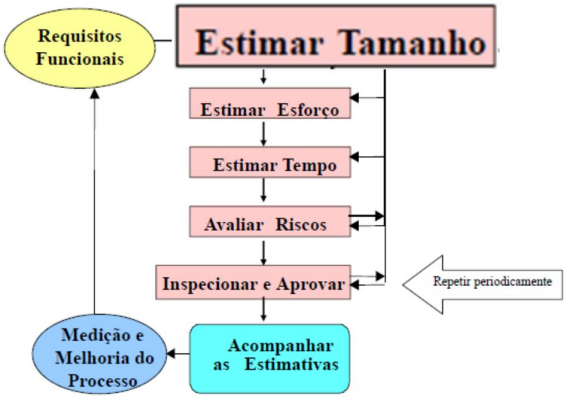
A arquitetura de software é usada para designar processo e produto.

Arquitetura de Software

O Processo de Arquitetura de Software

- Elaboração do modelo de negócio – envolve analisar custo, tempo de desenvolvimento, restrições de mercado, interfaces com outros sistemas, etc
- Entendimento dos requisitos: levantamento de requisitos e modelo do domínio.
- Criação ou seleção de uma arquitetura: identificação dos componentes e suas interações, das dependências e tecnologias que apoiam a implementação.
- Representação da arquitetura e divulgação: para permitir aos desenvolvedores e testadores o entendimento da arquitetura
- Implementação da arquitetura, seguindo seus protocolos e estruturas.
- Análise e avaliação: verificar a adequação da arquitetura, registrando impactos.

Arquitetura de Software



A definição da arquitetura

A fase de engenharia de requisitos é subsídio para a definição da arquitetura da qual se obtém:

- processo de negócio modelado
- planejamento estratégico das versões
- requisitos de cada versão, etc.

A definição está baseada na escolha de alternativas mais adequadas ao domínio da aplicação (reutilizar estratégias validadas, frameworks, estilos, padrões e linguagens).

Arquitetura de Software

Padrão

- É um template (formulário) de solução para um problema recorrente que seja comprovadamente útil em um determinado contexto.
- Um padrão de software é instanciado através da vinculação de valores a seus parâmetros.
- Os padrões podem existir em várias escalas e níveis de abstração.

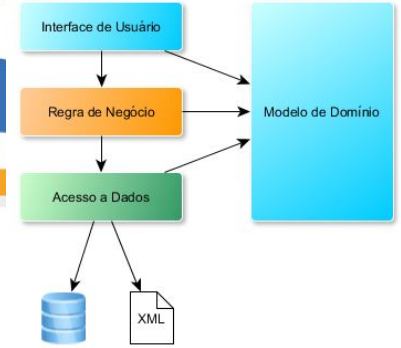
padrões de arquitetura, padrões de análise, padrões de projeto, padrões de teste e idiomas ou padrões de implementação.

Arquitetura de Software

Um padrão expressa uma solução reutilizável descrita através de três partes: um contexto, um problema e uma solução.

- **Contexto:** estende o problema a ser solucionado, apresentando situações de ocorrência desses problemas.
- **Problema:** determinado por um sistema de forças, onde estas forças estabelecem os aspectos do problema que devem ser considerados.
- **Solução:** mostra como resolver o problema recorrente e como balancear as forças associadas a ele.

Arquitetura de Software



Nome do Padrão: Camadas

Contexto Um sistema grande que requer decomposição.

Problema Um sistema que deve resolver as questões em diferentes níveis de abstração. Por exemplo: as questões de controle de hardware, as questões de serviços comuns e as questões específicas de domínio. Seria extremamente indesejável escrever componentes verticais que lidem com essas questões em todos os níveis. Uma mesma questão deveria ser resolvida (possivelmente de maneira inconsistente) várias vezes em diferentes componentes.

Solução Estruture os sistemas em grupos de componentes que formem camadas umas sobre as outras. Faça com que as camadas superiores utilizem os serviços somente das camadas abaixo (nunca das camadas acima). Tente não usar serviços que não sejam os da camada diretamente abaixo (não pule camadas, a menos que as camadas intermediárias somente adicionem componentes de acesso).

Arquitetura de Software

Estilos Arquiteturais e Escolhas de Projeto

- Um estilo arquitetural representa um conjunto de escolhas de projeto – Conjunto de características comuns a diversos sistemas nos quais as mesmas escolhas foram feitas
- Padrões arquiteturais – Um sistema aderente a determinado estilo “ganha" as características a ele inerentes
- Estilos podem ser usados para descrever uma determinada arquitetura – Foco nas soluções de projeto e não em sua documentação

Arquitetura de Software

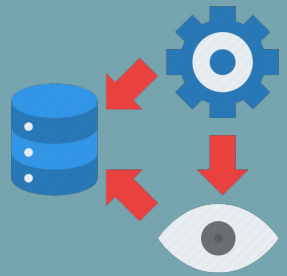
- A arquitetura de um sistema pode aderir a um ou mais estilos arquiteturais.
- Um estilo define os tipos de elementos que podem aparecer em uma arquitetura e as regras que regem a interconexão entre estes elementos.
- Esses estilos podem simplificar o problema de definição de arquiteturas de sistemas.
- A maioria dos sistemas de grande porte adere a vários estilos
- Estilos arquiteturais = “modelos arquiteturais”

Exemplos de Estilos Arquiteturais:

Cliente-Servidor • Camadas • MVC • Repositório

Modelo-Visão-Controlador

(model-view-controller)



Modelo-Visão-Controlador

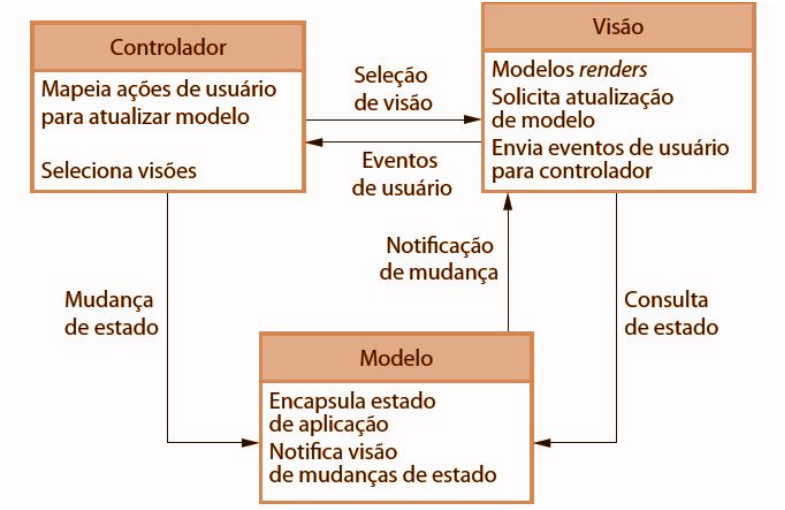
Descrição:

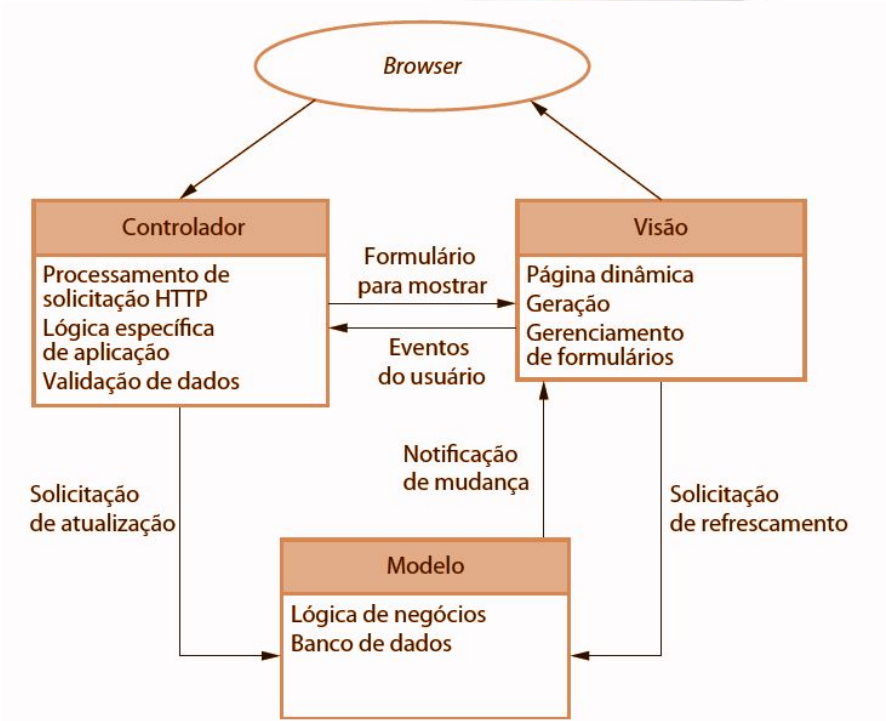
O sistema é estruturado em três componentes lógicos que interagem entre si:

1. O componente Modelo gerencia o sistema de dados e as operações associadas a esses dados.
2. O componente Visão define e gerencia como os dados são apresentados ao usuário.
3. O componente Controlador gerencia a interação do usuário (por exemplo, teclas, cliques do mouse etc.) e passa essas interações para a Visão e o Modelo.

É usado quando existem várias maneiras de se visualizar e interagir com dados e quando são desconhecidos os futuros requisitos de interação e apresentação de dados.

Modelo-Visão-Controlador





Exemplo: Arquitetura de aplicações Web usando o padrão MVC

Arquitetura em Camadas

Camadas

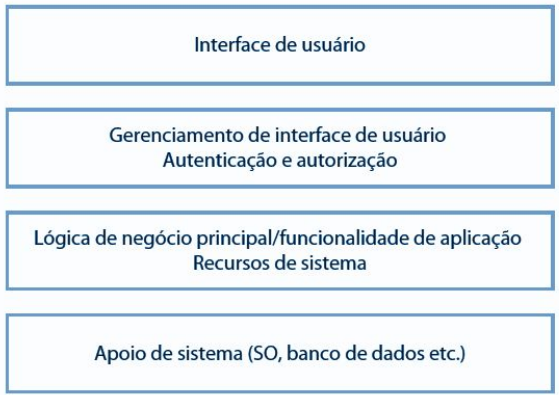
A funcionalidade do sistema é organizada em camadas separadas, e cada camada só depende dos recursos e serviços oferecidos pela camada imediatamente abaixo dela.

Cada camada é responsável por um conjunto relacionado de serviços.

Esse tipo de abordagem apóia o desenvolvimento incremental de sistemas.

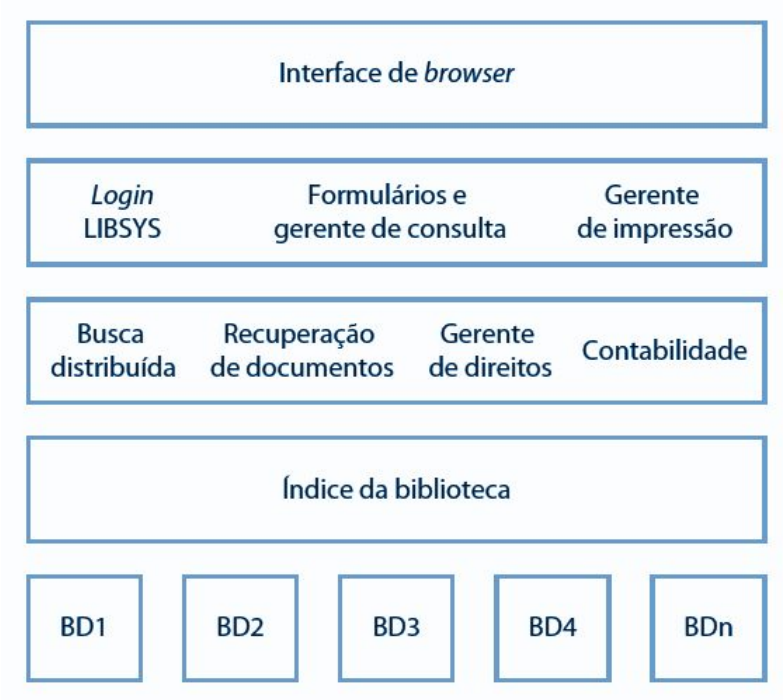
É usado:

- Na construção de novos recursos em cima de sistemas existentes;
- Quando o desenvolvimento está espalhado por várias equipes, com a responsabilidade de cada equipe em uma camada de funcionalidade.

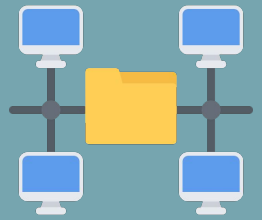


Camadas

Exemplo: sistema de biblioteca



Arquitetura com Repositórios



Repositório

Neste exemplo, todos os dados em um sistema são gerenciados em um repositório central, acessível a todos os componentes do sistema.

Os componentes não interagem diretamente, apenas por meio do repositório.

É usado:

- Quando tem um sistema no qual grandes volumes de informações são gerados e precisam ser armazenados por um longo tempo.
- Em sistemas dirigidos a dados, nos quais a inclusão dos dados no repositório dispara uma ação ou ferramenta.
- Exemplos: sistemas de comando e controle, sistemas de informações gerenciais (SIG), ambientes interativos de desenvolvimento de software

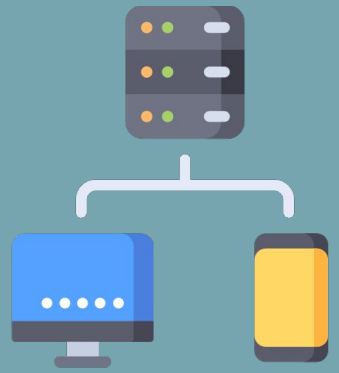
Descreve como um conjunto de componentes que interagem podem compartilhar dados.

Repositório

Cada ferramenta de software gera informações que ficam disponíveis para uso por outras ferramentas.

Exemplo: repositório para um IDE

Cliente-Servidor



Cliente-Servidor

O modelo de arquitetura cliente-servidor organiza o sistema como um conjunto de servidores e clientes associados que acessam e usam os serviços.

Os principais componentes desse modelo são:

1. Conjunto de servidores que oferecem serviços para outros subsistemas.
2. Conjunto de clientes que solicita os serviços oferecidos pelos servidores.
3. Uma rede que permite aos clientes acessarem esses serviços.

Características:

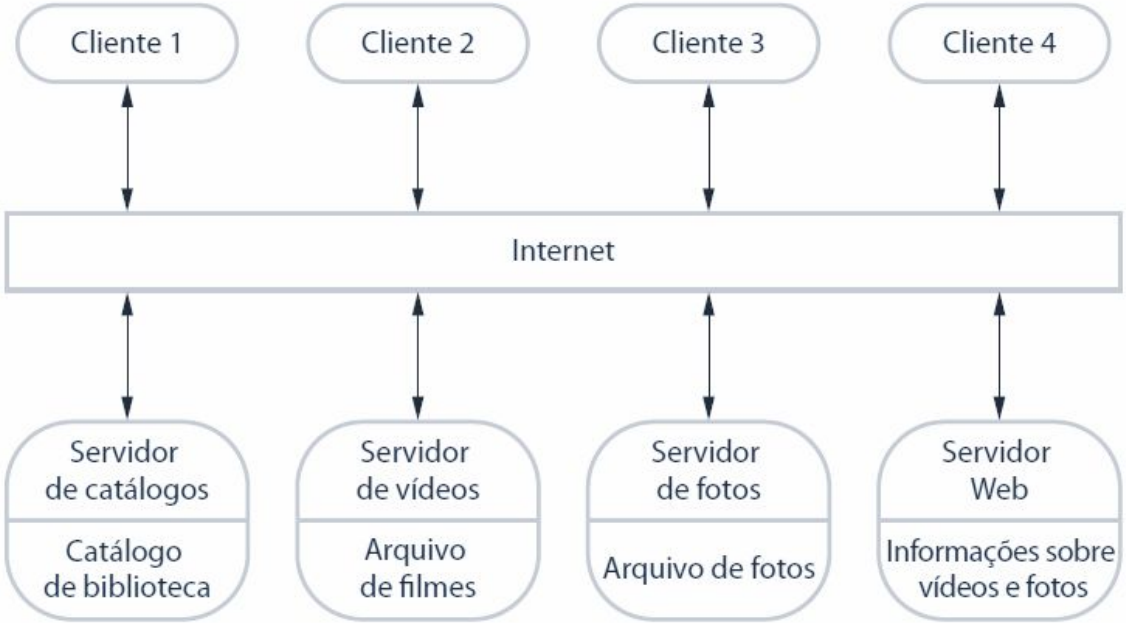
- Separação e Independência: Serviços e Servidores podem ser alterados sem afetar o sistema.
- Clientes sabem sobre os serviços e servidores disponíveis, mas servidores não conhecem clientes.

Usado em:

- Sistemas distribuídos em tempo de execução.
- É usado quando os dados em um banco de dados compartilhado precisam ser acessados a partir de uma série de locais.
- Quando os servidores podem ser replicados.

Cliente-Servidor

Exemplo: Biblioteca de filmes



Dúvidas?

Profa. Rafaella Nascimento
rafaellalnascimento@fac.pe.senac.br