

Técnico em Desenvolvimento de Sistemas

Projeto de Software I

Juliano Lucas Gonçalves

juliano.goncalves@ifsc.edu.br / julianolg@gmail.com

Projeto de Software I

Agenda

Projeto de Arquitetura

- Conceito
- Objetivos
- Benefícios
- Visualizações Arquiteturais
- Modelos Arquiteturais

Projeto de Software I

O que é arquitetura de software?

Uma arquitetura é o conjunto de **decisões significativas** sobre a **organização de um sistema de software**, a **seleção de elementos estruturais e suas interfaces**, juntamente com o comportamento especificado nas colaborações entre estes elementos, a composição destes elementos em subsistemas progressivamente maiores.

Projeto de Software I

Um sistema típico de software é perecível, resultado de:

- incertezas
 - no comportamento do sistema
 - nas próximas release
- baixa qualidade
 - difícil rastrear falhas
 - difícil consertar bugs
- difícil alterar
 - duro atender às mudanças
 - custa mais, leva mais tempo
- Baixo reuso
 - difícil isolar blocos para reuso
 - blocos são muito específicos (orientados)
- problemas éticos
- perda de market share
 - o concorrente é melhor
 - não há retorno

Projeto de Software I

Arquitetura de Software tem relação com as tomadas de decisões durante o processo de desenvolvimento assim como a estruturação dos elementos que compõe o sistema:

- Decisões
- Organização fundamental
- Elementos de software
- Relacionamento entre elementos
- Relacionamento com o ambiente
- Princípios que guiam o desenho e evolução
- Pode determinar o sucesso ou fracasso

Projeto de Software I

Uma série de causas individualmente ou em conjunto pode fracassar um projeto de desenvolvimento de software. Diversos riscos e restrições associados ao processo aumentam a probabilidade de cancelamento. Falta de controle, escopo volátil, prazos exorbitantes, falta de recursos e entre outros vários motivos.

A figura abaixo mostra uma pesquisa realizada com 345 organizações brasileiras. O objetivo da pesquisa foi identificar as principais causas de fracasso de projetos de desenvolvimento de novas aplicações.



Projeto de Software I

Objetivos da Arquitetura:

Reduzir os riscos associados ao projeto e das atividades que cercam o Processo de Desenvolvimento de Sistemas. Dentre os diversos benefícios adquiridos pelo processo de Arquitetura, destacam-se:

- › Redução de riscos associados ao negócio
- › Redução de riscos de desenvolvimento e manutenção
- › Alinhamento de expectativas dos stakeholders
- › Construção de aplicações flexíveis e de qualidade
- › Geração de valor aos negócios do cliente
- › Apoio ao Arquiteto na tomada de decisões técnicas

Projeto de Software I

Quais os benefícios ?

- **Aumento do desempenho**
- **Garantia de escalabilidade**
- **Personalização**

Projeto de Software I

Quais os benefícios ?

Aumento do desempenho

- **Softwares mal estruturados enfrentam um problema frequente com relação ao desempenho. Se houver uma demanda de aumento de capacidade não prevista, ele travará. Isso pode prejudicar diversos processos, além de demandar manutenção no sistema.**
- **Por outro lado, um software bem arquitetado tem maior capacidade de aumentar seu desempenho, dando conta do aumento de escala e do volume de dados gerados nessa operação.**

Projeto de Software I

Quais os benefícios ?

Garantia de escalabilidade

- A escalabilidade é um dos principais objetivos de muitas empresas, hoje em dia, que desenvolvem serviços baseados em software e que podem ser implementados não só na cidade em que estão situadas, mas em todo o mundo.

Projeto de Software I

Quais os benefícios ?

Personalização

- Outro benefício que pode ser obtido com um software bem arquitetado é a personalização. Essa é uma solução ideal em qualquer caso, afinal, cada empresa tem suas particularidades e precisa de um software que seja feito sob medida para suas necessidades.

Projeto de Software I

Visualizações Arquiteturais

- Cada visualização arquitetural trata de um conjunto específico de interesses, específicos dos envolvidos no processo de desenvolvimento: usuários, designers, gerentes, engenheiros de sistema, mantenedores e assim por diante.

Projeto de Software I

Conjunto de visualizações arquiteturais

- **A Visualização de Casos de Uso**, que contém casos de uso e cenários que englobam comportamento arquiteturalmente significativo, classes ou riscos técnicos. Ela é um subconjunto do Modelo de Caso de Uso.
- **A Visualização Lógica**, que contém as classes de design mais importantes e sua organização em pacotes e subsistemas e a organização desses pacotes e subsistemas em camadas. Também contém algumas realizações de casos de uso. É um subconjunto do Modelo de Design.
- **A Visualização de Implementação**, que contém uma visão geral do Modelo de Implementação e sua organização em termos de módulos em pacotes e camadas. A alocação de pacotes e classes (da Visão Lógica) nos pacotes e módulos da Visão de Implementação também é descrita. É um subconjunto do Modelo de Implementação.
- **A Visualização do Processo**, que contém a descrição das tarefas (processo e encadeamentos) envolvidas, suas interações e configurações, e a alocação de objetos e classes de design para tarefas.

Projeto de Software I

Conjunto de visualizações arquiteturais

O caminho da Arquitetura de Software
para a estruturação de uma aplicação



Projeto de Software I

Modelos Arquiteturais de Software

Padrões que são mais usados para a criação da solução do software, que também são conhecidos como estilo de arquitetura:

- Arquitetura cliente-servidor (Client-server pattern)
- Arquitetura em camadas (Layered pattern)
- Arquitetura MVC (Model-view-controller pattern)

Projeto de Software I

Modelos Arquiteturais de Software

- Arquitetura cliente-servidor (Client-server pattern)

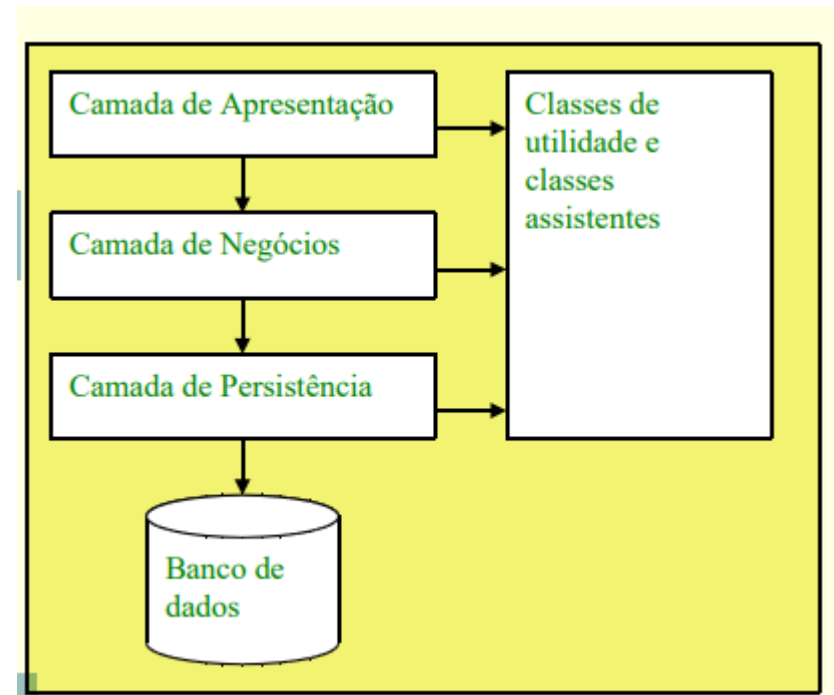
Estilo organizado em serviços combinando dados do cliente e do servidor. Para isso, é primordial que o cliente disponibilize uma rede de acesso às informações. Este cenário é um dos mais conhecidos na rotina das pessoas, já que podem ser aplicativos bancários e e-mail.

Projeto de Software I

Modelos Arquiteturais de Software

Arquitetura em camadas (Layered pattern)

visa a criação de aplicativos modulares, de forma que a camada mais alta se comunica com a camada mais baixa e assim por diante, fazendo com que uma camada seja dependente apenas da camada imediatamente abaixo.



Projeto de Software I

Arquitetura em camadas (Layered pattern)

- **Camada de Apresentação:** Lógica de interface do usuário (GUI). O código responsável pela apresentação e controle da página e tela de navegação forma a camada de apresentação;
- **Camada de Negócios:** Código referente a implementação de regras de negócio ou requisitos do sistema;
- **Camada de persistência:** Responsável por armazenamento e recuperação dos dados quando solicitado. Objetivo é o de garantir uma independência da fonte de dados (arquivos, bancos de dados, etc) e ao mesmo tempo manter as informações entre diferentes sessões de uso.

Projeto de Software I

Arquitetura em camadas (Layered pattern)

- **Banco de dados:** O BD existe fora da aplicação Java, é a atual representação persistente do estado do sistema.
- **Assistentes e Classes de utilidade:** São classes necessária para o funcionamento ou mesmo o complemento de uma aplicação ou parte dela, como por exemplo o Exception para tratamento de erros.

Projeto de Software I

Modelos Arquiteturais de Software

- Arquitetura MVC (Model-view-controller pattern)
 - Padrão projeto para o desenvolvimento de aplicações,
 - A implementação de aplicações usando este padrão são feitas com recurso a frameworks, apesar de não ser obrigatória a utilização de uma para seguir o padrão.

Projeto de Software I

- Arquitetura MVC (Model-view-controller pattern)
 - **Objetivo**
 - Isolar mudanças na GUI, evitando que estas mudanças acarretem em mudanças na Camada de Negócios da Aplicação
 - **Vantagens**
 - Facilita a manutenção
 - Facilita o desenvolvimento por times multi-disciplinares:

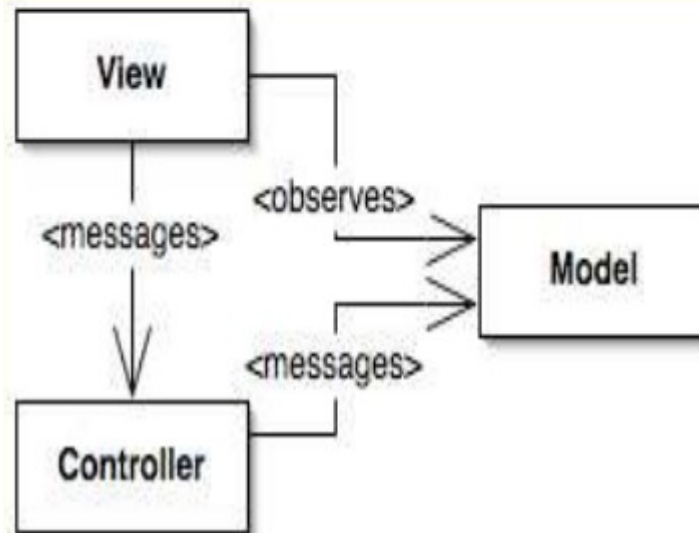
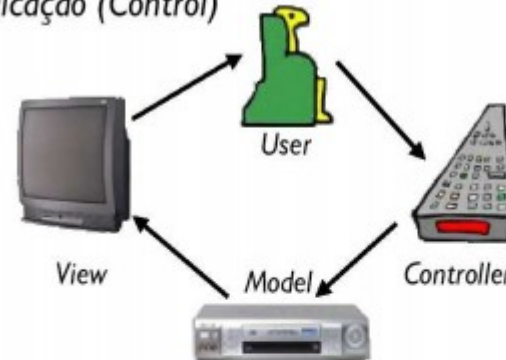
Projeto de Software I

- Arquitetura MVC (Model-view-controller pattern)
- **Objetivo**
 - Isolar mudanças na GUI, evitando que estas mudanças acarretem em mudanças na Camada de Negócios da Aplicação
- **Vantagens**
 - Facilita a manutenção
 - Facilita o desenvolvimento por times multi-disciplinares:

Projeto de Software I

Arquitetura MVC (Model-view-controller pattern)

Técnica para separar dados ou lógica de negócios (Model) da interface do usuário (View) e do fluxo da aplicação (Control)



Projeto de Software I

Arquitetura MVC (Model-view-controller pattern)

Model:

Define as regras de acesso e manipulação dos dados Armazenados em bases de dados ou ficheiros, mas nada indica que sirva só para alojamento persistente dos dados. Todas as regras relacionadas com tratamento, obtenção e validação dos dados devem ser implementados nesta camada.

View:

Responsável por gerar a forma como a resposta será apresentada, página web, formulário, relatório, etc...

Controller:

Responsável por responder aos pedidos por parte do utilizador. Sempre que um utilizador faz um pedido ao servidor esta camada é a primeira a ser executada.

Projeto de Software I

Considerações Finais

Uma arquitetura Boa, Correta e de Sucesso:

- Não aparece “do nada”;
- É necessário:
 - Planejar (tempo!)
 - Documentar e seguir avaliando (não é algo estático)
 - As vezes, joga-se fora e recomeça do zero