

Gerenciamento de Projetos de Software

Prof. Marina Girolimetto

Créditos Prof. Raquel Pegoraro



Questões levantadas pelo cliente antes de iniciar o projeto



Quanto vai custar?



Quanto tempo vai levar?

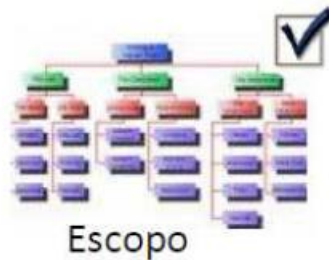


*Minhas necessidades
Serão atendidas?*

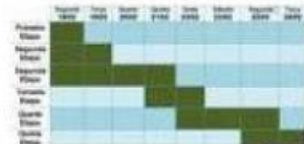
Vou planejar o projeto!



Engenheiro de Software



Escopo



Tempo



Custos



Equipe



Riscos



Questões levantadas pelo cliente antes de iniciar o projeto



Quanto vai custar?



Quanto tempo vai levar?



*Minhas necessidades
Serão atendidas?*

Um cliente quer saber se seu fornecedor é capaz de realizar esse trabalho, quanto o projeto custará e qual será a sua duração.



Engenheiro de Software



Custos



Equipe



Riscos



Questões levantadas pelo cliente antes de iniciar o projeto

Para responder a essas perguntas, é necessário definir o escopo do projeto, através de um levantamento preliminar de requisitos, realizar estimativas, levantar riscos, locar recursos e definir um cronograma de execução.

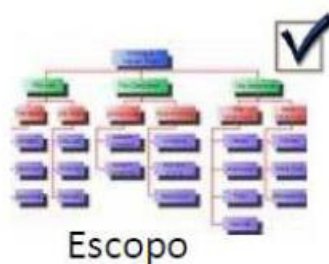


*Minhas necessidades
Serão atendidas?*

Vou planejar o projeto!



Engenheiro de Software



Escopo



Tempo



Custos



Equipe



Riscos



Questões levantadas pelo cliente antes de iniciar o projeto



Quanto vai custar?



Quanto tempo vai levar?



*Minhas necessidades
Serão atendidas?*

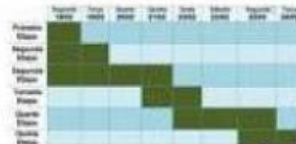
Todas essas informações são registradas em um documento, chamado Plano de Projeto.



Engenheiro de Software



Escopo



Tempo



Custos



Equipe



Riscos



Questões levantadas pelo cliente antes de iniciar o projeto



Quanto vai custar?



Quanto tempo vai levar?

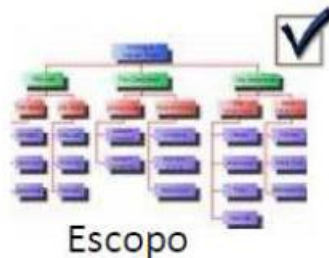


O gerenciamento de projetos tenta adquirir controle sobre três variáveis principais: Tempo; Custo; Escopo.

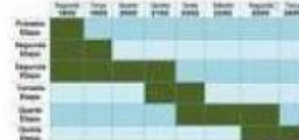
Vou planejar o projeto!



Engenheiro de Software



Escopo



Tempo



Custos



Equipe



Riscos



Stakeholders

- **Indivíduos e organizações envolvidos no projeto**, ou que serão afetados positivamente ou negativamente pelo resultado final.
- Devem ser identificados e gerenciados, pois são elementos chaves no projeto.
- Exemplos: **Cliente; Gerente de Projeto; Organização; Patrocinador; Chefe; Time.**



Arquitetura de Software: Modelo MVC

Prof. Marina Girolimetto

Créditos Prof. Raquel Pegoraro



Modelo MVC

- Padrões Arquiteturais: expressam um esquema de organização estrutural fundamental para sistemas de software. Exemplo: MVC.



Modelo MVC

- Organiza uma aplicação em três camadas para aumentar a flexibilidade;
- Separa as regras de negócio da interface;
- Possibilita a reutilização de código (principalmente em projetos diferentes);
- Facilita a compreensão e manutenção das aplicações;
- Facilita a criação de múltiplas interfaces do sistema (ex: temas e templates, sem precisar modificar o código), portanto passou a ser adotado por diversos *frameworks*.



Modelo MVC

- Determina as responsabilidades de cada camada e como essas partes devem se relacionar.



Modelo MVC

- A ideia por trás do padrão de arquitetura Model-View-Controller é separar claramente a responsabilidade de cada parte do sistema.

Manipulação dos dados

→ **M**odel

**Apresenta os dados para
o usuário**

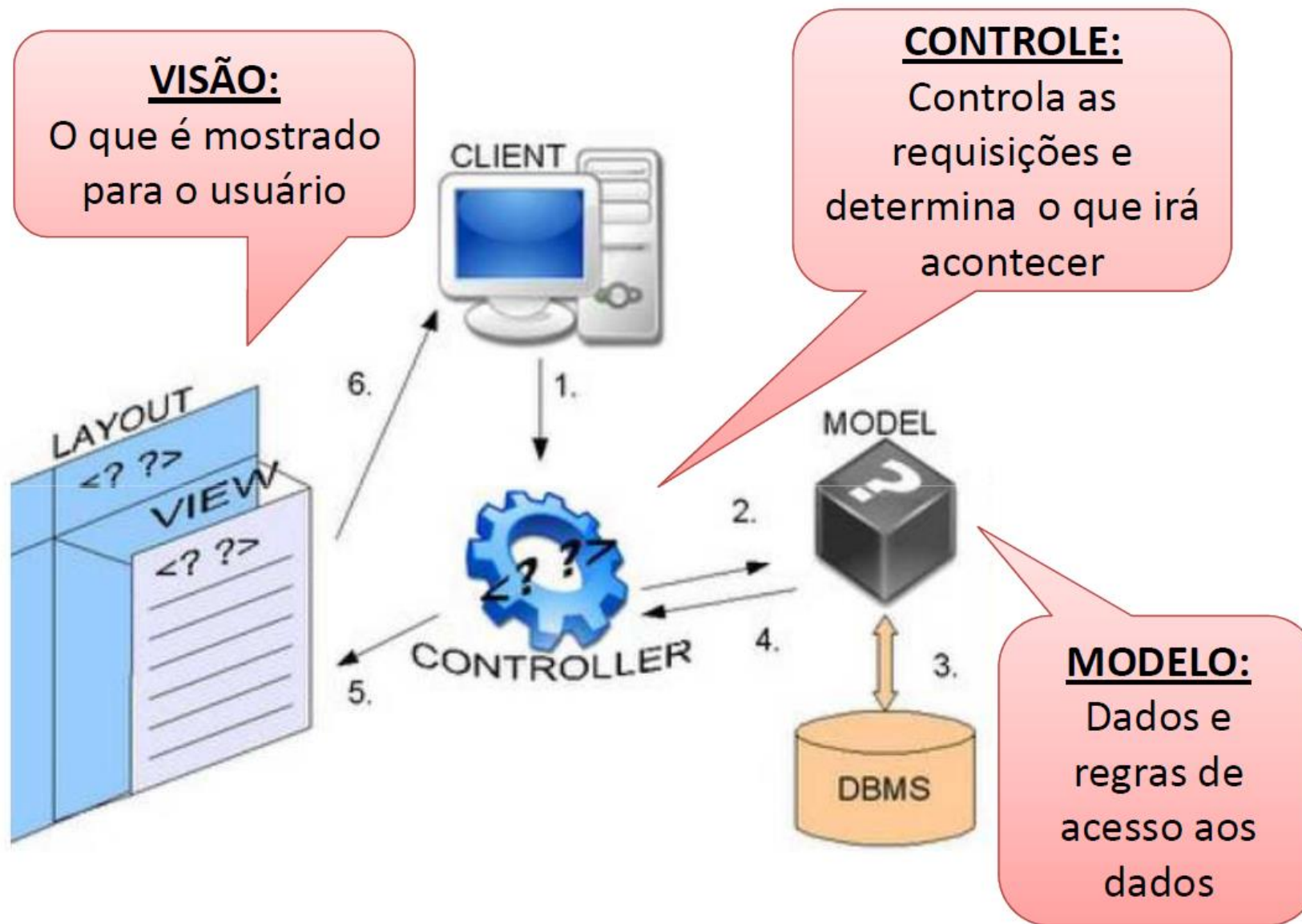
→ **V**iew

**Recebe e responde as
requisições**

→ **C**ontroller



Modelo MVC



Modelo MVC - Controle

- Comanda o fluxo de apresentação das informações fazendo a intermediação entre as camadas de visão e de modelo.
- É responsável por organizar as requisições feitas pelo usuário.
- A sua função principal é chamar e coordenar os recursos e objetos necessários para executar a ação requisitada pelo usuário.
- Normalmente o Controller irá chamar o Model apropriado, receber os dados dele e enviar para a View apropriada.



Modelo MVC - Modelo

- Os objetos desta camada contêm os dados do negócio e as regras do negócio que ditam como acessar e modificar os dados de maneira consistente.
- Representa os dados e as regras que se aplicam a esses dados.
- O Model dá ao Controller uma representação dos dados que o usuário requisitou (uma mensagem, uma lista de livros, uma foto e etc.). Este modelo de dados será sempre o mesmo, não importa como ele será apresentado para o usuário.



Modelo MVC - Modelo

- É por isso que podemos escolher qualquer View disponível para apresentar os dados.
- O View permite diferentes formas de apresentação para os dados recebidos do Model.



Modelo MVC - Visão

- É responsável pela interação com o usuário e por apresentar as diversas visões dos dados do negócio.
- Não se preocupa em como os dados foram obtidos, apenas em apresentá-los.
- São como templates onde os dados preenchem as lacunas. Podem existir várias Views diferentes e é o Controller que decide qual será usada.
- O Controller deve ser estruturado como grandes organizadores, que recebem as requisições, decidem quais ações serão tomadas, quais Models serão usados e quais Views serão apresentadas.



Modelo MVC

- Desta forma é possível alterar uma camada sem afetar as outras.



Modelo MVC - Vantagens

- Como o modelo MVC gerencia múltiplos visualizadores usando o mesmo modelo é fácil manter, testar e atualizar sistemas múltiplos;
- É possível ter desenvolvimento em paralelo para o modelo, visualizador e controle pois são independentes.



Modelo MVC - Desvantagens

- Compreender o conceito;
- Requer uma quantidade maior de tempo para analisar e modelar o sistema;
- Requer pessoal especializado.



Modelo MVC - Frameworks

- Framework é uma abstração que une códigos comuns entre vários projetos de software provendo uma funcionalidade genérica.
- Frameworks que usam MVC ou seu conceito:
 - Rails e Merb (para Ruby);
 - Spring, Struts (para Java);
 - Laravel, CodeIgniter, CakePhp, Symfony (para PHP);
 - React, Backbone, Angular (para Javascript);
 - Asp.Net MVC (para C Sharp);
 - Django, CherryPy (para Python).

