

Arquitetura MVC (Model-View-Controller)

Projeto de Software

Profª Aline Brito



Arquitetura MVC (Model-View-Controller)

Arquitetura MVC

Foi proposto no final da década de 70 e, em seguida, usado na implementação de Smalltalk-80, que é considerada uma das primeiras linguagens orientadas a objetos

Pioneiros no uso de interfaces gráficas, com janelas, botões, scroll bars, mouse

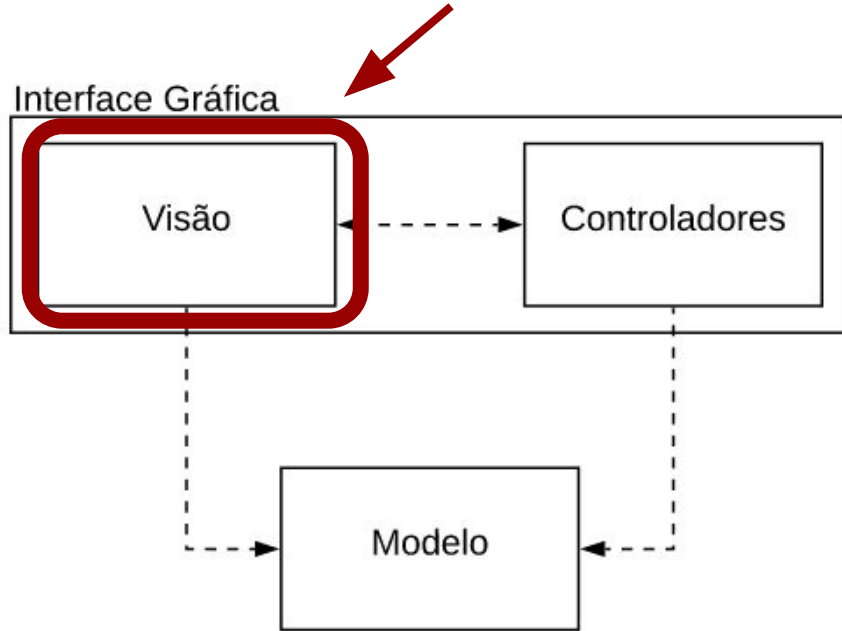
Arquitetura MVC

MVC foi o padrão arquitetural escolhido pelos projetistas de Smalltalk para implementação de interfaces gráficas

Arquitetura MVC

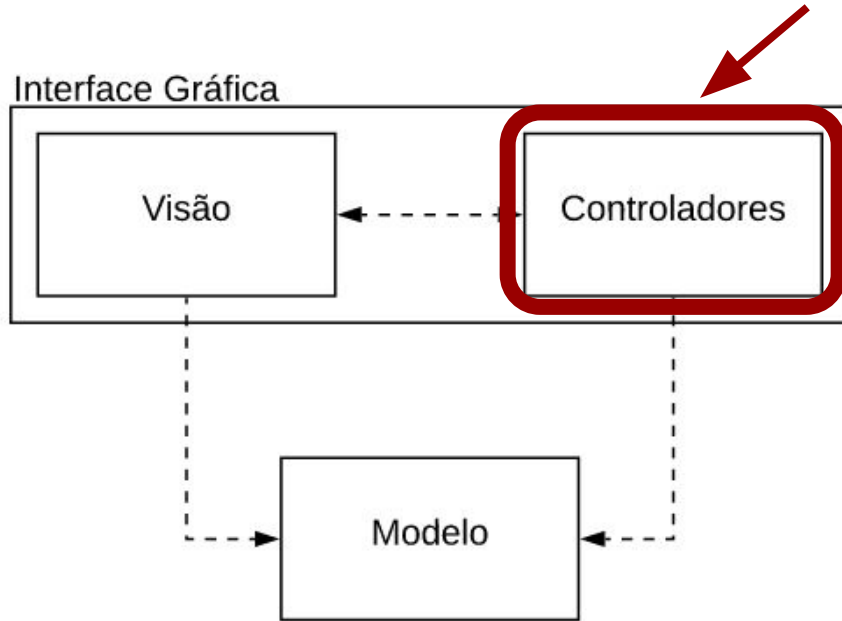
MVC define que as classes de um sistema devem ser organizadas em três grupos: **Visão, Controladoras, Modelos**

Arquitetura MVC



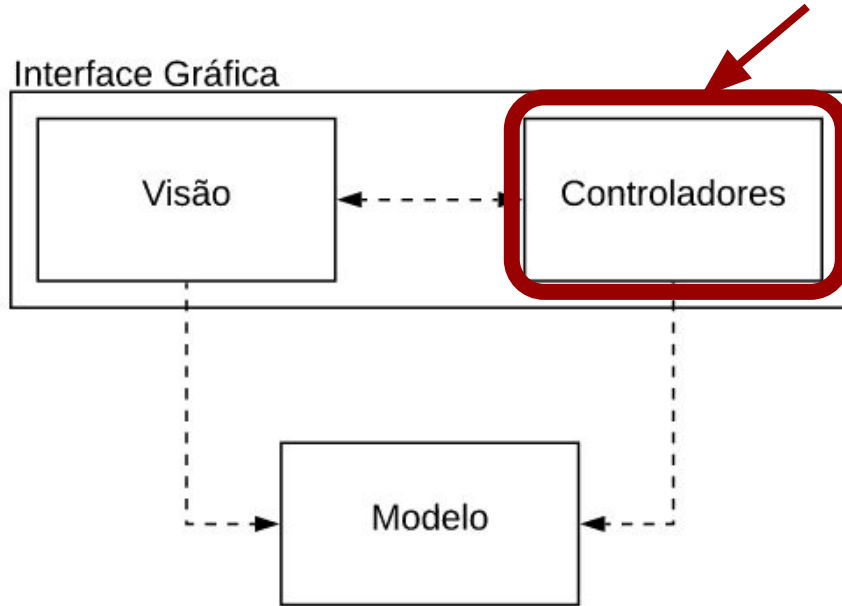
Visão: classes responsáveis pela apresentação da interface gráfica do sistema, incluindo janelas, botões, menus, barras de rolagem, etc

Arquitetura MVC



Controle: classes que tratam e interpretam eventos gerados por dispositivos de entrada, como mouse e teclado.

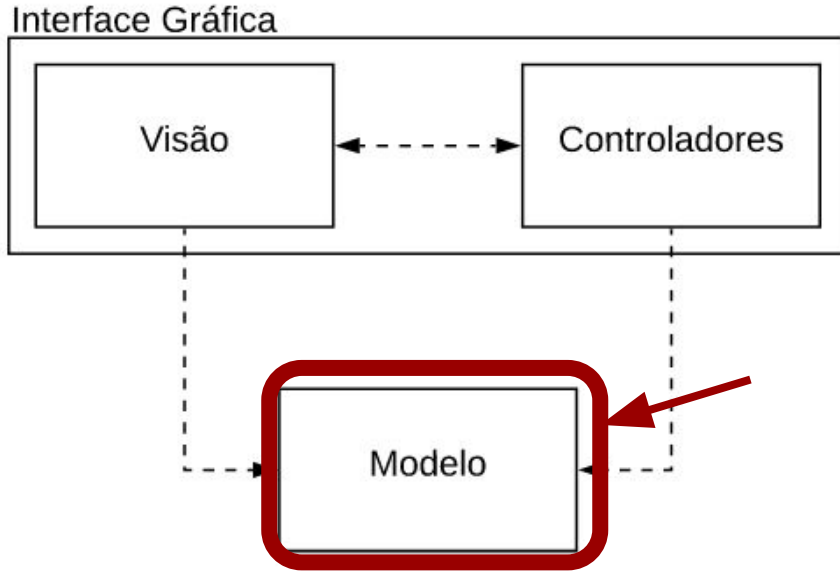
Arquitetura MVC



Ex.: Calculadora

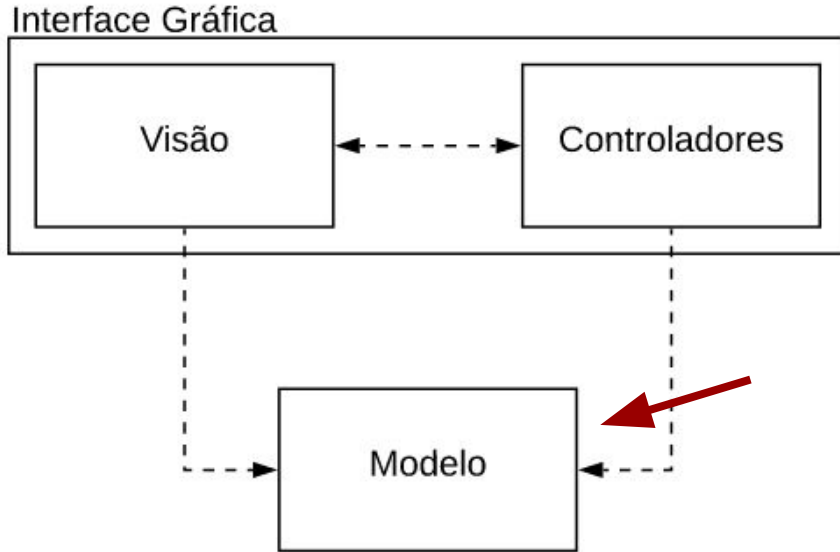
Quando o usuário clica em um botão “+”, uma classe Controladora deve capturar esse evento e chamar um método do Modelo

Arquitetura MVC



Modelo: classes que armazenam os dados manipulados pela aplicação e que têm a ver com o domínio do sistema em construção.

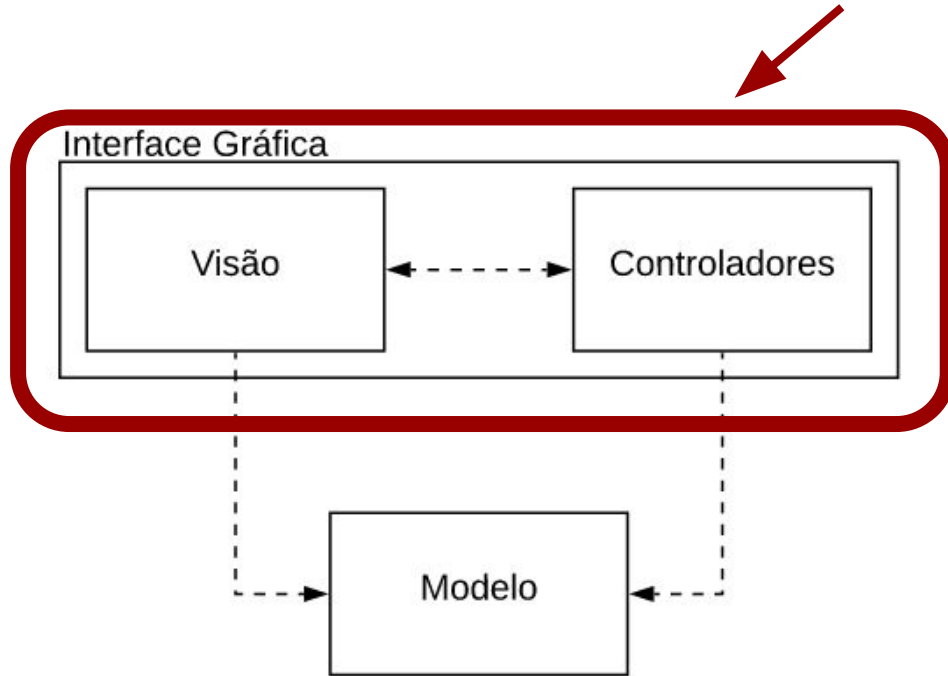
Arquitetura MVC



Classes de modelo não têm qualquer conhecimento ou dependência para classes de Visão e Controladoras



Arquitetura MVC



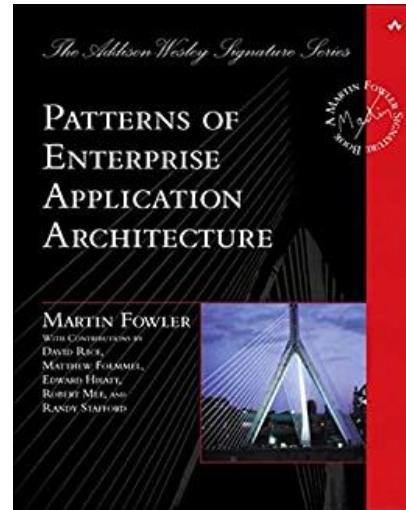
Em uma arquitetura MVC, a interface gráfica é formada por objetos de visão e por controladores

Arquitetura MVC

Porém, em muitos sistemas não existe uma distinção clara entre Visão e Controladores

Arquitetura MVC

A maioria das versões de Smalltalk não separa esses dois componentes



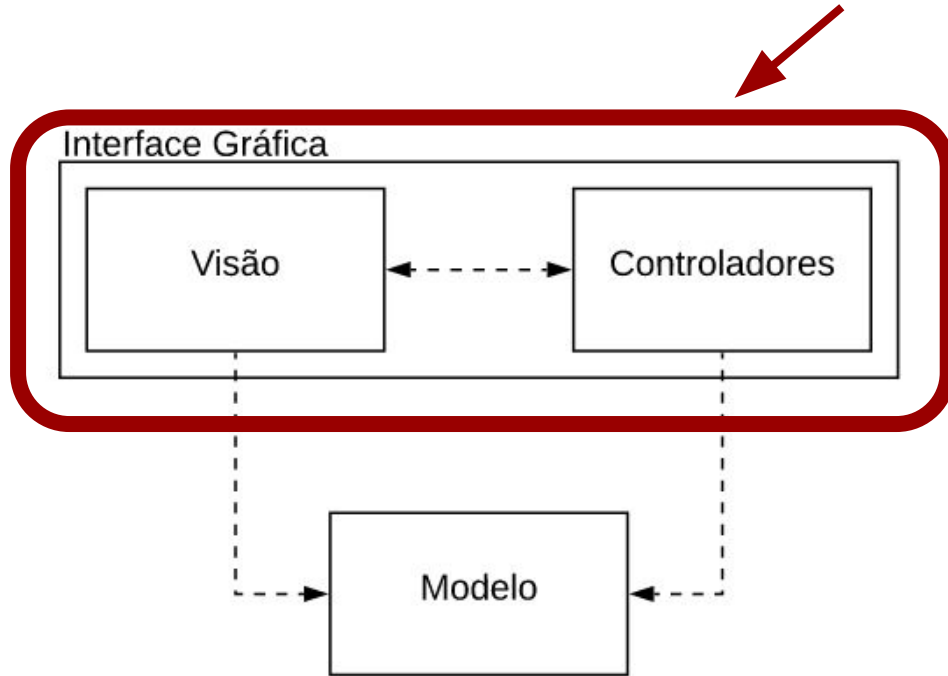
Arquitetura MVC

Podemos entender MVC como:

(Visão + Controladores) + Modelo
= Interface Gráfica + Modelo

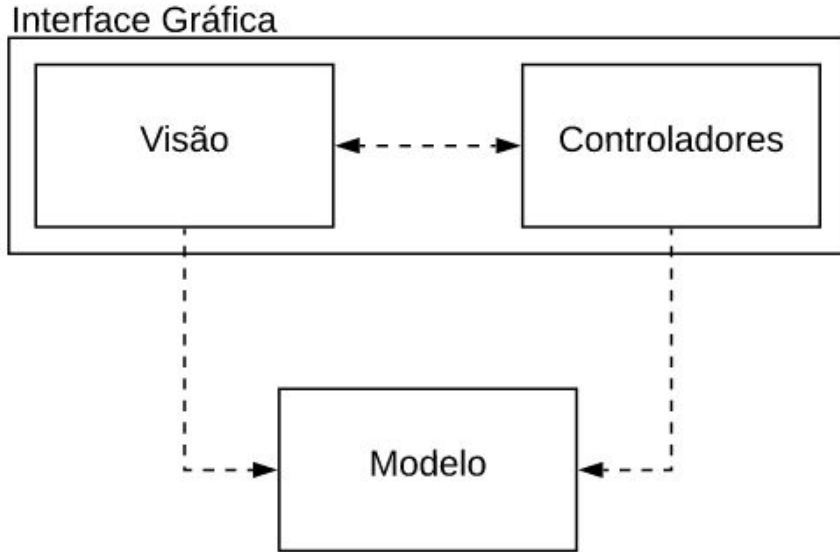


Arquitetura MVC



A interface gráfica é composta pela Visão e por Controladores

Arquitetura MVC



A Interface Gráfica pode depender do Modelo. Porém, classes de Modelo não têm dependências para classes da Interface Gráfica

Vantagens MVC

MVC favorece a especialização do trabalho de desenvolvimento



Ex.: Podemos ter desenvolvedores especialistas na implementação de interfaces gráficas (desenvolvedores front-end)

Vantagens MVC

MVC favorece a especialização do trabalho de desenvolvimento



Ex.: Desenvolvedores de classes de Modelo não precisam conhecer e implementar código de interface com usuários

Vantagens MVC

MVC permite que classes de Modelo sejam usadas por diferentes Visões



Exemplo

Um objeto de Modelo armazena dois valores (hora e minutos)



23:55

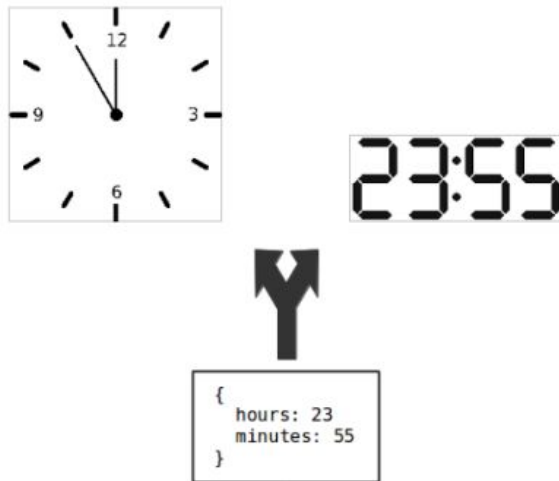


```
{  
  hours: 23  
  minutes: 55  
}
```

Exemplo

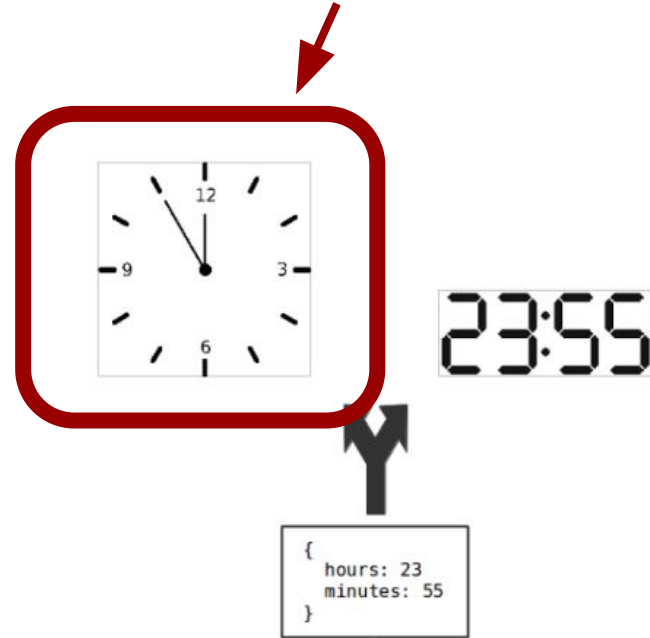
Um objeto de Modelo armazena dois valores (hora e minutos)

Duas visões diferentes



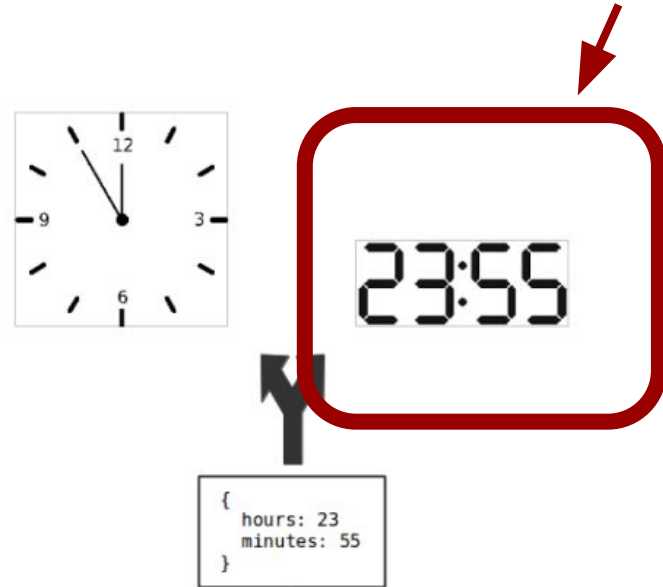
Exemplo

1) Relógio analógico



Exemplo

2) Relógio digital



Vantagens MVC

Favorece testabilidade



É mais fácil testar objetos não relacionados
com a implementação de interfaces gráficas

Vantagens MVC

Favorece testabilidade



Ao separar objetos de apresentação de objetos de Modelo, fica mais fácil testar esses últimos

MVC atualmente

MVC

Como comentamos, MVC surgiu no final da década de 70, para ajudar na construção de interfaces gráficas

MVC

Foi proposto para aplicações desktop

MVC

Foi proposto para aplicações desktop

Aplicações que incluem uma interface com janelas, botões, caixas de texto, etc

MVC

Foi proposto para aplicações desktop

Ex.: Aplicações como Word, Excel e Powerpoint

MVC

No início dos anos 2000, a Web se popularizou e a interface das aplicações migrou para HTML e, depois, para HTML e JavaScript



MVC

A confusão entre os termos MVC e três camadas surgiu então nessa época, principalmente devido ao aparecimento de frameworks MVC



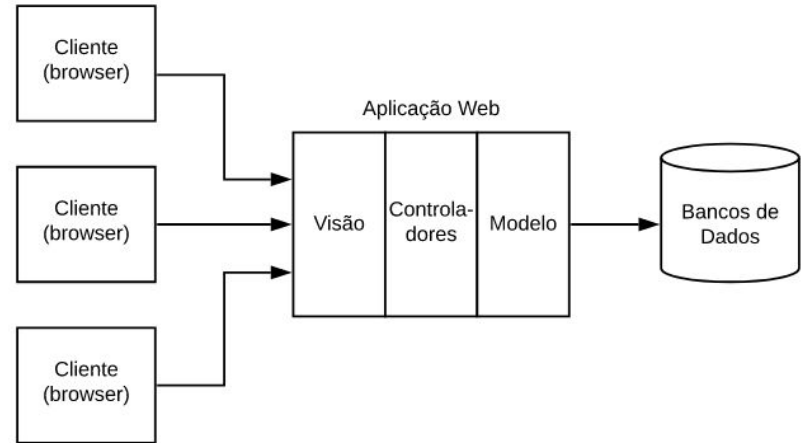
MVC

Esses frameworks expandiram e adaptaram o conceito de MVC para Web

Adaptação para construir sistemas
distribuídos

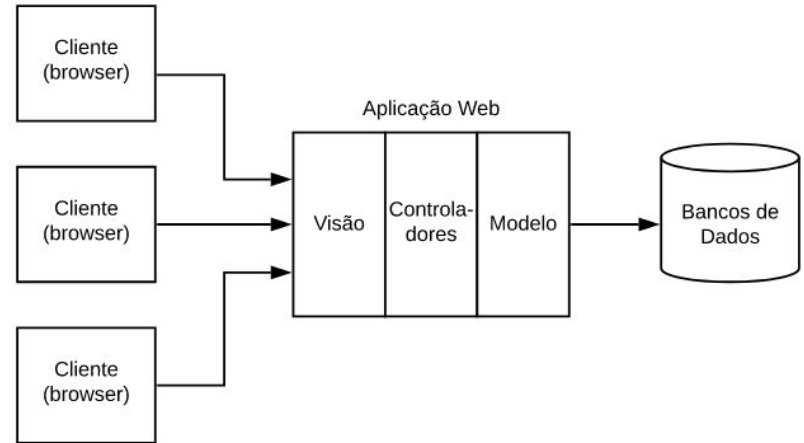
MVC em sistemas Web

Por exemplo, eles forçam a organização de um sistema Web em três partes



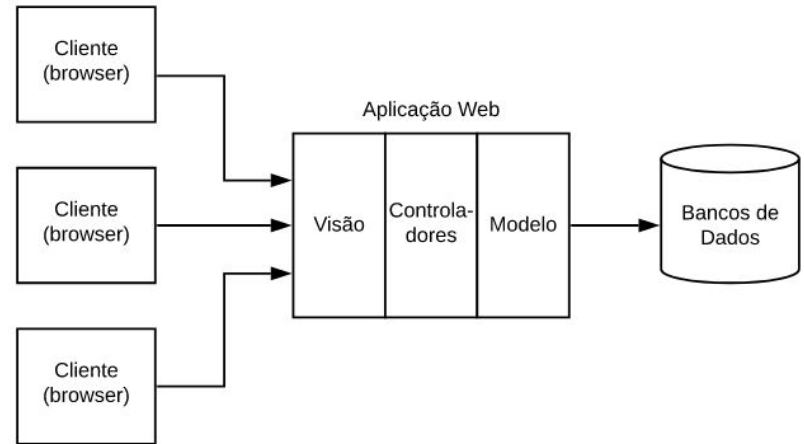
MVC em sistemas Web

Visão: composta por páginas HTML



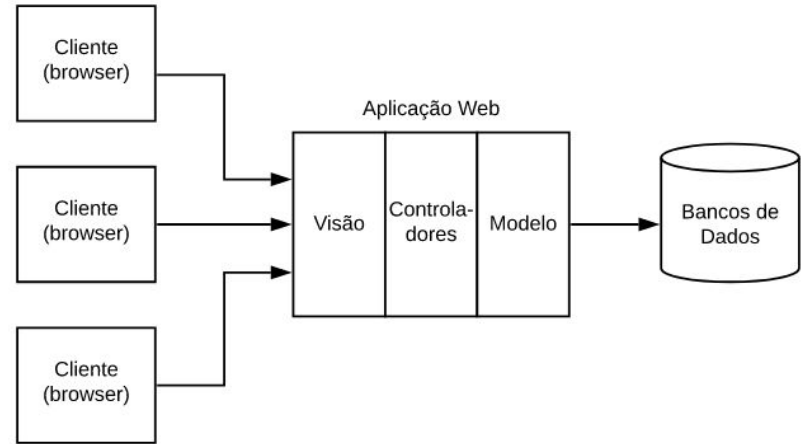
MVC em sistemas Web

Controladores: processam uma solicitação e geram uma nova visão como resposta

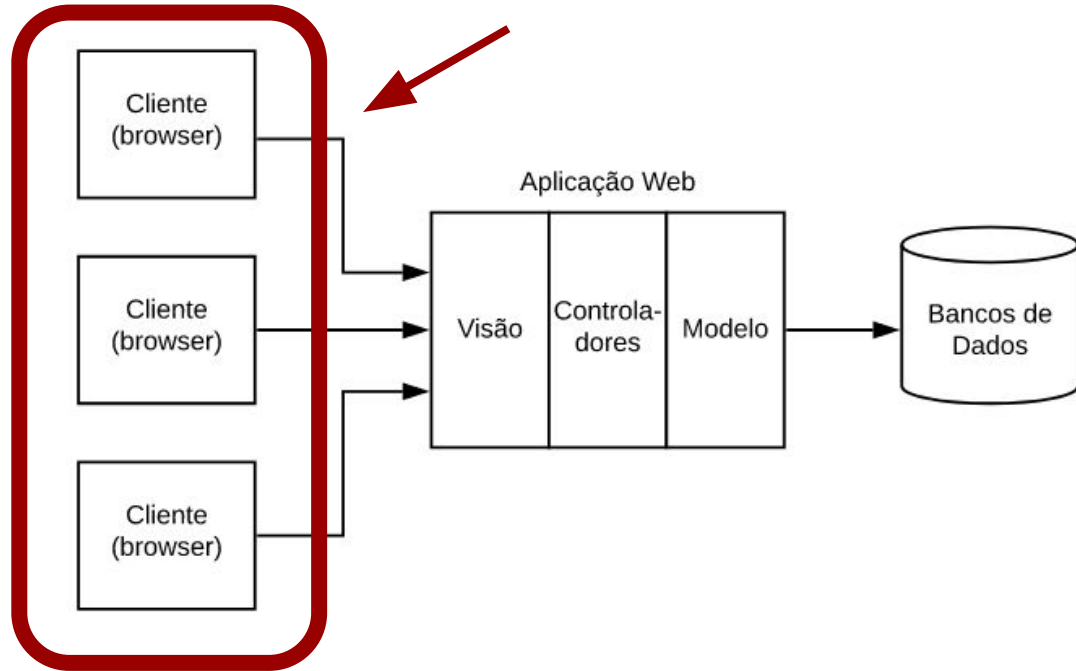


MVC em sistemas Web

Modelo: camada que persiste os dados em um banco de dados

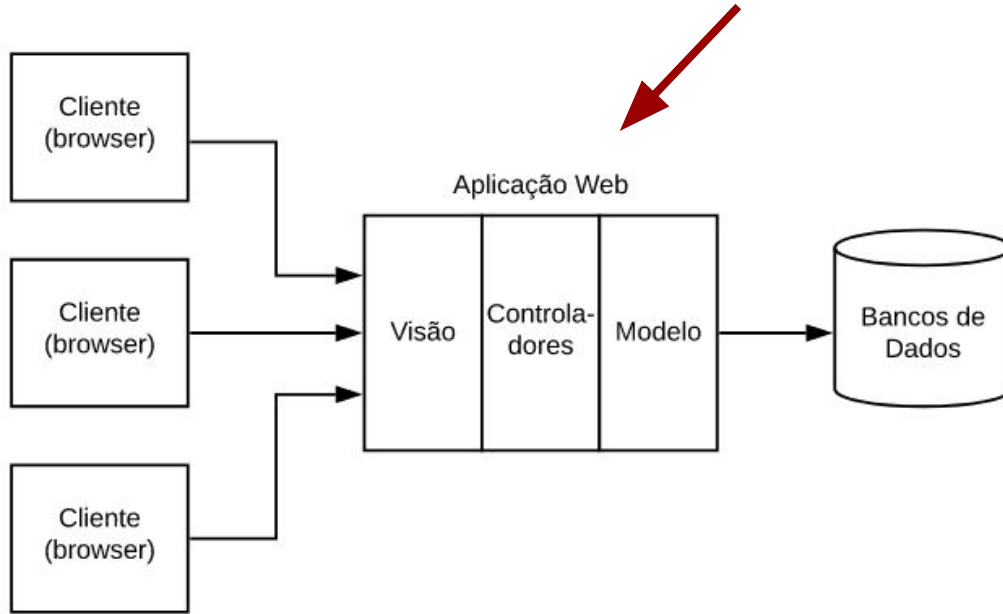


MVC em sistemas Web



- Distribuída
- Temos browsers que exibem páginas Web (interface HTML, JavaScript, etc)

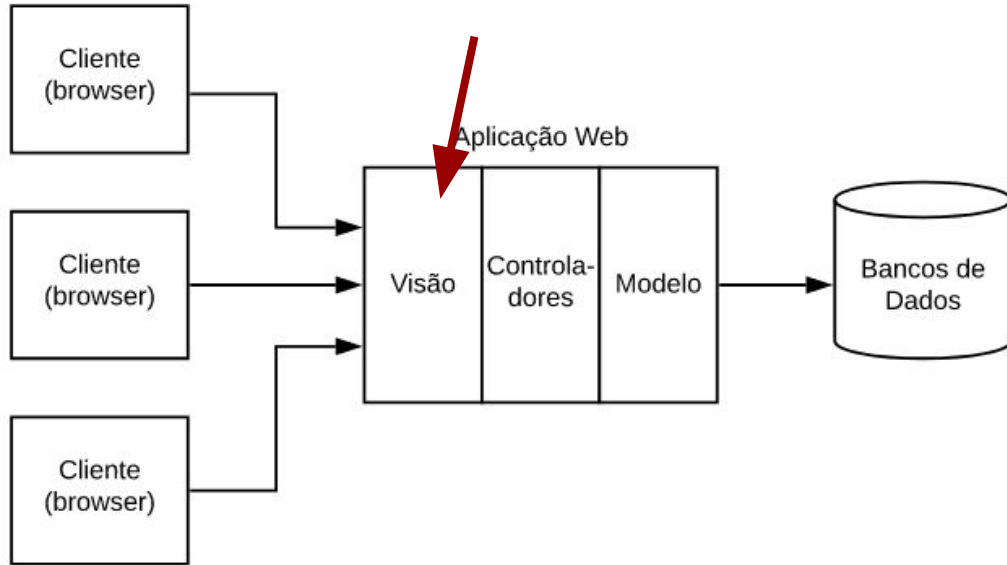
MVC em sistemas Web



MVC está na aplicação Web:

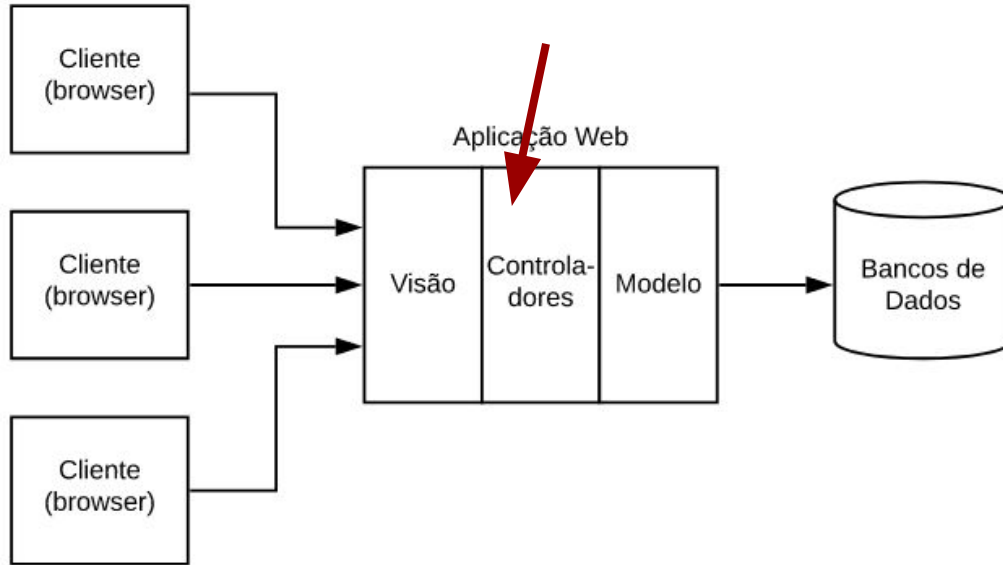
- Executa no servidor web
- Primeiro componente é o browser
- Segundo componente é o servidor web
- Usamos MVC para organizar essa aplicação

MVC em sistemas Web



Visão: Páginas HTML, CSS, JavaScript (o que o usuário visualiza), enviados para o browser para o cliente ver e interagir

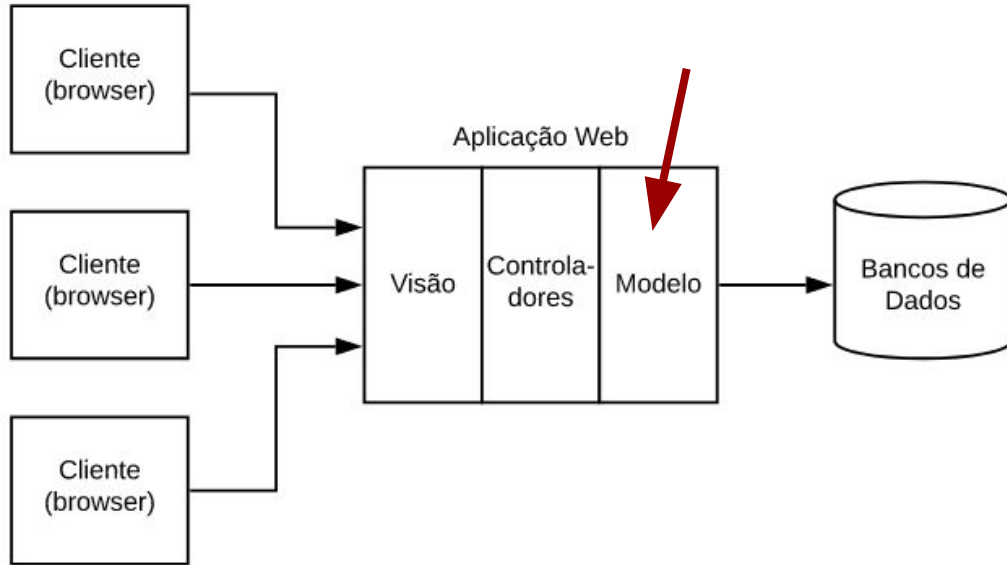
MVC em sistemas Web



Controladores:

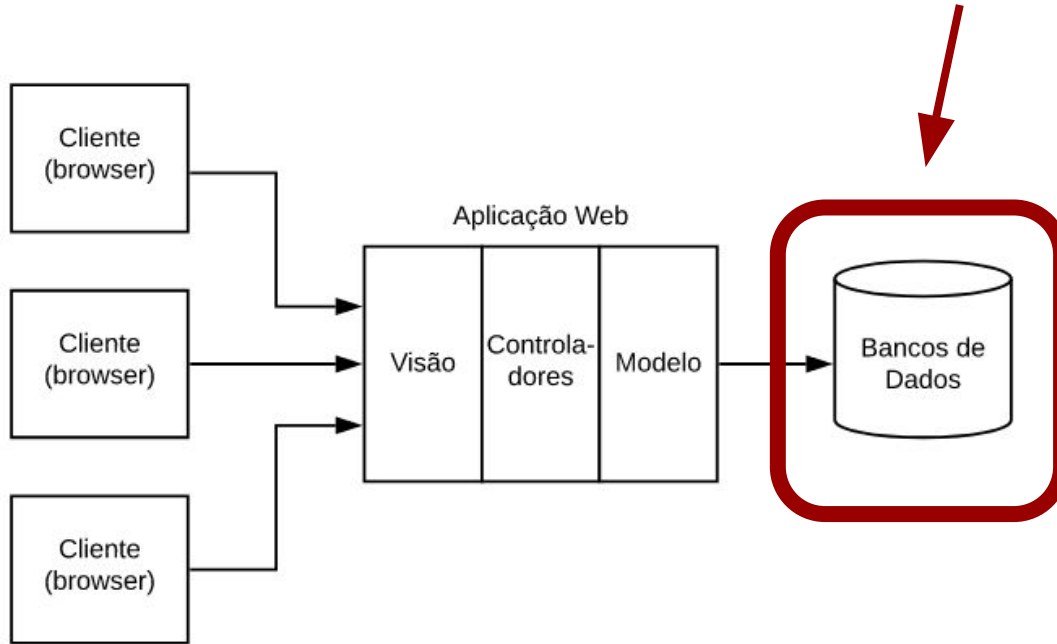
Fornecem dados para páginas HTML, recebem dados de entrada, fornecem informações para páginas de saída

MVC em sistemas Web



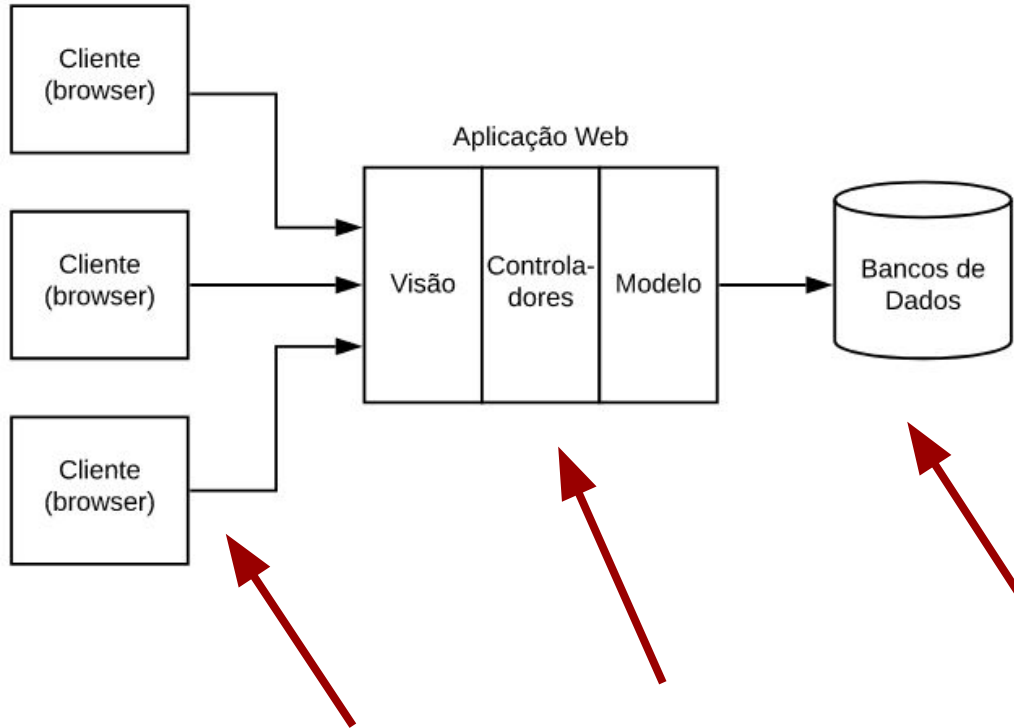
Modelo: é a camada que persiste os dados em um banco de dados

MVC em sistemas Web



O banco de dados é o terceiro componente

MVC em sistemas Web



Parece uma arquitetura com 3 camadas

MVC em sistemas Web

Os sistemas Web são parecidos com sistemas três camadas

MVC em sistemas Web

Porém, os frameworks Web mais populares optaram por usar termos típicos de MVC para nomear seus componentes

MVC em sistemas Web

Portanto, existem duas vertentes de sistemas MVC

MVC em sistemas Web

- 1) A vertente clássica, que surgiu com Smalltalk-80

MVC em sistemas Web

2) A vertente Web, que se tornou comum na década de 90 e início dos anos 2000, lembrando bastante sistemas três camadas

Referências



Capítulo 7: Arquitetura

<https://engsoftmoderna.info/cap7.html#arquitetura>

Arquitetura MVC (Model-View-Controller)

Projeto de Software

Profª Aline Brito

