

---

# AGENDA10

---

## PADRÃO MVC: MODEL VIEW CONTROLLER





## MERGULHANDO NO TEMA...

### MVC o que é?

MVC é a sigla para as palavras Model-View-Controller (Modelo-Visão-Controle), trata-se de um padrão de arquitetura para o desenvolvimento de software, podendo ser utilizado em desenvolvimento Web, Desktop ou Mobile, sem nenhuma objeção.

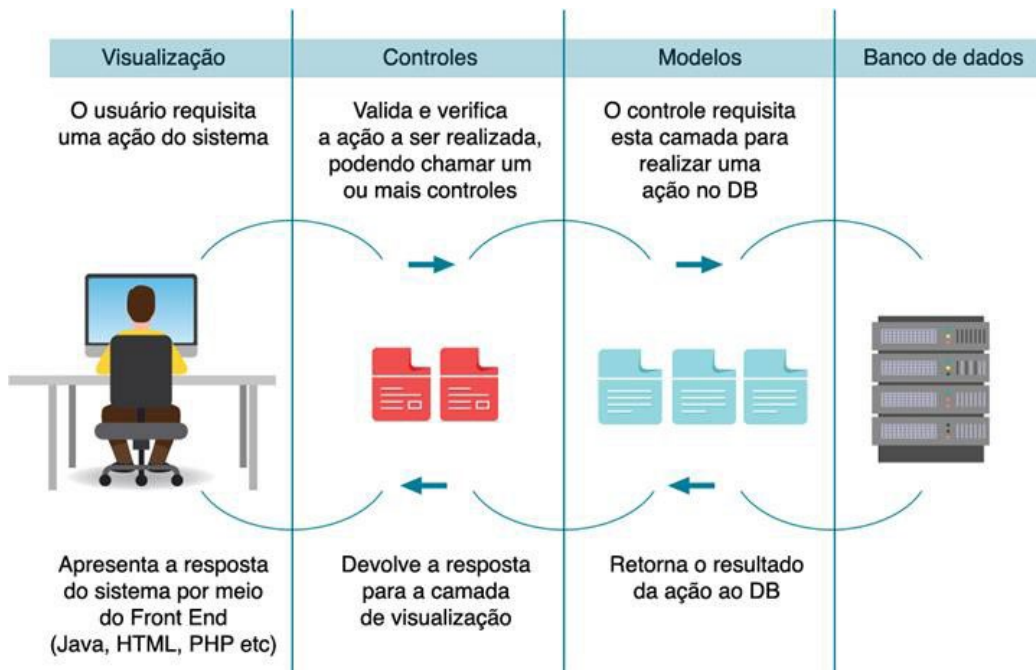


Imagem 02: Quadro explicativo do Padrão MVC

Basicamente esse padrão tem como característica a separação de regras e lógicas do negócio, da apresentação ao usuário, o que permite ao desenvolvedor maior controle sobre o projeto, tornando possível a manutenção individual de cada camada (MVC), proporcionando maior segurança no ciclo de desenvolvimento do projeto.

### Como surgiu?

Christopher Alexander, um engenheiro civil da década de 70, criou o que é considerado o primeiro padrão de projeto. Por meio dessa iniciativa e conceitos apresentados, o desenvolvimento de software começou a ser encarado de uma maneira diferente e as primeiras documentações de padrões começaram a se desenvolver, para então Trygve Reenskaug, em 1979, começar a idealização do que hoje é chamado de padrão MVC.

Basicamente, Reenskaug gerou um padrão de arquitetura que projetava o software em três camadas independentes, essas camadas são: modelo, visão e controle (MVC).

### Quais Benefícios de usar um padrão / arquitetura de projeto?

Em qualquer área é essencial seguir padrões para atingir um nível mais elevado de profissionalismo. Podemos levantar alguns benefícios que serão agregados ao projeto ao utilizar padrões para o desenvolvimento. São eles:

- Aumento de produtividade;
- Uniformidade na estrutura do site;
- Redução de complexidade no código;
- Sites ficam mais fáceis de manter;
- Códigos e sites mais fáceis para manutenções e reuso;
- Facilita a documentação;
- Estabelece um vocabulário comum de projeto entre desenvolvedores;
- Permite a reutilização de módulos do site em outros sites;
- É considerada uma boa prática utilizar um conjunto de padrões para resolver problemas maiores que, sozinhos, não conseguiriam;
- Ajuda a construir sites confiáveis com arquiteturas testadas;
- Reduz o tempo de desenvolvimento de um projeto / site.

### O padrão MVC

Quando utilizamos o padrão MVC, buscamos sua principal característica: isolar as regras de negócios, da lógica e da apresentação de interface ao usuário, buscando como principal benefício a possibilidade de que cada item, em cada camada, seja modificados sem a necessidade da alteração de outras camadas, tornando o projeto mais flexível e com uma grande possibilidade de reaproveitamento de código.

Vamos agora entender melhor essas três camadas:

- Model;
- View;
- Controller.

#### Camada Model

Basicamente é responsável pela modelagem e manipulação de informações de forma detalhada, então, todas as entidades / classes, consultas, cálculos e regras de negócio do site ou sistema devem permanecer nessa camada que tem acesso a toda e qualquer informação modelada ou não, proveniente na maioria dos casos de um Banco de Dados ou arquivo XML.

Exemplo:

- Modela os dados e o comportamento por trás do processo de negócios;
- Preocupa-se com o armazenamento, manipulação e geração de dados;
- É um encapsulamento de dados e de comportamento independente da apresentação.

## Camada View

Como o próprio nome desta camada indica, fica a seu cargo apresentar tudo ao usuário final, então, é fácil afirmar que toda a interface faz parte dessa camada. Logo, todos os dados, informações, gráficos etc quando exibidos para o usuário final, é de responsabilidade desta camada.

Exemplo:

- Os elementos de exibição ao cliente: HTML, XML, ASP, Applets;
- Interface com o usuário;
- Camada usada para receber a entrada de dados e apresentar o resultado.

## Camada Controller

Responsável pelo fluxo de informação que passa pelo site. Esta camada irá gerir e definir quais dados/informações ou regras devem ser acionadas e para onde serão encaminhadas para posteriormente serem exibidas.

Em resumo, esta camada age como uma intermediária, buscando as informações da camada Model e entregando para a camada View para a exibição ao usuário final. Portanto, o controle e as decisões do site devem fazer parte desta camada.

Exemplo:

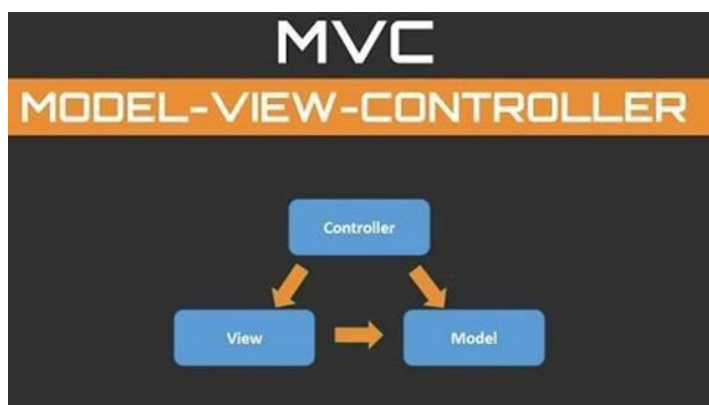
- Um arquivo que recebe os dados de um formulário e os encaminha para a gravação dos mesmos em banco de dados ou encaminha o resultado de uma pesquisa para o usuário posteriormente visualizar.



*Mas como funciona este padrão?*

*Assista aos vídeos a seguir e entenda o funcionamento deste padrão.*

O que é MVC? Aprenda MVC de maneira fácil



Disponível em: <https://www.youtube.com/watch?v=ZW2JLtX4Dag>.

Para entender melhor o funcionamento do modelo, imaginemos uma conversa entre o funcionamento do padrão e a relação entre suas camadas, (Tableless.com.br).



Imagem 03: : GEEaD - Diálogo entre as camadas MVC. Adaptado de Tableless.com.br<sup>1</sup>



## VOCÊ NO COMANDO

*Utilizando o que foi visto até agora....*

### 1. Vamos relacionar alguns elementos de um site a uma camada do padrão MVC.

.PHP

#### A) Classe Usuário.

Essa classe traz todas as informações pertinentes, inclusive a persistência dessas informações em Banco de Dados.

CAMADA: \_\_\_\_\_

.CSS

.HTML

#### B) Formulário de Contato.

Esse formulário traz os campos de texto para os usuários digitarem informações, e exibe informações sobre a empresa como e-mail, telefone etc.

CAMADA: \_\_\_\_\_

.PHP

#### C) Instância do Usuário.

Esse arquivo instancia a classe usuário, obtém as informações de um formulário e insere essas informações no objeto instanciado.

CAMADA: \_\_\_\_\_

**D) SlideShow de fotos.**

Apresenta fotos de setores da empresa ao visitante do site.

CAMADA: \_\_\_\_\_

**E) Classe Fornecedor.**

Essa classe traz todas as informações pertinentes a fornecedores, inclusive a persistência dessas informações em Banco de Dados.

CAMADA: \_\_\_\_\_

**F) Gráficos de Vendas.**

Apresenta gráfico de venda dos últimos anos de empresa.

CAMADA: \_\_\_\_\_

**G) Página Sobre a Empresa.**

Esta página traz todas as informações da empresa, desde seu surgimento até o momento atual.

CAMADA: \_\_\_\_\_

**H) Validação de Fornecedor.**

Código que permite a liberação do pagamento, somente após a entrega do pedido com as métricas definidas no contratado pelo fornecedor.

CAMADA: \_\_\_\_\_

**I) Data e hora do site.**

Apresenta data e hora para os usuários que acessam a página.

CAMADA: \_\_\_\_\_

**A seguir, confira se você conseguiu resolver os desafios propostos!**

**A) Classe Usuário.**

CAMADA: MODEL

**B) Formulário de Contato.**

CAMADA: VIEW

**C) Instância do Usuario.**

CAMADA: CONTROLLER

**D) Slide Show de fotos.**

CAMADA: VIEW

**E) Classe Fornecedor.**

CAMADA: MODEL

**F) Gráficos de Vendas.**

CAMADA: VIEW

**G) Página Sobre a Empresa.**

CAMADA: VIEW

**H) Validação de Fornecedor.**

CAMADA: MODEL

**I) Data e hora do site.**

CAMADA: VIEW

**Explicações:**

**CAMADA MODEL:** Como descrevemos anteriormente, esta camada é, basicamente responsável pela modelagem e manipulação de informações de forma detalhada, então, todas as entidades/classes, consultas, cálculos e regras de negócio do site ou sistema devem permanecer nessa camada.

Então o resultado deve ficar:

Classe Fornecedor:

Classe Usuário. Validação  
de Fornecedor.



**CAMADA VIEW:** responsável por apresentar tudo para que o usuário final visualize e interaja com a interface então, é fácil afirmar que toda a interface faz parte dessa camada, logo, fazem parte desta camada todos os dados, como informações, gráficos etc.

Então nessa camada deve ficar:

Formulário de Contato.

Página Sobre a Empresa.

Data e hora do site.

Gráficos de Vendas.

SlideShow de fotos.



**CAMADA CONTROLLER:** controla o fluxo de informação que passa pelo site. Esta camada irá gerir e definir quais dados/informações ou regras devem ser acionadas e para onde serão encaminhadas para posteriormente serem exibidas.

Então, nessa camada deve ficar:

