

MVC

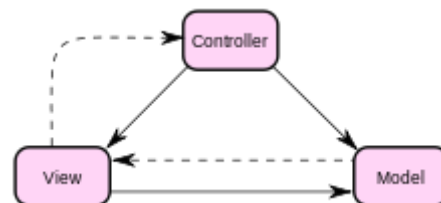
Origem: Wikipédia, a enciclopédia livre.

Model-view-controller (MVC), em português **modelo-vista-controlador**, é um padrão de arquitetura de software (não confundir com um *design pattern*) que separa a representação da informação da interação do usuário com ele.

É normalmente usado para o desenvolvimento de interfaces de usuário que divide uma aplicação em três partes interconectadas. Isto é feito para separar representações de informação internas dos modos como a informação é apresentada para e aceita pelo usuário.^{[1][2]} O padrão de projeto MVC separa estes componentes maiores possibilitando a reutilização de código e desenvolvimento paralelo de maneira eficiente.

O *modelo* (*model*) consiste nos dados da aplicação, regras de negócios, lógica e funções. Uma *visão* (*view*) pode ser qualquer saída de representação dos dados, como uma tabela ou um diagrama. É possível ter várias visões do mesmo dado, como um gráfico de barras para gerenciamento e uma visão tabular para contadores. O *controlador* (*controller*) faz a mediação da entrada, convertendo-a em comandos para o modelo ou visão. As ideias centrais por trás do MVC são a reusabilidade de código e separação de conceitos

Tradicionalmente usado para interfaces gráficas de usuário (GUIs), esta arquitetura tornou-se popular para projetar aplicações web e até mesmo para aplicações móveis, para desktop e para outros clientes.^[3] Linguagens de programação populares como Java, C#, Ruby, PHP e outras possuem frameworks MVC populares que são atualmente usados no desenvolvimentos de aplicações web.



Um diagrama simples exemplificando a relação entre *Model*, *View* e *Controller*. As linhas sólidas indicam associação direta e as tracejadas indicam associação indireta.

Índice

História

Interação dos componentes

Uso em aplicações web

Justificativa

Plataformas de desenvolvimento

COLDFUSION

ActionScript 3

ASP

.NET

Delphi

Harmony

Java

Perl

PHP

Python

Ruby

Ver também

Referências

História

O padrão MVC foi descrito pela primeira vez em 1979^[4] por [Trygve Reenskaug](#) que trabalhava no [Smalltalk](#), na [Xerox PARC](#).

Interação dos componentes

Além de dividir a aplicação em três tipos de componentes, o desenho MVC define as interações entre eles.

- Um **controlador (controller)** envia comandos para o modelo para atualizar o seu estado (por exemplo, editando um documento). O controlador também pode enviar comandos para a visão associada para alterar a apresentação da visão do modelo (por exemplo, percorrendo um documento).
- Um **modelo (model)** armazena dados e notifica suas visões e controladores associados quando há uma mudança em seu estado. Estas notificações permitem que as visões produzam saídas atualizadas e que os controladores alterem o conjunto de comandos disponíveis. Uma implementação *passiva* do MVC monta estas notificações, devido a aplicação não necessitar delas ou a plataforma de software não suportá-las.
- A **visão (view)** Gera uma representação (visão) dos dados presentes no modelo solicitado.

Uso em aplicações web

Apesar de desenvolvida originalmente para computação pessoal, o MVC foi amplamente adaptado como uma arquitetura para as aplicações [World Wide Web](#) em todas as linguagens de programação maiores. Muitos [frameworks de aplicação](#) comerciais e não comerciais foram criados tendo como base esse modelo. Estes frameworks variam em suas interpretações, principalmente no modo que as responsabilidades MVC são divididas entre [cliente](#) e [servidor](#).

Os frameworks web MVC mais recentes levam uma abordagem de [thin client](#) que colocou quase o modelo, a visão e a lógica do controlador inteiros no servidor. Nesta abordagem, o cliente envia requisições de [hiperlink](#) ou entrada de [formulário](#) ao controlador e então recebe uma página web completa e atualizada (ou outro documento) da visão. O modelo existe inteiramente no servidor. Como as tecnologias de cliente amadureceram, frameworks como [JavaScriptMVC](#) e [Backbone](#) foram criados o que permite que os componentes MVC executem parcialmente no cliente (ver também [AJAX](#)).

Um caso prático é uma [aplicação web](#) em que a visão é um documento HTML (ou derivado) gerado pela aplicação. O controlador recebe uma entrada GET ou POST após um estímulo do utilizador e decide como processá-la, invocando objetos do domínio para tratar a lógica de negócio, e por fim invocando uma visão para apresentar a saída

Justificativa

Com o aumento da complexidade das aplicações desenvolvidas, sempre visando a programação orientada a objeto, torna-se relevante a separação entre os dados e a apresentação das aplicações. Desta forma, alterações feitas no *layout* não afetam a manipulação de dados, e estes poderão ser reorganizados sem alterar o *layout*.

Esse padrão resolve este problema através da separação das tarefas de acesso aos dados e lógica de negócio, lógica de apresentação e de interação com o utilizador, introduzindo um componente entre os dois: o controlador

Plataformas de desenvolvimento

COLDFUSION

- [ColdBox](#)
- [FW/1](#)
- [FarCry](#)
- [FuseBox](#)
- [CF Wheels](#)
- [Mach-II](#)

- [ColdSpring](#)
- [ModelGlue](#)
- [LightWire](#)

ActionScript 3

- [Cairngorm](#) - da Adobe
- [PureMVC](#)
- [Swiz](#)

ASP

- [ASP Xtreme Evolution](#)
- [Toika](#)
- [AJAXED](#)

.NET

- [ASP.NET MVC](#) - oficial da [Microsoft](#)
- [C#](#) - oficial da [Microsoft](#)
- [Versões Existentes: MVC 2, MVC 3, MVC 4, MVC 5](#)

Delphi

- [MVCB](#)^[5]^[6]

Harmony

- [Harmony Framework](#)- Oficial da [Vilesoft](#)

Java

- [Apache Struts](#)
- [Brutos Framework](#)
- [Click Framework](#)
- [JSF](#)
- [Mentawai](#)
- [MVC 1.0 \(Ozark\)](#)
- [Neo Framework](#)
- [PlayFramework](#)
- [Spring MVC](#)
- [Tapestry](#)
- [VRaptor](#)
- [WebWork](#)

Perl

- [Catalyst](#)
- [Mojolicious](#)
- [Gantry](#)

PHP

- [Akelos](#)
- [CakePHP](#) - para as versões 4 e 5
- [CodeIgniter](#) - para as versões 4 e 5
- [FuelPHP](#) - para versões 5.3+
- [iGrape](#)
- [Kohana Framework](#)- para a versão 5
- [LightVC](#) - para a versão 5.
- [Laravel](#) - para a versão 5.3+
- [Megiddo](#) - para a versão 5
- [Oraculum PHP Framework](#)- para a versão 5
- [PageletBox](#) - IDE para Celular WAP com framework MVC embutido em [PHP 5](#)
- [Phalcon](#) - Extensão em C visando alta performance e baixo consumo de recursos - para a versão 5
- [PHPBurn](#)
- [PHPonTrax](#) - para a versão 5
- [PRADO](#) - para a versão 5
- [Seagull](#)
- [Spaghetti*](#)
- [Symfony](#) - para a versão 5
- [Vórtice Framework](#)
- [XPT Framework](#)- para a versão 5
- [Yii Framework](#)- para a versão 5
- [Zend Framework](#)- da ZEND, mantenedora oficial do [PHP 5](#) no padrão MVC
- [Zend Framework](#)- da ZEND, mantenedora oficial do [PHP 6](#) no padrão MVC

Python

- [Django](#)
- [TurboGears](#)
- [Web2py](#)
- [Zope](#) / [Plone](#)

Ruby

- [Rails](#)
- [Merb](#)

ADVPL

- [Microsiga Protheus](#)

Ver também

- [Arquitetura multicamada](#)
- [Modelo em três camadas](#)
- [Cliente-servidor](#)
- [Cluster](#)
- [Lista de protocolos de redes](#)
- [Protocolo RPC](#)
- [Redes de computadores](#)
- [Sistema de processamento distribuído](#)
- [Sockets](#)

Referências

1. "More deeply, the framework exists to separate the representation of information from user interaction. The DCI Architecture: A New Vision of Object-Oriented Programming (http://www.artima.com/articles/dci_vision.html) – Trygve Reenskaug and James Coplien – 20 de março de 2009.
 2. Burbeck (1992): "... the user input, the modeling of the external world, and the visual feedback to the user are explicitly separated and handled by three types of object."
 3. Davis, Ian. «What Are The Benefits of MVC?» (<http://blog.iandavis.com/2008/12/what-are-the-benefits-of-mvc/>) *Internet Alchemy*. Consultado em 29 de novembro de 2016.
 4. Trygve M. H. Reenskaug/MVC (<http://heim.ifi.uio.no/~trygver/themes/mvc/mvc-index.html>) – XEROX PARC 1978-79
 5. [Git/Fontes MVCBr \(https://github.com/amarildolacerda/MVCBr\)](https://github.com/amarildolacerda/MVCBr)
 6. [Blog TireiDeLetra \(http://tireideletra.com.br\)](http://tireideletra.com.br)
-

Obtida de "<https://pt.wikipedia.org/w/index.php?title=MVC&oldid=53099830>

Esta página foi editada pela última vez às 03h13min de 10 de setembro de 2018.

Este texto é disponibilizado nos termos da licença [Atribuição-CompartilhaIgual 3.0 Não Adaptada \(CC BY-SA 3.0\)](#) da [Creative Commons](#) pode estar sujeito a condições adicionais. Para mais detalhes, consulte [as condições de utilização](#)