## **11** 02

## O padrão Model-View-Controller

## Capítulo 6 - O padrão Model-View-Controller

Antes de continuarmos com o nosso curso, façamos uma pausa para falarmos de um assunto importante. Ele nos ajudará a entender diversas questões relacionadas a *Desenvolvimento Web*.

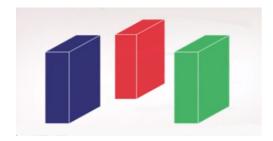
Não importa a tecnologia que estamos utilizando, seja ela Java, PHP, Ruby ou C Sharp, o mundo Web funciona da seguinte maneira:

- O usuário final abre o navegador e digita uma URL;
- A URL faz o que chamamos de requisição para o servidor;
- O servidor devolve uma resposta, geralmente um HTML renderizado, para o usuário final.

O que vamos discutir é como o servidor é programado. Antigamente o código do desenvolvedor era dessa forma:

[linhas	de	interf	ace	<b>e</b> ]		
				_	_	
						_
[linhas	de	regra	de	negó	cio]	
					_	
						_
[linhas	de	acesso	a	banc	o de	dados]
				_	_	
						_

E assim por diante. O código era totalmente procedural, misturado e complexo (chamávamos de "código macarrônico"). Era muito difícil mantê-lo no futuro. Precisávamos de *boas práticas* em programação. E uma delas era separar esses diversos tipos de códigos, com responsabilidades definidas, em arquivos diferentes, para serem mais fáceis de serem lidos e mantidos.



A partir daí vamos repensar a programação do servidor:

- O usuário final abre o navegador e digita uma URL;
- A URL faz a requisição para o servidor;

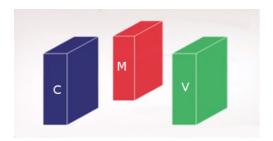
- Uma primeira instância recebe e lê esta requisição e a interpreta para saber a qual regra de negócio ela deve ser levada;
- A regra de negócio é processada e entrega um *output* para a instância de interface;
- A interface dá a resposta para o usuário.

## Cada etapa tem um nome:

• Regras de Negócio: Model

• Interface: View

• Receptor da requisição: Controller



A esta maneira de organização do código e de suas responsabilidades - *Model-View-Controller* - damos o nome de *MVC*. Esta ideia é tão popular que todo *framework* moderno de desenvolvimento para web a usa.