

02

O Padrão MVC

Até agora vimos como nós clientes utilizamos nosso navegador em nossa máquina para enviar uma requisição ao servidor que está rodando um apache com um programa PHP. Este PHP realiza alguma ação e devolve uma resposta ao cliente.



Deste modo temos uma arquitetura cliente-servidor, o cliente consome dados e o servidor executa lógicas, gera dados, faz buscas no banco e devolve ao cliente. Podemos ter vários clientes, para um mesmo servidor.

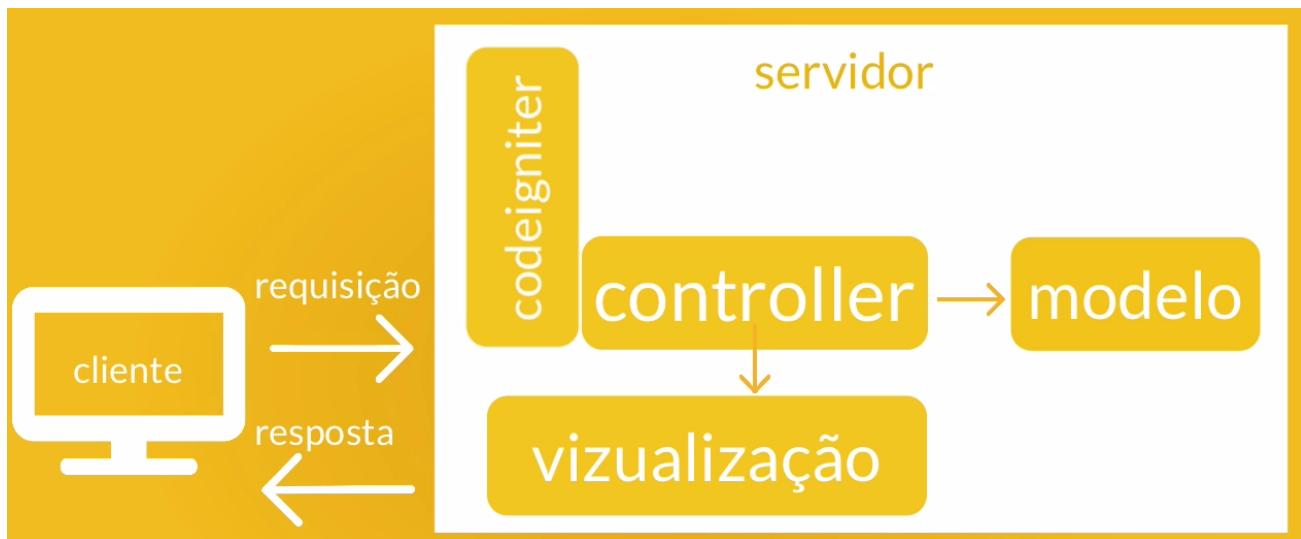
Vamos entender o que acontece quando nosso cliente envia uma requisição ao nosso servidor, primeiramente temos um arquivo PHP, o codeigniter, um cara que está frente de todos, ele cuida das requisições e as envia para o local correto, no caso os nossos controllers, este executa tarefas como:

- adicionar produtos.
 - fazer buscas.
 - logar o usuário.
 - gerar nota fiscal.

Executamos essa lógica no controller, acessando os dados que modelam nosso negócio, como exemplo: produtos, vendas, compras, clientes, esses são nossos modelos de negócio.

É tarefa do controller executar nossas lógicas de negócio e fazer o direcionamento para a página, esta página é a nossa visualização.

Relembrando, o cliente envia uma requisição que chega ao codeigniter, este redireciona para um controller nosso, o controller faz alguma coisa com o modelo (acessa o banco, gera relatórios etc...) e envia para a visualização, no nosso caso um HTML.



Vimos nesta aula que temos quatro peças interessantes, o nosso front-controller(codeigniter), nossos controllers, nossos modelos e as visualizações. Em inglês Model, View, Controller, mais conhecido por MVC, este padrão de desenvolvimento vem evoluindo e no nosso curso definimos as responsabilidades da seguinte forma:

Model: - Tudo aquilo que modela nossos objetos de negócio - Acessa o banco de dados

Controller: - Classes que preparam a lógica de negócio - Fazem o direcionamento para a visualização

Views - Os arquivos PHP que geram nosso HTML que é devolvido para o cliente.