

UTILIZAÇÃO DE ARQUITETURA EM CAMADAS BASEADA NO MODEL VIEW CONTROLLER, EM APLICAÇÕES WEB

Viviani Priscila Piloni VILHEGAS¹

RESUMO: Este trabalho procura mostrar a importância da utilização de um modelo de arquitetura de aplicações, neste caso o Model View Controller (MVC), para desacoplar a lógica da aplicação, da interface do usuário e do fluxo da aplicação, em aplicações web. Permitindo que a mesma lógica de negócios possa ser visualizada e acessada por várias interfaces.

Palavras-chave: Programação. Arquitetura em camadas. Lógica. Interface. Web.

1 INTRODUÇÃO

O MVC (Model View Controller) foi inicialmente descrito por Trygve Reenskaug em 1979 quando trabalhava no projeto Smaltalk no Centro de Pesquisa da Xerox em Palo Alto, Califórnia. O paradigma MVC foi proposto para ser a ligação entre o modelo intelectual do usuário e o modelo computacional.

Em um projeto de software baseado no padrão MVC, a arquitetura básica definida é com três camadas.

A primeira camada, a *model*, implementa o modelo representando a estrutura de baixo nível do projeto, podendo ser o modelo objeto-relacional que implementa a camada de dados. Já a segunda, a *controller*, implementa a camada responsável pelo gerenciamento de eventos no projeto, tais como cliques do usuário, chamando a camada *model* para processar os eventos. E a terceira camada, a *view*, gera a interface com usuário de modo que esta somente requisiute o processamento de eventos pelo *controller*.

É muito importante que a implementação das camadas seja feita de forma que a inversão da ordem não acarrete problemas por dependência.

¹ Coordenadora de TI das Faculdades Integradas “Antonio Eufrásio de Toledo” de Presidente Prudente. piloni@unitoledo.br

Neste artigo será possível conhecer a utilização do MVC em aplicações web, mostrando que sua aplicação em sistemas grandes e complexos, facilita o trabalho, desde a organização do projeto até as divisões das responsabilidades.

2 ASP .NET MVC

2.1 Definição

O ASP .NET MVC é uma implementação do MVC para o ASP .NET em um framework com o objetivo de criar aplicações WEB no padrão MVC e fornecer uma alternativa ao modelo WebForm do ASP .NET. Já que os desenvolvedores web tradicionais eram acostumados com a manipulação direta dos elementos HTML, JavaScript e linguagens *server side*, o ASP .NET WebForms apresentou-se de forma estranha, apesar de ter ganho na produtividade. Com isto, em 2007 a Microsoft lançou um *preview* da *framework* para desenvolvimento de aplicações web, o ASP .NET MVC.

Este modelo de programação mostra como objetivos principais:

- separação de responsabilidades: a ideia principal da *framework* MVC é separar as diferentes responsabilidades do código, ou seja, “controladores” somente processam as requisições realizadas pelo *browser*, “modelo” é responsável isoladamente pelo controle de acesso aos dados e “visões” são responsáveis apenas pela exibição das informações;
- incentivo a boas práticas: é possível visualizar isto na escrita de testes unitários;
- fácil manutenibilidade: como as aplicações trabalham com diversas visualizações para um mesmo controlador, torna-se fácil a tarefa de adicionar ou remover características.

Esta arquitetura ocorre em função ao mecanismo de rotas.

2.2 Mecanismo de Rotas

Na arquitetura WebForms existe uma *url* que aponta para um arquivo físico no servidor (.aspx), onde estão implementadas todas as ações solicitadas pelo *browser*. No ASP .NET MVC é notada uma pequena diferença, pois a requisição é tratada por um elemento controlador e este decide como gerar a visualização, portanto, ao invés de apontar para um arquivo físico, é apontado para uma ação dentro de um elemento conceitual, o controlador. Com isto é necessária a existência de uma rota para definir o caminho final dos dados.

A definição de rotas da aplicação ASP .NET MVC se encontra no arquivo "*Global.asax*". Além da rota é preciso especificar valores padrão para cada elemento da rota. O framework fornece um padrão de roteamento, mas é possível personalizar de acordo com a necessidade do desenvolvedor.

Basicamente para seu funcionamento uma requisição do browser é mapeada para uma ação do *controller* através da funcionalidade *Routing* do ASP .NET, que é utilizada pelo ASP .NET MVC *framework* para rotear requisições para as ações do *controller*. O *Routing* utiliza o Route Table para manipular as requisições que chegam. Esta tabela de rotas é criada quando a aplicação roda pela primeira vez.

A rota padrão atende praticamente todas as necessidades, mas não dispensa a implementação de rotas customizadas.

2.3 Versão atual

Atualmente está disponível a versão ASP .NET MVC 3, que é totalmente compatível com o ASP .NET MVC 2. Dentre as novidades desta versão,

destacam-se: Razor View Engine, Dynamic View e View Model, Master Pages com Razor e ViewStart. Ela pode ser utilizada tanto no Visual Studio 2010 como na versão Express.

3 CONCLUSÃO

As opiniões sobre o uso do ASP .NET MVC ainda são divergentes, mas suas vantagens são claras, algumas como controle total sobre o html gerado, separação limpa de responsabilidades, permite Test Driven Development (TDD), simplicidade na manutenção do sistema

Não pode ser deixado de citar que o WebForm também possui suas vantagens, mas é válido o estudo e pesquisa no ASP .NET MVC para projetos grandes e complexos, para facilitar e agilizar a gerência.

O mais indicado é que seu uso seja aplicado em novos projetos, e que estes não dependam de aplicações existentes em Web Forms, apesar da Microsoft garantir a existência harmoniosa dos dois tipos de aplicação, pois os conflitos existem, desde a concepção do projeto, até o modo de pensar e praticar o desenvolvimento.

REFERÊNCIAS BIBLIOGRÁFICAS

PRESSMAN, Roger S. **Engenharia de Software**. 6. ed. Porto Alegre: McGraw-Hill, Bookman, AMGH, 2010;

ARAÚJO, Carlos. **Criando aplicações com MVC**: Java Magazine. Disponível em : <http://www.devmedia.com.br/articles/viewcomp_forprint.asp?comp=18021>. Acesso em: 09 Ago. 2011;

MVC: o que é?! Disponível em: <<http://www.htmlstaff.org/ver.php?id=4500>>. Acesso em: 09 Ago. 2011;

QUINTEIRO, Tobias. **Abordando a arquitetura MVC, e Design Patterns**: observer, composite, strategy. Disponível em: <<http://pt.scribd.com/doc/46429964/Abordando-a-Arquitetura-Mvc>>. Acesso em: 10 Ago. 2011.