



Padrão de estrutura de projetos MVC4.Net

Revisões

Autor	Data
Renan Moreira Rotondo	23/09/2013
Renan Moreira Rotondo	21/01/2014

Sumário

Introdução	4
Estrutura básica.....	4
Regras de nomenclatura.....	4
Estrutura de pastas.....	6

Introdução

Esta documentação foi criada para ser a norteadora da criação/reorganização de projetos de aplicativos MVC4.Net criados na webAula. A ideia é que ela se torne uma referência para o dia-a-dia dos profissionais do setor, com o objetivo de aperfeiçoar o serviço e tornar mais simples o entendimento das estruturas de projeto dos aplicativos da empresa.

Estrutura básica

A base do projeto será pautada em uma estrutura modular. **Módulo** é a parte do sistema responsável por uma tarefa bem definida e que pode ser acoplado a um sistema para permitir ao mesmo executar a tarefa disponibilizada pelo módulo.

Módulo é uma parte do sistema que utiliza a mesma arquitetura tecnológica do sistema, é responsável por atividades que satisfaz um assunto bem definido, as atividades do módulo utilizam tarefas e componentes comuns do sistema, um módulo ou vários módulos compõem um sistema. Um módulo também é representado por um grupo de componentes de software que atende a um assunto bem definido.

O módulo realiza um assunto no processo de informação que o sistema propõe atender.

Exemplo: imagine um site que possui 3 tipos de recurso: Fórum, Banner e Calendário de eventos. Cada um desses recursos representará um módulo na estrutura do projeto. Em um projeto MVC4, os módulos serão representados por AREAS dentro do projeto.

Regras de nomenclatura

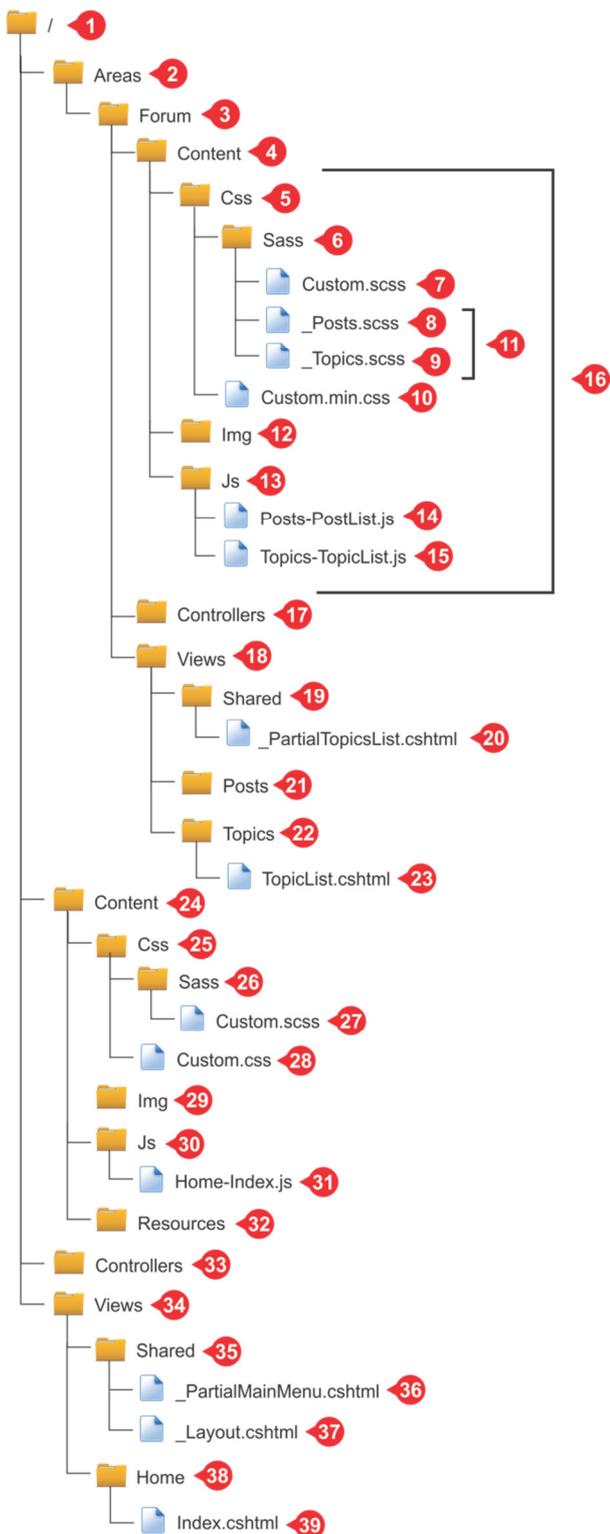
Algumas premissas para a criação e organização projetos MVC4.NET deverão ser seguidas. São elas:

- A nomenclatura dos arquivos deve seguir o padrão Upper CamelCase para as junções de palavras (primeira letra da palavra maiúscula, primeira letra das palavras anexadas maiúscula). <http://pt.wikipedia.org/wiki/CamelCase>
- A nomenclatura das classes Css devem seguir o padrão Lower camelCase (primeira letra da palavra minúscula, primeira letra das palavras anexadas maiúscula)..
- Os arquivos, pastas e estilos de cascata (CSS) deverão ter nome em inglês (Ex: 'ListTopics.cshtml', 'TopicDetail.cshtml', '.boxCalendar', Modules > Authentication)
- Evite manter arquivos de backup na estrutura do projeto no repositório de arquivos. Esses arquivos acabam se transformando em lixo e são replicados para os clientes (Ex: listTopics____.cshtml, listTopics_v2.cshtml).
- Evite manter comentadas estruturas HTML que não serão aproveitadas no código. Lembre-se que para backup de informações antigas, temos um repositório de arquivos com o histórico de versões.

- Os nomes dos arquivos devem ser facilmente identificáveis e devem explicitar a sua real funcionalidade. (Ex: tela de visualizar dados de um usuário = `UserView.cshtml`). Evite criar arquivos com nomes confusos ou abstratos (Ex: `tela1.cshtml`).
- Todos os arquivos do tipo partial deverão começar com a nomenclatura `_Partial`. (Ex: `_PartialMainMenu.cshtml`)
- Sempre utilize nomenclaturas sem acentos, espaços ou caracteres especiais.
- Sempre atribua um ID específico para containers que envolvam blocos de código que contenham um grupo de informações, uma área ou uma funcionalidade. (Ex: `#header`, `#content`, `#footer`, `#boxFeaturedNews`). Com isso, fica mais fácil criar especificidades nos códigos CSS.
- Sempre trabalhar os CSSs de modo que eles possuam alto grau de especificidade. (Ex: `#content .nav {color: #ff0000}; .message.active {background: black}`). Assim, você evita que ocorra de declarações CSS genéricas atrapalharem áreas do site que não deveriam considerá-las.
- Os javascripts serão separados sendo pareados pelo arquivo `.cshtml`. Desta forma, um arquivo dentro de `Views/Upload/View.cshtml` terá seu javascript em `Views/Js/Upload-View.js` para que seja de fácil localização o js de cada página.

Estrutura de pastas

A estrutura de pastas da do projeto seguirá o modelo esquemático disposto na figura abaixo:



Legenda:

- 1) **Pasta Raiz do projeto**
- 2) **Areas:** Pasta onde ficarão alocados os módulos do projeto. (Ex: Na imagem, foi inserido 1 exemplo de módulo, nomeado Fórum)
- 3) **Fórum:** Pasta de módulo do projeto
- 4) **Content:** Pasta onde estarão armazenados os conteúdos de CSS/Imagens/Javascrpts da área/módulo correspondente
- 5) **Css:** Pasta onde ficarão armazenados os arquivos CSS do módulo Fórum. Poderão haver arquivos com extensão Sass ou scss (<http://sass-lang.com/>), caso esse tipo de tecnologia esteja sendo utilizado.
- 6) **Sass:** Pasta onde estarão as declarações CSS específicas do módulo. Cada Css é separado de acordo com as Views do módulo.
- 7) **Custom.scss:** Arquivo Sass com os estilos gerais do módulo Fórum.
- 8) **_Posts.scss:** Arquivo Sass com os estilos da View 'Posts' do módulo Fórum.
- 9) **_Topics.scss:** Arquivo Sass com os estilos da View 'Topics' do módulo Fórum.
- 10) **Declarações CSS:** Cada View de módulo deverá ter suas declarações CSS específicas separadas, conforme exibido na imagem.
- 11) **Custom.min.css:** Arquivo minificado e compilado, gerado dinamicamente pelo Sass. Nos projetos que não utilizarem Sass, esse será o arquivo que fará o armazenamento das declarações CSS nativas do módulo.
- 12) **Img:** Pasta onde ficarão armazenadas as imagens específicas, que serão utilizadas no apenas módulo Fórum.
- 13) **Js:** Pasta onde ficarão armazenados os métodos JavaScript padrão do módulo Fórum. (Ex: funções JavaScript genéricas, usadas apenas no módulo Fórum). Além dessas, serão armazenados os JavaScripts exclusivos das Views e PartialViews do módulo. Nesse último caso, os arquivos deverão vir adicionados com o prefixo do nome da View, como demonstrado na imagem.
- 14) **Posts-PostList.js:** Arquivo JavaScript com os métodos utilizados na página PostList.cshtml, da View 'Posts' do módulo Fórum.
- 15) **Topics-TopicList.js:** Arquivo JavaScript com os métodos utilizados na página TopicList.cshtml, da View 'Topic' do módulo Fórum.

- 16) Estrutura de pastas e arquivos de um módulo:** Essa estrutura deve se repetir em todos os módulos do projeto
- 17) Controllers:** Pasta de armazenamento de controles da área/módulo
- 18) Views:** Views do módulo Fórum
- 19) Shared:** Pasta destinada a alocação dos PartialViews do módulo Fórum
- 20) _PartialTopicsList.cshtml:** PartialView da View Tópicos do módulo Fórum. As PartialViews devem ser criadas iniciando-se com a nomenclatura _Partial+nome.
- 21) Posts:** View 'Post' do módulo Fórum
- 22) Topics:** View 'Topics' do módulo Fórum
- 23) TopicList.chnml:** Arquivo de página da View 'Topics' do módulo Fórum
- 24) Content:** os conteúdos de CSS/Imagens/Javascrpts genéricos do sistema como um todo
- 25) Css:** Pasta onde ficarão armazenados os arquivos CSS genéricos do projeto. (Ex: estilos referentes ao cabeçalho e rodapé do site). Poderão haver arquivos com extensão Sass (<http://sass-lang.com/>), caso esse tipo de tecnologia esteja sendo usado.
- 26) Sass:** Pasta onde estarão as declarações CSS genéricas do projeto. (Ex: estilos do cabeçalho e do rodapé).
- 27) Custom.scss:** Arquivo com as declarações CSS genéricas do projeto (Ex: estilos do cabeçalho e do rodapé).
- 28) Custom.min.css:** Arquivo minificado e compilado, gerado dinamicamente pelo Sass. Nos projetos que não utilizarem Sass, esse será o arquivo que fará o armazenamento das declarações JavaScript genéricas do projeto.
- 29) Img:** Pasta onde ficarão armazenadas as imagens genéricas do projeto. (Ex: logomarca, ícones, etc.).
- 30) Js:** Pasta onde ficarão armazenados os métodos JavaScript genéricos do projeto. (Ex: funções JavaScript genéricas, plug-ins genéricos, usados em todo o projeto). Além dessas, serão armazenados os JavaScripts exclusivos das Views e PartialViews genéricas do projeto. Nesse último caso, os arquivos deverão vir adicionados com o prefixo do nome da View, como demonstrado na imagem.
- 31) Home-Index.js:** Arquivo JavaScript com os métodos utilizados na página Index.cshtml, da View 'Home' do projeto.
- 32) Resources:** pasta que armazena os recursos e aplicativos de terceiros de grande porte, usados no projeto (Ex: CKEditor).

33)

34) Views: Pasta onde ficarão armazenados os arquivos e Views genéricos do projeto. (Ex: PartialViews de cabeçalho, rodapé e demais PartialViews genéricas que não tem ligação direta com módulos específicos).

35) Shared: Pasta destinada a alocação dos PartialViews genéricos do projeto (Ex: PartialViews de cabeçalho, rodapé e demais PartialViews genéricas que não tem ligação direta com módulos específicos). Nesta pasta serão inseridos também os layouts do tipo Masterpage do projeto.

36) _PartialMainMenu.cshtml: PartialView de um menu genérico do projeto.

37) _Layout.cshtml: Página mestre (Masterpage) do projeto

38) Home: View 'Home' projeto.

39) Index.cshtml: Arquivo de página principal do projeto