



Padrão de estrutura de projetos ASP

Revisões

Autor	Data
Renan Moreira Rotondo	17/07/2013
Renan Moreira Rotondo	25/09/2013

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 ASP 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.

Regras de nomenclatura

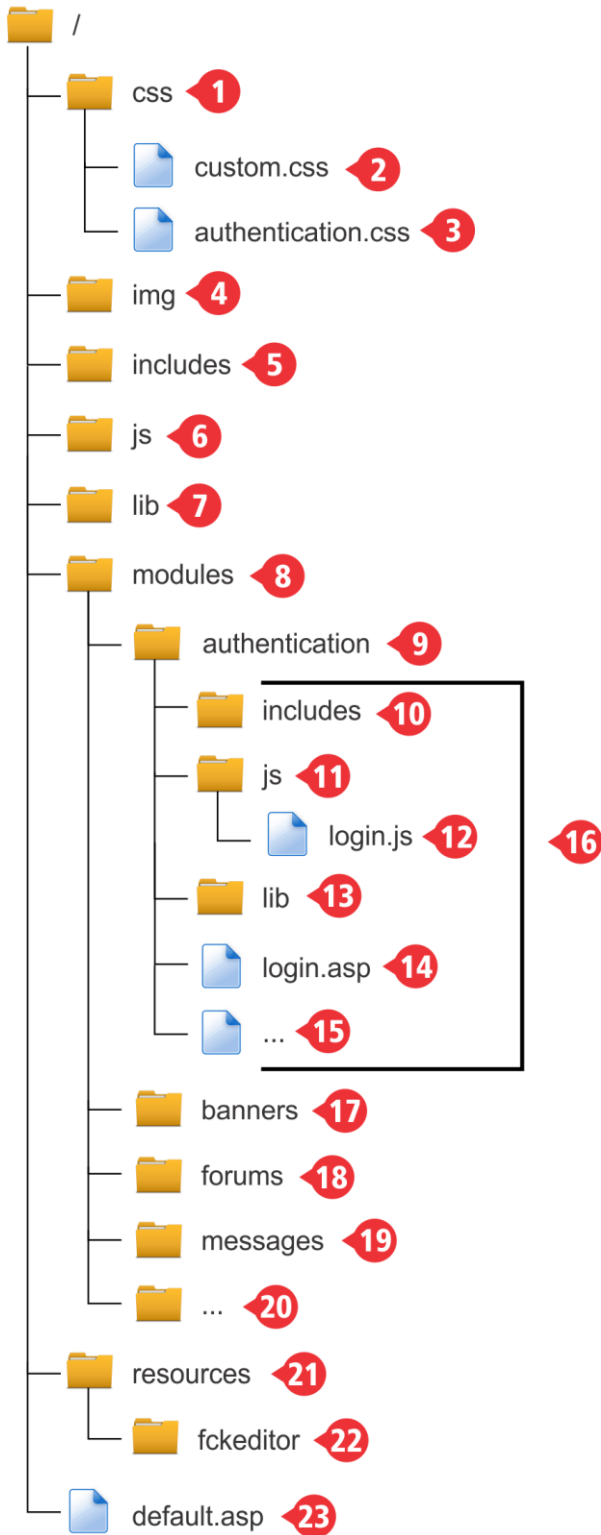
Algumas premissas para a criação e organização projetos ASP deverão ser seguidas. São elas:

- A nomenclatura dos arquivos deve seguir o padrão lower camelCase para as junções de palavras (primeira letra da palavra minúscula, primeira letra das palavras anexadas maiúscula). <http://pt.wikipedia.org/wiki/CamelCase>
- Os arquivos, pastas e estilos de cascata (CSS) deverão ter nome em inglês (Ex: 'listTopics.asp', 'topicDetail.asp', '.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____.asp, listTopics_v2.asp)
- Os nomes dos arquivos devem ser facilmente identificáveis e devem explicitar a sua real funcionalidade. (Ex: tela de visualizar dados de um usuário = userView.asp).
- Sempre utilize nomenclaturas sem acentos, espaços ou caracteres especiais.

- Atribuir um ID específico para blocos de código que contemplem 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})

Estrutura de pastas

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



Legenda:

- 1) **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. As declarações CSS pertencentes a um determinado módulo, deverão estar separadas das declarações genéricas presentes no arquivo custom.css.
- 2) **custom.css:** Arquivo com os estilos básicos do projeto.
- 3) **authentication.css:** Exemplo de arquivo com os estilos do módulo denominado authentication.
- 4) **img:** Pasta onde ficarão armazenadas as imagens genéricas do projeto. (Ex: logomarca do cliente, plano de fundo do corpo da página, etc.)
- 5) **includes:** Pasta onde ficarão armazenados os arquivos de controle padrão do projeto. (Ex: include do menu principal, include do rodapé principal, include da biblioteca webAula Standards – <http://wastandards.webaula.com.br> , etc.)
- 6) **js:** Pasta onde ficarão armazenados os métodos javascript padrão do projeto. (Ex: funções javascript genéricas, usadas em todo o projeto, biblioteca jquery, etc.).
- 7) **lib:** Pasta onde ficarão armazenadas as bibliotecas ASP genéricas do projeto. (Ex: classes de entidades, métodos e usados em todo o projeto. livrarias de código, etc.)
- 8) **modules:** Pasta onde ficarão estocados os módulos do projeto. (Ex: Na imagem, foram inseridos 4 exemplos – módulo de autenticação, módulo de banners, módulo de fóruns, módulo de mensagens)
- 9) **pasta de módulo:** Pasta de um módulo em específico
- 10) **includes:** Pasta com os arquivos de inclusão exclusivos do módulo. (Ex: includes de boxes e de widgets do módulo).
- 11) **js:** Pasta com os arquivos javascript exclusivos das telas utilizadas pelo módulo. Cada tela terá um arquivo javascript correspondente. Esse javascript deverá ter o nome da tela ao qual ele se refere.
- 12) **login.js:** Arquivo javascript da tela de login, usada como exemplo na imagem.
- 13) **lib:** Pasta onde ficarão armazenadas as bibliotecas ASP exclusivas do módulo. (Ex: classes de entidades, métodos. livrarias de código, etc.)
- 14) **login.asp:** Arquivo de tela modelo do módulo em questão.
- 15) **...:** Demais arquivos de tela do módulo.

- 16) Estrutura de pastas e arquivos de um módulo:** Essa estrutura deve se repetir em todos os módulos do projeto
- 17) pasta de módulo:** Pasta de um módulo em específico
- 18) pasta de módulo:** Pasta de um módulo em específico
- 19) pasta de módulo:** Pasta de um módulo em específico
- 20) demais pastas de módulos do projeto.**
- 21) resources:** pasta que armazena os recursos e aplicativos de terceiros de grande porte, usados no projeto
- 22) fckeditor:** Exemplo de recurso
- 23) default.asp:** Página inicial do projeto