VISUAL STUDIO 2015 TIPS AND TRICKS

Aqui você irá aprender a navegar mais facilmente pelo Visual Studio 2015

Aprendendo o caminho das pedras

Sumário

[IDE 2](#_Toc440966740)

[Editor 4](#_Toc440966741)

[Debugging 6](#_Toc440966742)

[Snippets 7](#_Toc440966743)

[Outras Dicas 8](#_Toc440966744)

[Criando Atalhos 8](#_Toc440966745)

[Comentando Sessões de Debugging 8](#_Toc440966746)

# IDE

**Roaming Settings**: É necessário logar com seu perfil no VS2015. Assim você poderá configurar seu ambiente, subir ele para a nuvem, e, posteriormente, recuperá-lo em uma outra máquina. Caminho:

TOOLS > OPTIONS > ENVIRONMENT > SYNCHRONIZED SETTINGS

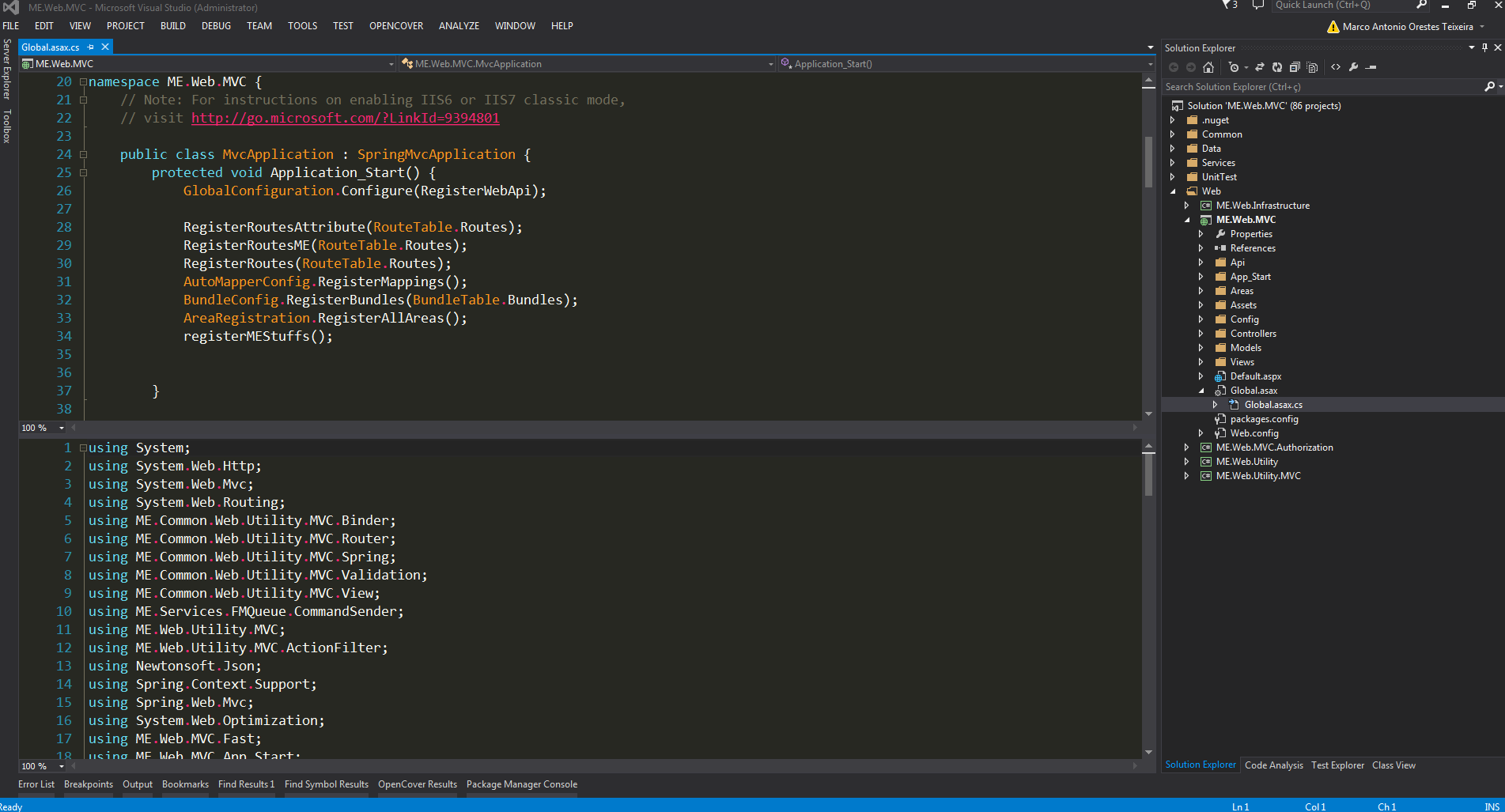
**Dock & Undock**: Quando você quer “desplugar / plugar” uma aba do VS2015.

CTRL + “DOUBLE CLICK”

**Multiple Drag & Drop Tabs**: É possível selecionar múltiplas abas de código (ou texto) de uma única vez e arrastá-las para outro lugar.

CTRL + CLICK, nas abas desejadas

**Split Tab View**: Na aba de código desejada, clique e arraste o ícone  de divisão da janela. Assim você poderá ter duas visualizações para uma mesma aba.



**Quick Search**: Procura por todos os recursos do VS2015 e também pela sua solução. É possível utilizar tags especiais para filtrar, basta iniciar a pesquise com o caractere “@”, seguido da opção desejada.

CRTL + Q

**Inline Search**: Esse recurso é prático para encontrar arquivos dentro de uma solução, ou, até mesmo, trechos de código.

CRTL + ,

**Open Immediate Window**: Abre a janela imediata, muito útil para substituir valores de variáveis on-the-fly.

CTRL + ALT + I

**Go to Solution Explorer**: Abre o Solution Explorer, caso não esteja, e coloca o cursor sobre o último item selecionado.

CTRL + SHIFT + L

**Add New Item**: Adiciona um novo item ao projeto.

CTRL + SHIFT + A

Nota: Esse comando pode ser associado com o **Go to Solution Explorer**

# Editor

**Move Code (Up-Down)**: Move o código selecionado (ou apenas a linha onde o cursor estiver parado) para cima ou para baixo.

ALT + ARROW UP / ARROW DOWN

**Collapse Region**: Diferente da tag “region”, esse comando cria regiões colapsáveis no código selecionado.

Criar: CTRL + M + H

Remover: CTRL + M + U

**Comment Block**: Comenta um bloco de código.

Comentar: CTRL + K + C

Descomentar: CTRL + K + U

**Pick Definition**: Ao invés de utilizar o famoso F12, você pode mostrar, em uma pequena janela, a definição de um método. Selecione (ou deixe o cursor em cima do método) e clique:

ALT + F12

**Find All References**: Encontra todas as referências para o dado método. Basta deixar o cursor em cima do método e clicar:

SHIFT + F12

**View Call Hierarchy**: Visualiza toda árvore hierárquica de uma chamada. Tanto para cima como para baixo:

CTRL + K, CTRL + T

**Navigate Forward / Backward**: Utilizando o F12, podemos navegar de definição em definição. Mas, para voltar atrás no histórico, basta utilizar:

Para frente: CTRL + -

Para trás: CTRL + SHIFT + -

**Toggle Bookmark**: Adiciona, ou remove, uma marca de referência ao local do cursor.

CTRL + K + K

**Bookmark Previous / Next**: Navega pelas marcas de referência adicionadas.

Anterior: CTRL + K, CTRL + P

Próxima: CTRL + K, CTRL + N

**Show Open Files**: Exibe uma lista de arquivos abertos.

CTRL + ALT + ARROW DOWN

**Close Current Tab**: Fecha a aba atual.

CTRL + F4

**Remove All Breakpoints**: Remove todos os breakpoints de uma única vez.

CTRL + SHIFT + F9

**Format Document**: Formata o documento que você estiver editando, dado suas características de formação. Vale para código, HTML, SQL Scripts, XML, etc.

CTRL + K, CTRL + D

**Format Selection**: Quase igual ao **Format Document**, porém formata apenas o código que foi selecionado.

CTRL + K, CTRL + F

**Go to Line**: Útil quando você quer ir para uma linha do documento.

CTRL + G

# Debugging

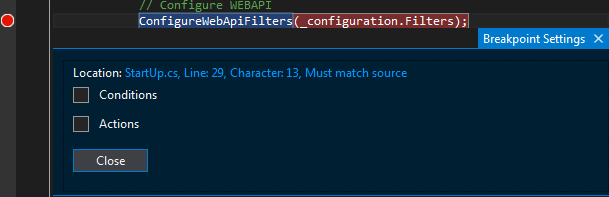
Esses são os principais comandos, e podem ser muito úteis para agilizar o processo de debugging.

Definir breakpoint: F9, para adicionar, F9, novamente, para remover.

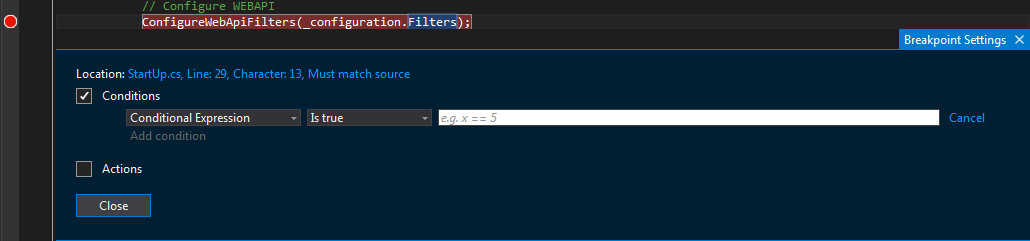
Remover todos os breakpoints: CTRL + SHIFT + F9

Editar configurações condicionais do breakpoint: ALT + F9, C

**Dica**: é possível definir uma condição para um breakpoint ocorrer. Quando definir um breakpoint, chame a janela de configuração condicional do breakpoint:



Selecione o checkbox “Conditions” e adicione a condição desejada, no caso, uma expressão que possa ser avaliada para um boolean:



Existem outras formas para condicionais em breakpoints. Explorer-as!

Iniciar o debugging: F5

Parar o debugging: SHIFT + F5

Reiniciar: CTRL + SHIFT + F5

Passar por um método: F10

Entrar em um método: F11

Sair do método (uma vez que entrou) e voltar de onde começou: SHIFT + F11

Ir até o cursor: CTRL + F10. Esse recurso é bem interessante. Ele inicia o debugging onde o cursor estiver posicionado.

# Snippets

Snippets são pequenos trechos que código que expandem para criar um trecho maior. Existem vários snippets built-in no VS2015, um deles é o “ctor” que cria um construtor sem parâmetros na classe que você estiver editando. Para inseri-los, basta digitar suas iniciais e clicar TAB + TAB.

Segue uma lista dos quais acredito serem os mais úteis:

ctor: Criar construtor

try: Cria um bloco instrução try com cláusula catch.

tryf: Cria um bloco de instrução try apenas com cláusula finally.

Attribute: Cria uma classe que extende de Attribute, completa.

Exception: Idem a Attribute, porém com a classe Exception.

prop: Cria uma propriedade automática, pública.

propfull: Cria uma propriedade completa, com campo backing field privado.

foreach: Cria um bloco de instrução laço foreach.

for: Cria um bloco de instrução laço for.

forr: Cria um bloco de instrução laço for, reversa.

while: Cria um bloco de instrução laço while.

do: Cria um bloco de instrução laço do.

switch: Cria um bloco de instrução switch.

lock: Cria um bloco de instrução lock.

~: Cria um bloco de instrução destructor.

Mas também é possível criar seus próprios snippets, basta seguir esse tutorial [Walkthrough: Creating a Code Snippet](https://msdn.microsoft.com/en-us/library/ms165394.aspx) (https://msdn.microsoft.com/en-us/library/ms165394.aspx)

# Outras Dicas

## Criando Atalhos

Alguns atalhos não estão disponíveis no VS2015 para coisas que podem ter uma valor na organização do código. Um desses exemplos é organizar e remover “using” desnecessários. Nesses casos você pode criar seu próprio atalho para um comando. Para fazer isso, acesse as opções de teclado nas opções do VS2015. (ALT + T + O)

Na tela de opções, use a caixa de busca (CTRL + E) e procure por “Keyboard”.

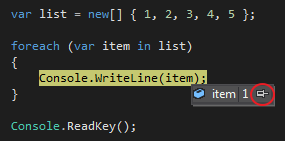
Na tela de configuração de teclado, selecione o **Scheme** “Default” para o teclado. Em “Show commands containing:” digite “RemoveAndSort”. Isso irá procurar pelo comando que desejamos. O comando é “EditorContextMenus.CodeWindow.OrganizeUsings.RemoveAndSort”, selecione-o e vá a caixa “Use new shortcut in:” (ALT + N) e selecione **Text** **Editor**. Agora, na caixa “Press shortcut key” (ALT + P), escolha o shortcut que você deseja para esse comando, por exemplo CTRL + ‘. Nota: Eu prefiro esse comando pois fica bem próximo da mão esquerda, e é fácil de decorar.

Caso o shortcut já esteja sendo utilizado por outro comando, ele irá aparecer na caixa “Shortcut current used by:”. Escolha outro comando, para evitar colisões ou sobrescrita de atalhos.

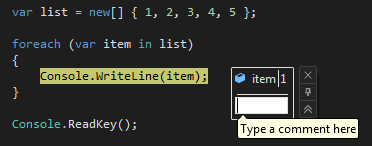
Feito isso, clique em **Assign** (ALT + S). Depois em **OK** para fechar a janela. Esse tipo de ação vale para todos os tipos de novos atalhos que você deseja criar no futuro.

## Comentando Sessões de Debugging

No VS2015 é possível criar comentários (persistentes) para sessões de debugging. Para fazer isso, passe o cursor do mouse sobre a variável (ou trecho de código) que deseja inspecionar e clique no ícone de “pin”:



Isso irá travar a inspeção do código, feito isso você poderá adicionar comentários relativos a observação do debugging.



Mesmo que você finalize a sessão de debugging, esses comentários ainda ficarão presentes para a próxima sessão, assim que você passar pelo trecho em questão:

