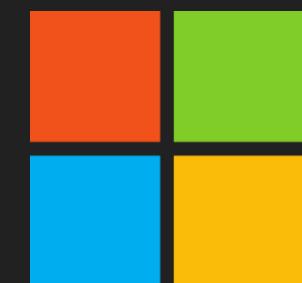


Crea aplicaciones web dinámicas con ASP.NET Core

RC1, MVC 6, Entity Framework 7
y Angular JS

Por: Karen Goytizolo Cáceres



Microsoft



AGENDA – PARTE 1



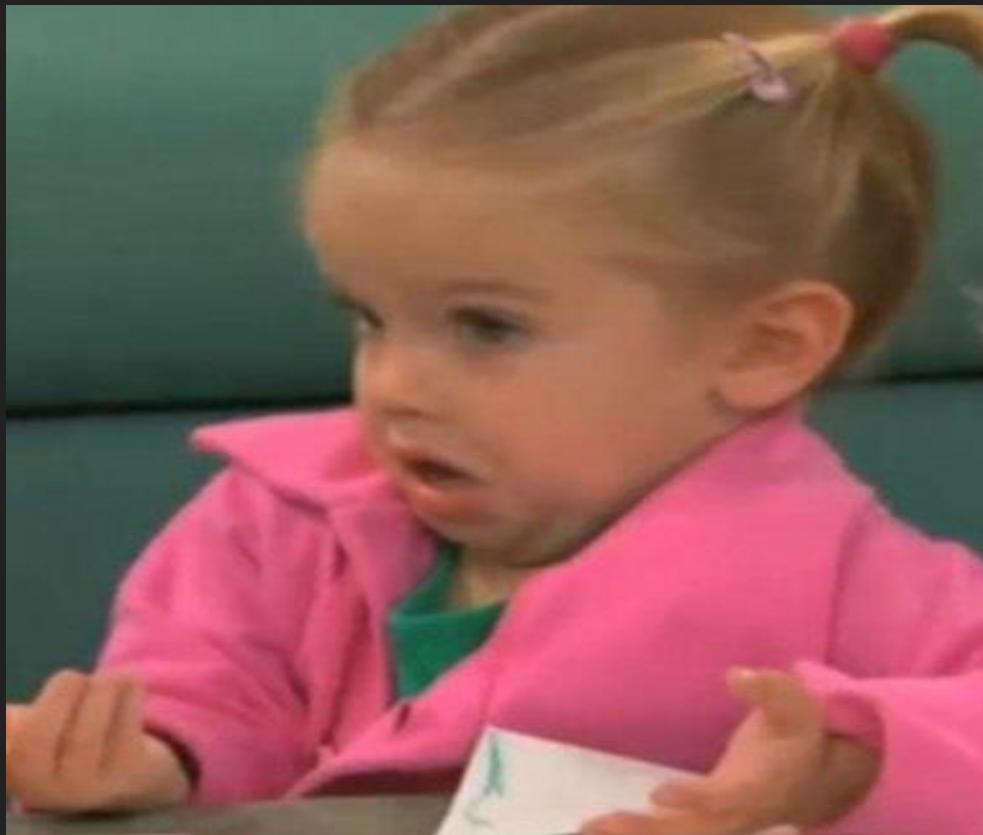
- ✓ ¿Por qué ASP.NET?
- ✓ Conociendo un poco la evolución de ASP.NET
- ✓ Principales aplicaciones web ASP.NET disponibles ¿Cuál utilizar?
- ✓ ¿Por donde empezar? Instalando ASP.NET 5 y nuevos conceptos.
- ✓ Hola Mundo! Mi primera app con ASP.NET 5 y Visual Studio Code.
- ✓ Usando aplicaciones MVC plantilla con DNX y Visual Studio Code.

AGENDA – PARTE 2



- ✓ Intellisense + Bower: Adicionando dependencias por el lado del cliente.
- ✓ Diseñando aplicaciones web y servicios web API RESTFul con MVC 6
- ✓ Aplicando Entity Framework 7 para armar tu base de datos.
- ✓ Uso de Angular JS para potenciar el lado del cliente.
- ✓ Tips y mejores prácticas para desarrollo de aplicaciones web
- ✓ Conclusiones

¿Por qué ASP.NET?



¿Por qué ASP.NET?



**Desarrollo mucho más
rápido y amigable**

- Lenguaje C#
- .NET Framework
- Fácil desarrollo mediante Visual Studio IDE
- NuGet packages – integración
- Intellisense
- Opciones de back-end multi dispositivos
- Diversas opciones de arquitectura de aplicaciones web



¿Por qué ASP.NET?



**Flexibilidad y
continua evolución**



- Multi - Lenguaje
- Aplicaciones web Multi – Plataforma.
- Uso de packaging y comandos (ASP.NET 5 +)
- Compatible con frameworks de patrones de diseño front-end
- Aplicación de patrones de diseño Back-end
- ASP.NET Identity
- Mayores opciones para Unit Testing
- A la vanguardia con tendencias de TI
- Open source – tecnología híbrida



¿Por qué ASP.NET?



Alta adopción en las empresas



- Alta demanda de desarrolladores .NET
- Entorno RAD completo para el ciclo de vida del producto SW
- Integración con otros productos de Microsoft
- Provee soluciones web con arquitectura sólida y escalable
- Más económico que IBM, Oracle y otras empresas de SW
- Mayor soporte por parte de socios Microsoft
- Buen nivel de productividad



ASP.NET – 14 años de evolución

Enero 2002
.NET 1.0
VS .NET



Abril 2003
.NET 1.1
VS .NET 2003



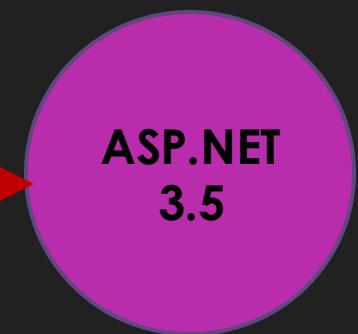
Noviembre 2005
.NET 2.0
VS 2005



Noviembre 2006
.NET 3.0
Expression Blend



Noviembre 2007
.NET 3.5
VS 2008



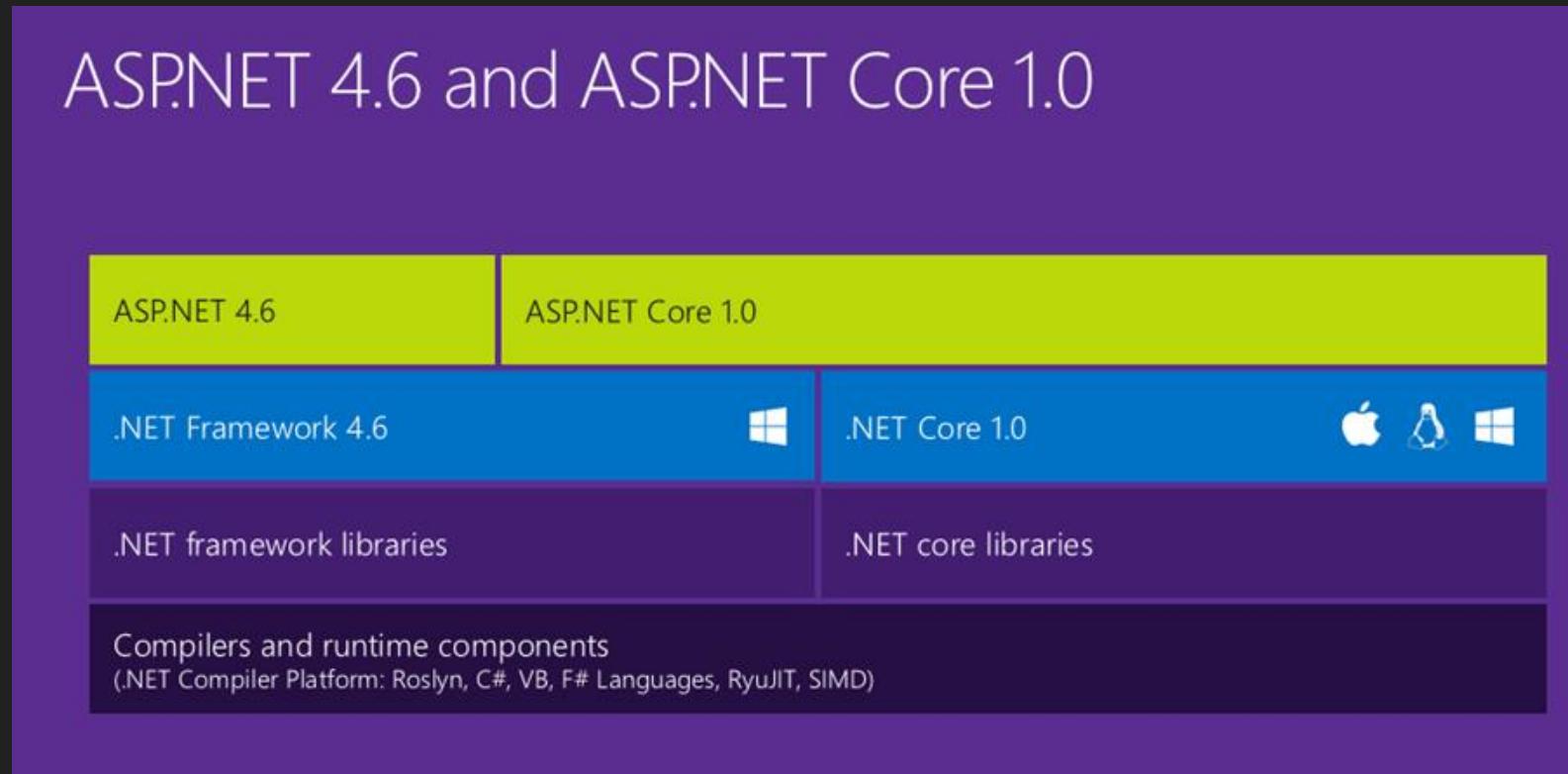
ASP.NET – 14 años de evolución

A timeline diagram illustrating the evolution of ASP.NET over 14 years. It consists of five circular nodes connected by arrows, each containing the version name and its release date, .NET framework, and Visual Studio edition.

Fecha	Version	.NET	VS
Abril 2010	ASP.NET 4.0	.NET 4.0	VS 2010
Agosto 2012	ASP.NET 4.5	.NET 4.5	VS .NET 2012
Octubre 2013	ASP.NET 4.5.1	.NET 4.5.1	VS 2013
Julio 2015	ASP.NET 4.6 y 5.0 beta	.NET 4.6 - .NET 5.0	VS 2015
Noviembre 2015	ASP.NET 4.6.1 y Core 1.0	.NET 4.6.1 .NET CORE 1.0	VS 2015 Update 1



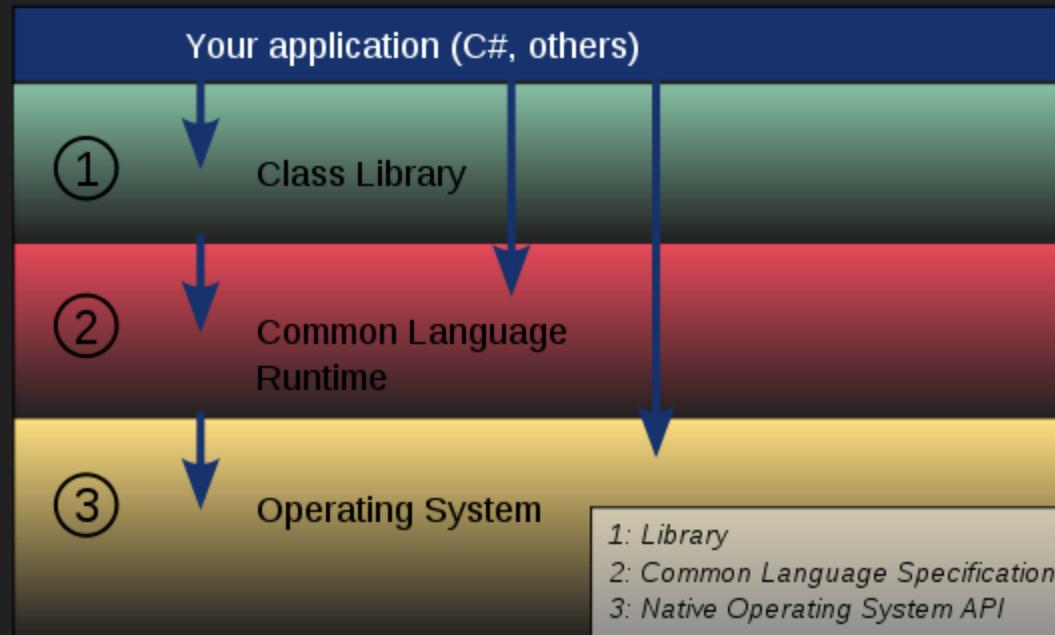
Comparando ASP.NET 4.6 y ASP.NET CORE 1.0



Fuente: <http://www.hanselman.com/blog/ASPNET5IsDeadIntroducingASPNETCore10AndNETCore10.aspx>



¿NET y ASP.NET ya funcionaban en entornos multiplataforma antes de Core 1.0?



- Mono tiene implementaciones de:
 - ASP.NET 2.0,
 - ASP.NET MVC
 - ASP.NET AJAX
- Mono Project ASP.NET soporta:
 - Web Forms
 - Web Services (basado en SOAP).
- Mono Open Source Project comienza el 07/2001
- Mono 1.0 es lanzado el 30/07/2004



Microsoft



Empezando a usar ASP .NET

Apps Web

- ASP.NET Web Forms
- ASP.NET MVC
- ASP.NET Web Pages
- ASP.NET SPA

APIs

- ASP.NET Web APIs

Tiempo-Real

- ASP.NET SignalR

Extensiones

- ASP.NET AJAX
- ASP.NET Dynamic Data



Microsoft



Trabajando con ASP.NET Web Forms

- Páginas maestras .aspx
- Code behind
- Html dinámico generado por el servidor
- Controles predefinidos de servidor <asp: >
- Manejo de eventos en los controladores
- Herramienta de diseño para UI

The screenshot shows the Microsoft Visual Studio IDE interface for developing ASP.NET Web Forms. The top menu bar includes 'File', 'Edit', 'View', 'Project', 'Server Explorer', 'Toolbox', 'Task List', 'Properties', 'Help', and 'Help Contents'. The title bar displays 'Default2.aspx.cs' and 'Default2.aspx'. The main window is divided into several panes:

- Solution Explorer:** Shows the project structure for 'D:\school\WebSite2\'. It includes a folder named 'Account' and files such as 'App_Data', 'Scripts', 'Styles', 'About.aspx', 'Default.aspx', 'Default2.aspx', 'Default2.aspx.cs', 'Global.asax', 'Site.master', and 'Web.config'.
- Properties:** A pane showing properties for the current selected item, which is a 'div' element with ID 'form1'.
- Toolbox:** A pane containing various ASP.NET controls like 'Label', 'Text Box', 'Button', etc.
- Client Objects & Events:** A pane showing the client-side code generated from the server-side code.
- Code Editor:** The bottom pane where the server-side code is written. The code for Default2.aspx.cs is as follows:

```
1  <%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default2.aspx.cs" Inhe
2
3  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.
4
5  <html xmlns="http://www.w3.org/1999/xhtml">
6  <head runat="server">
7      <title></title>
8  </head>
9  <body>
10     <form id="form1" runat="server">
11         <div style="height: 80px">
12             <asp:Label ID="Label1" runat="server" Text="Your Name: "></asp:Label>
13             <asp:TextBox ID="NameTextBox" runat="server"></asp:TextBox>
14             <br />
15             <asp:Button ID="Submit" runat="server" onclick="Submit_Click"
16                 style="z-index: 1; left: 61px; top: 57px; position: absolute" Text
17             </div>
18
19     </form>

```



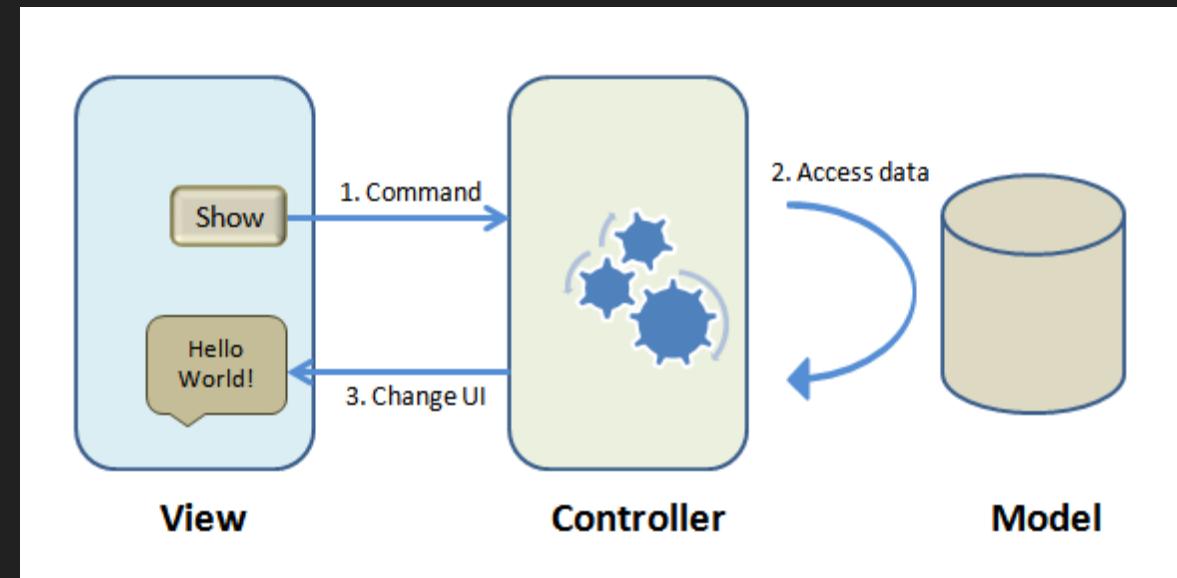
Trabajando con ASP.NET Web Forms

- Separación de UI y lógica del negocio
- URL Routing
- Lenguajes: C#, VB.NET
- Adquiere beneficios de .NET y uso de FW
- Controles data-bound
- Membership - registro de usuarios



Trabajando con ASP.NET MVC

- Patrón Modelo – Vista - Controlador
- Modelo:
 - Reglas de negocio
 - Datos devueltos / Acceso a datos
- Vista: Presentación de la aplicación
- Controlador: Intermediario entre
 - Usuario > Modelo > Vistas



Trabajando con ASP.NET MVC

- Arquitectura orientada a eventos
- URLs semánticas – URL Routing
- Lenguajes: C#, VB.NET,
- Markup language (Razor)
- Adquiere beneficios de .NET y uso de FW
- Curva de aprendizaje > Web Forms
- Mayor facilidad para pruebas unitarias

The screenshot shows the Microsoft Visual Studio IDE interface. The code editor displays the `MyMVCController.cs` file, which contains the following C# code:

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Web;
5  using System.Web.Mvc;
6  copyright@dotnet-tricks.com
7  namespace Mvc4_WebGrid.Controllers
8  {
9      public class MyMVCController : Controller
10     {
11         // GET: /MyMVC/
12         public ActionResult Index()
13         {
14             return View();
15         }
16     }
17 }
18
19
20
21
```

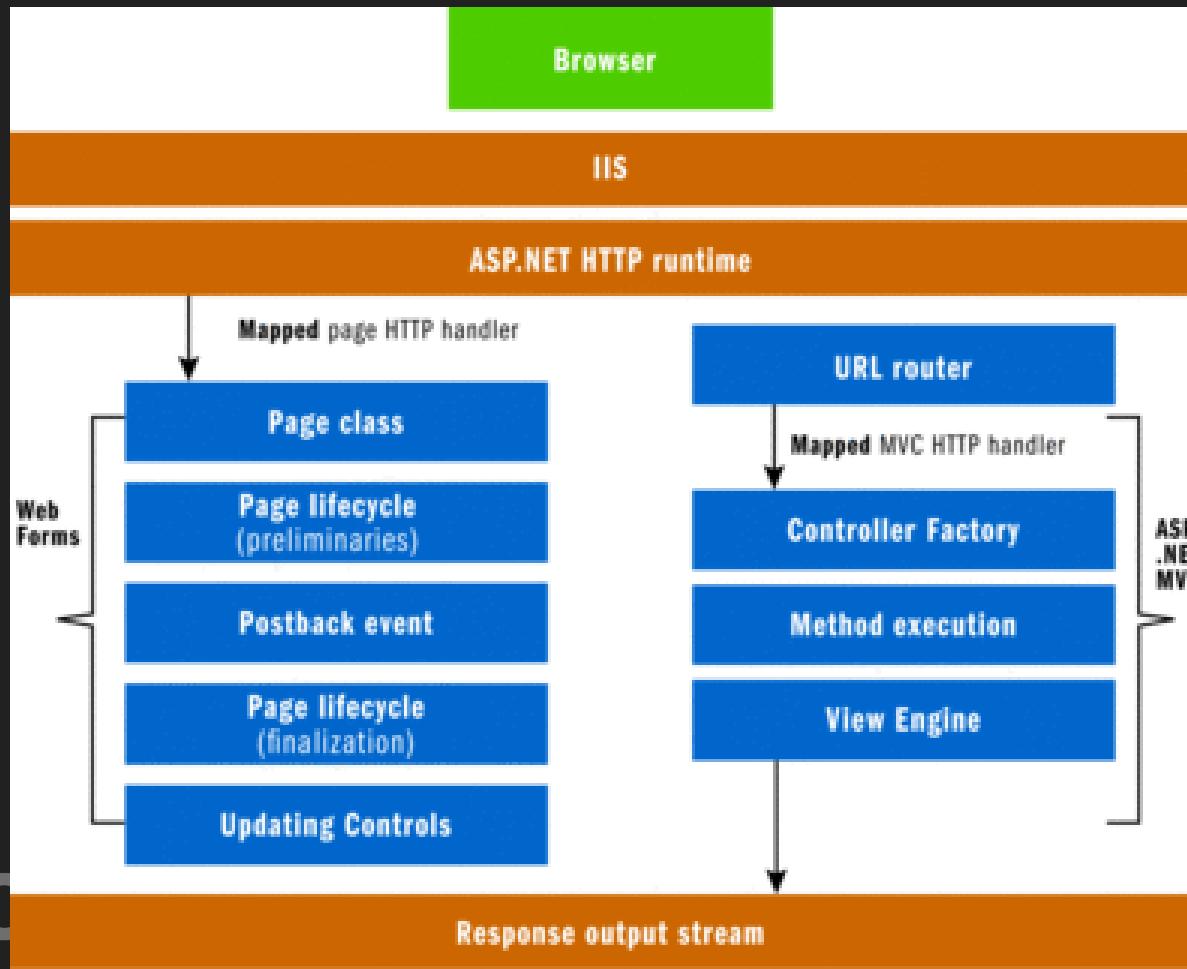
A red arrow points from the line `// GET: /MyMVC/` in the code editor to the `MyMVCController.cs` file in the Solution Explorer. The Solution Explorer pane on the right shows the project structure for 'Mvc4_WebGrid' with files like `HomeController.cs`, `MyMVCController.cs`, and `AccountController.cs`.



Microsoft

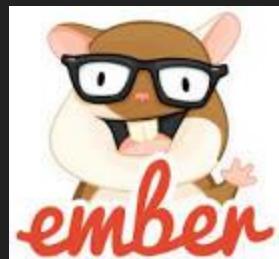
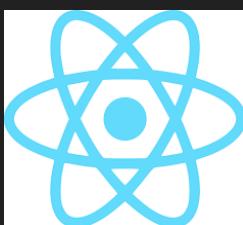


Arquitectura Web Forms vs MVC



Sinergia entre ASP.NET web applications y Opensource

Front-end

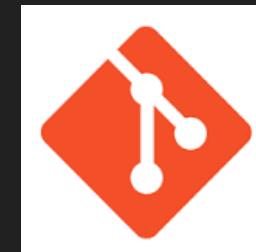


Integración NPM y manejo de Dependencias



 Microsoft

Repositorios

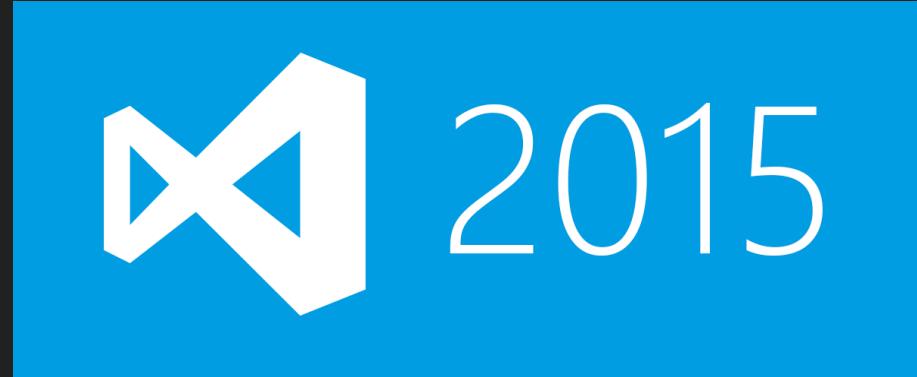


Bases de Datos No Sql

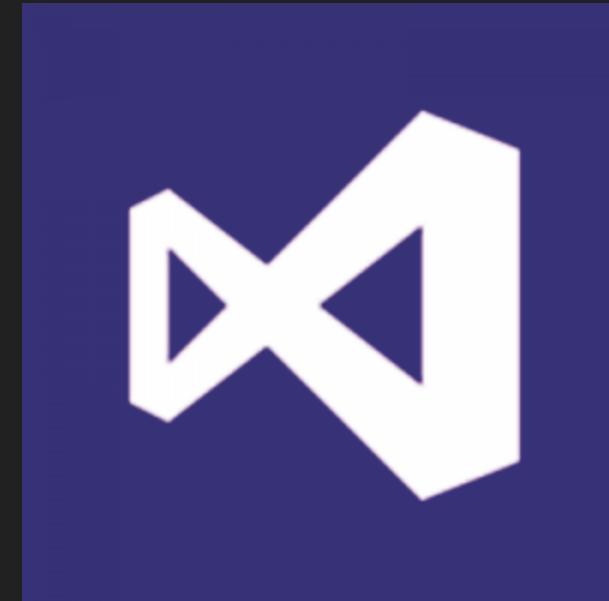


<http://nosql-database.org/>

Empezando a meter mano en el código (Demo)



Visual Studio 2015



Visual Studio Code



¿Por donde empezar? Instalando ASP.NET 5 y nuevos conceptos

1. Descargar e instalar ASP.NET 5 (.NET Core):

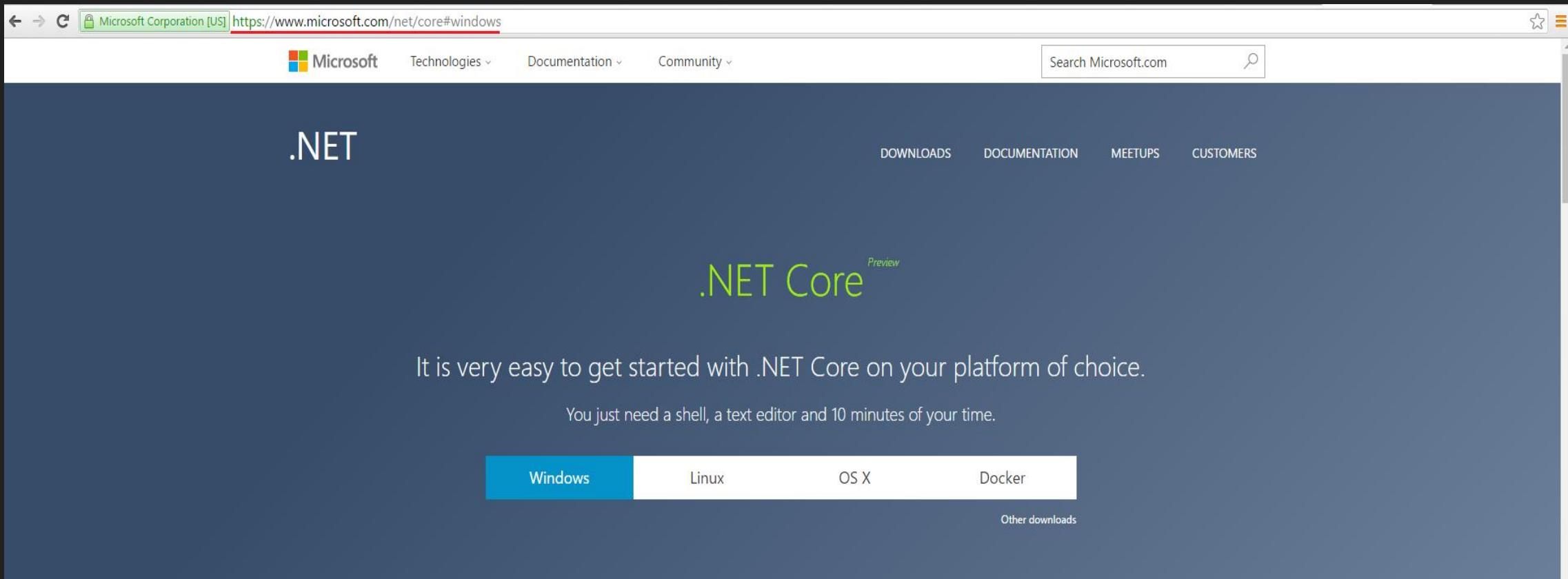
The screenshot shows a web browser displaying the Microsoft .NET website at <https://www.microsoft.com/net>. The page has a dark blue header with the Microsoft logo and navigation links. Below the header, there's a large green banner with the text "Any developer, any app, any platform". The main content area is divided into three columns:

- Windows development**: Describes the .NET Framework as a comprehensive programming model for building mobile, desktop, and web applications on Windows. It includes a "Learn more" link and a large blue "Download .NET Framework 4.6.1" button.
- Cross-platform apps**: Describes .NET Core and ASP.NET Core as a fast and modular platform for creating server applications across Windows, Linux, and Mac. It includes a "Learn more" link and a large blue "Download .NET Core" button.
- Mobile apps**: Describes Xamarin as bringing .NET power and productivity to iOS and Android, reusing skills and code. It includes a "Learn more" link and a large blue "Download Xamarin" button.

At the bottom of each column, there's a "Other downloads" link. A large circular arrow icon is located in the bottom right corner of the page.

¿Por donde empezar? Instalando ASP.NET 5 y nuevos conceptos

2. Descargar e instalar ASP.NET 5 (.NET Core) – Te selecciona OS:



The screenshot shows a web browser displaying the Microsoft .NET Core download page at <https://www.microsoft.com/net/core#windows>. The page has a dark blue header with the Microsoft logo and navigation links for Technologies, Documentation, and Community. A search bar is also present. Below the header, the word ".NET" is displayed in large white letters. In the center, ".NET Core" is shown in green with a "Preview" tag above it. A subtext states: "It is very easy to get started with .NET Core on your platform of choice. You just need a shell, a text editor and 10 minutes of your time." At the bottom, there are four download buttons: "Windows" (highlighted in blue), "Linux", "OS X", and "Docker". A "Other downloads" link is located below these buttons. On the far right, there is a large circular arrow icon.

¿Por donde empezar? Instalando ASP.NET 5 y nuevos conceptos

3. Si usas Windows, antes descarga e instala *Visual Studio Community 2015*:

The screenshot shows a web browser displaying the Microsoft .NET Core SDK download page for Windows. The URL in the address bar is <https://www.microsoft.com/net/core#windows>. The page title is "Install for Windows". A large blue circle with the number "1" contains the text "Install .NET Core SDK". Below it, the heading "Before you start" is followed by the instruction: "Remove all previous versions of .NET Core from your system by using Add/Remove programs.". Under "Visual Studio users", it says: "The best way to develop with .NET Core on Windows is to download the [Visual Studio official MSI Installer](#) and the latest [NuGet Manager extension for Visual Studio](#). If you don't have Visual Studio already, you can download [Visual Studio Community 2015](#) for free.". Under "Command line users", it says: "If you prefer to develop with the command line, you can download [the .NET Core SDK for Windows](#).". A red box highlights the "Visual Studio Community 2015" link in the "Visual Studio users" section.

Microsoft Corporation [US] <https://www.microsoft.com/net/core#windows>

Install for Windows

1 Install .NET Core SDK

Before you start

Remove all previous versions of .NET Core from your system by using Add/Remove programs.

Visual Studio users

The best way to develop with .NET Core on Windows is to download the [Visual Studio official MSI Installer](#) and the latest [NuGet Manager extension for Visual Studio](#). If you don't have Visual Studio already, you can download [Visual Studio Community 2015](#) for free.

Command line users

If you prefer to develop with the command line, you can download [the .NET Core SDK for Windows](#).

¿Por donde empezar? Instalando ASP.NET 5 y nuevos conceptos

4. Pero si usas otro OS <> Windows, puedes descargar *Visual Studio Code*:

The screenshot shows the official Visual Studio Code download page at <https://code.visualstudio.com/Download>. The page has a dark blue header with the Visual Studio Code logo, navigation links (Docs, Updates, Blog, Extensions, FAQ), a star rating (14,434), and a search bar. A yellow banner at the top states "Version 1.1 is now available! Read about the new features and fixes in April." Below the banner, the main heading is "Download Visual Studio Code" with the subtext "Free and open source. Integrated Git, debugging and extensions." There are two tabs: "Stable" (selected) and "Insiders". Under the "Stable" tab, there are download links for Windows (Windows 7, 8, 10), Linux (.deb for Debian, Ubuntu; .rpm for Red Hat, Fedora, CentOS), and OS X (OS X Yosemite, El Capitan). A red box highlights the "Visual Studio Code es la herramienta multiplataforma otorgada por Microsoft" text in the sidebar. At the bottom left, a download progress bar shows "VSCodeSetup-stable.exe 5.0/29.2 MB, 59 secs left". A large circular arrow icon is in the bottom right corner.

Visual Studio Code es la herramienta multiplataforma otorgada por Microsoft

Stable Insiders

Windows

.deb .rpm

OS X

VSCodeSetup-stable.exe 5.0/29.2 MB, 59 secs left

¿Por donde empezar? Instalando ASP.NET 5 y nuevos conceptos

5. Descargar e instalar ASP.NET 5 (.NET Core) - versión actual RC2 :

The screenshot shows a web browser displaying the Microsoft .NET Core SDK download page for Windows. The URL in the address bar is <https://www.microsoft.com/net/core#windows>. The page title is "Install for Windows". A large red callout box on the left contains the instruction: "* Asegúrate que tengas instalado dentro de tu Visual Studio 2015 el Update 2 en adelante." (Make sure you have Update 2 installed in your Visual Studio 2015). The main content area is divided into two sections: "1 Install .NET Core SDK" and "2 Initialize some code". The "1" section includes steps for Visual Studio users (downloading the Visual Studio official MSI Installer and Visual Studio Community 2015) and Command line users (downloading the .NET Core SDK for Windows). The "2" section shows a terminal window with the commands "mkdir hwapp" and "cd hwapp". At the bottom left, there is a download progress bar for "DotNetCore.1.0.0.RC....exe". A large red arrow icon is located at the bottom right.

* Asegúrate que tengas instalado dentro de tu Visual Studio 2015 el Update 2 en adelante.

1

Install .NET Core SDK

Before you start

Remove all previous versions of .NET Core from your system by using Add/Remove programs.

Visual Studio users

The best way to develop with .NET Core on Windows is to download the [Visual Studio official MSI Installer](#) and the latest NuGet Manager extension for Visual Studio. If you don't have Visual Studio already, you can download [Visual Studio Community 2015](#) for free.

Command line users

If you prefer to develop with the command line, you can download [the .NET Core SDK for Windows](#).

2

Initialize some code

Let's initialize a sample Hello World application!

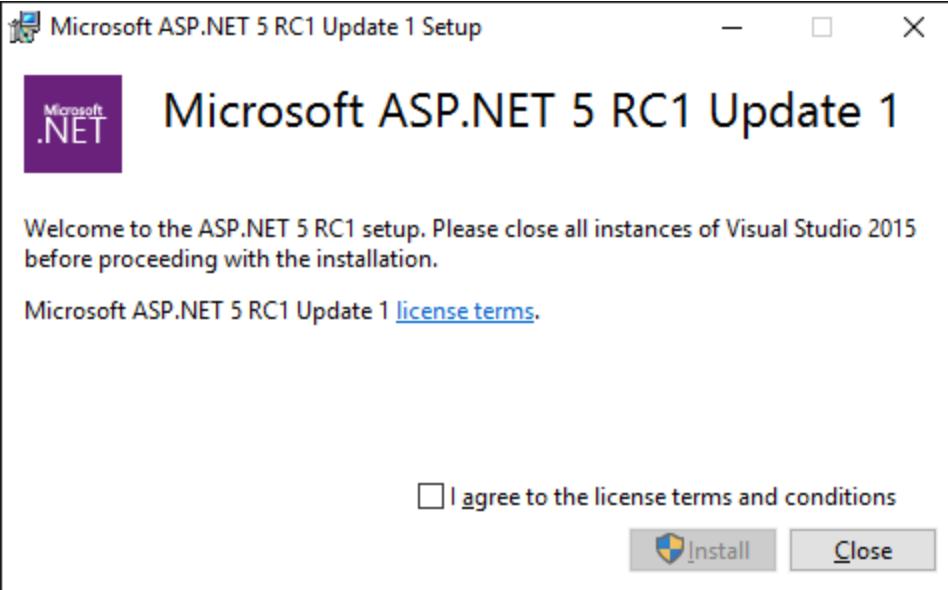
```
mkdir hwapp
cd hwapp
```

DotNetCore.1.0.0.RC....exe

Show all downloads...

¿Por donde empezar? Instalando ASP.NET 5 y nuevos conceptos

6. Descargar e instalar ASP.NET 5 (.NET Core) – versión anterior RC1:



The screenshot shows the Microsoft ASP.NET 5 RC1 Update 1 Setup window. The title bar reads "Microsoft ASP.NET 5 RC1 Update 1 Setup". The main content area says "Microsoft ASP.NET 5 RC1 Update 1" and "Welcome to the ASP.NET 5 RC1 setup. Please close all instances of Visual Studio 2015 before proceeding with the installation." It includes a "Microsoft Web Developer Tools" checkbox and a "I agree to the license terms" checkbox. At the bottom are "Install" and "Close" buttons.

* Optar por descargar este instalador para trabajar con la versión RC1 aplicada a este ejemplo

https://docs.asp.net/en/1.0.0-rc1/getting-started/installing-on-windows.html#install-asp-net-5-with-visual-studio

Microsoft ASP.NET 5 RC1 Update 1 Setup

Microsoft ASP.NET 5 RC1 Update 1

Welcome to the ASP.NET 5 RC1 setup. Please close all instances of Visual Studio 2015 before proceeding with the installation.

Microsoft ASP.NET 5 RC1 Update 1 [license terms](#).

Microsoft Web Developer Tools

I agree to the license terms and conditions

Install Close

In this article:

- [Install ASP.NET 5 with Visual Studio](#)
- [Install ASP.NET 5 from the command-line](#)
- [Related Resources](#)

Install ASP.NET 5 with Visual Studio

The easiest way to get started building apps (including the free Community edition).

1. Install Visual Studio 2015

Be sure to specify that you want to include:

Microsoft Web Developer Tools

2. [Install ASP.NET 5](#).

This will install the latest ASP.NET 5 runtime.

3. Enable the ASP.NET 5 command-line tools.

Was this page helpful? Your feedback about this content is important. Let us know what you think.

YES NO

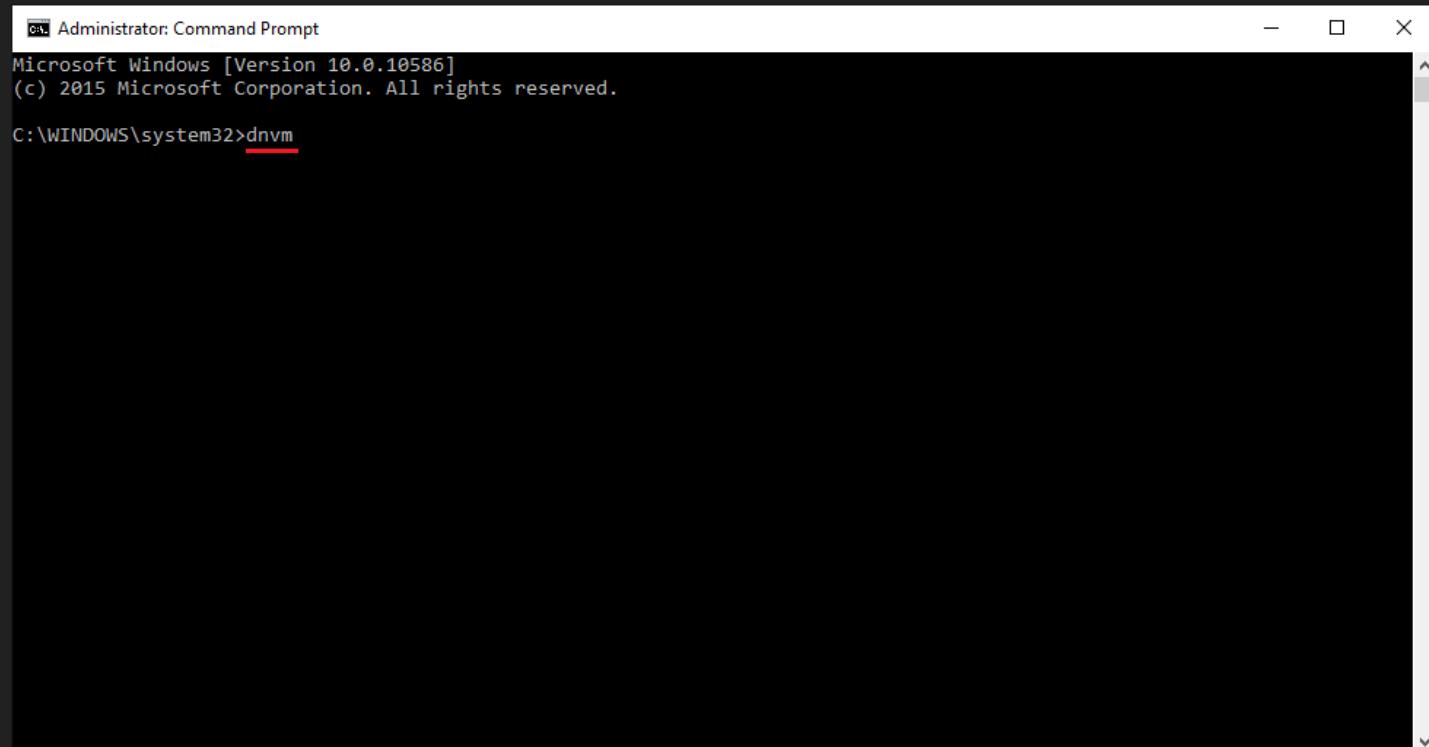
Read the Docs v: 1.0.0-rc1

AspNet5.ENU.RC1_U....exe

¿Por donde empezar? Instalando ASP.NET 5 y nuevos conceptos

7. Luego de la instalación, abre tu línea de comandos (cmd – Windows) o equivalente para otro OS y digita lo siguiente:

> dnvm



```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.10586]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>dnvm
```



¿Por donde empezar? Instalando ASP.NET 5 y nuevos conceptos

8. Esto es nuevo en el mundo de ASP.NET... ¿Qué es el DNVM?

```
Administrator: Command Prompt
You must specify a command!
/ \ V / \ / | / \ / \ / \
/ / / / [ / \ / / ] / /
_____| / | ____| / /
.NET Version Manager v1.0.0-beta5-10384
By Microsoft Open Technologies, Inc.
usage: dnmv <command> [<arguments...>]

Current feed settings:
Default Stable: https://www.nuget.org/api/v2
Default Unstable: https://www.myget.org/F/aspnetvnext/api/v2
Current Stable Override: <none>
Current Unstable Override: <none>

commands:
alias           Lists and manages aliases
exec            Executes the specified command in a sub-shell where the PATH has been augmented to include the specified DNX
help            Displays a list of commands, and help for specific commands
install         Installs a version of the runtime
list             Lists available runtimes
run              Locates the dnx.exe for the specified version or alias and executes it, providing the remaining arguments to dnx.exe
setup            Installs the version manager into your User profile directory
update-self     Updates DNVM to the latest version.
upgrade         Installs the latest version of the runtime and reassigns the specified alias to point at it
use              Adds a runtime to the PATH environment variable for your current shell

C:\WINDOWS\system32>
```

- ❑ DNVM es el Administrador de versiones de ASP.NET 5 DNX en adelante.
 - ❑ Es un nuevo modelo de herramienta runtime (usando línea de comandos), junto con DNX y DNU.
 - ❑ Permite seleccionar la versión de DNX .NET con la que se trabajará.
 - ❑ Si te aparece esta pantalla, quiere decir que el framework ASP.NET 5 ha sido instalado correctamente en el sistema operativo.

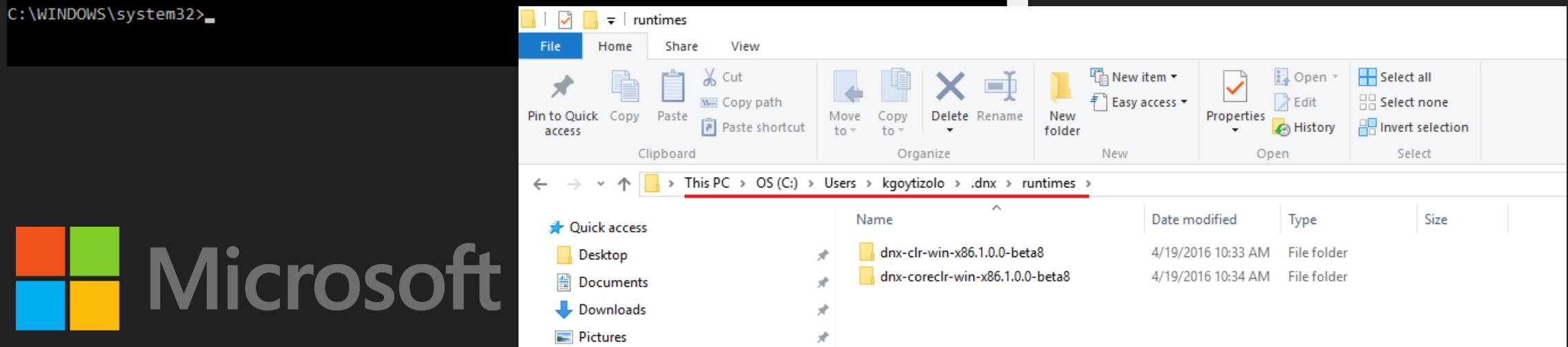


¿Por donde empezar? Instalando ASP.NET 5 y nuevos conceptos

9. Para ver el listado actual de entornos runtime instalados, digitar:

```
c:\ Administrator: Command Prompt  
C:\WINDOWS\system32>dnvm list  
Active Version Runtime Architecture Location Alias  
-----  
1.0.0-beta8 clr x86 C:\Users\kgoytizolo\.dnx\runtimes  
1.0.0-beta8 coreclr x86 C:\Users\kgoytizolo\.dnx\runtimes
```

> Dnvm list

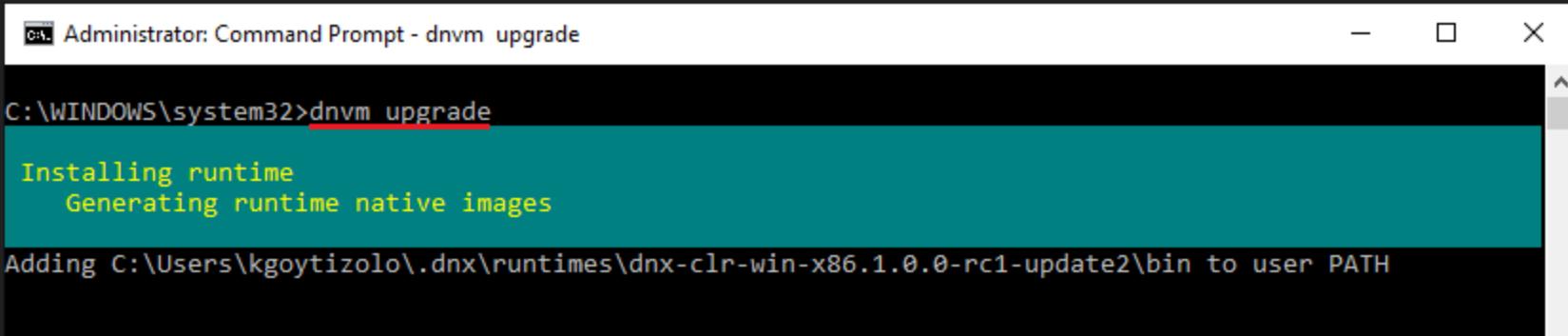


Microsoft



¿Por donde empezar? Instalando ASP.NET 5 y nuevos conceptos

10. En caso requieran obtener la última versión del runtime, digitar:

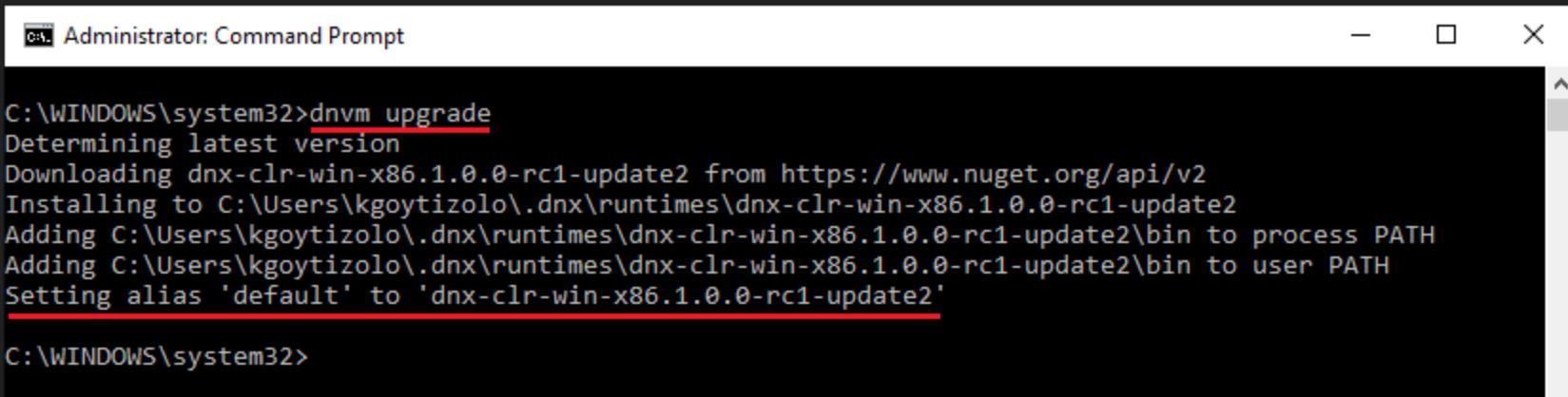


```
C:\Administrator: Command Prompt - dnvm upgrade  
C:\WINDOWS\system32>dnvm upgrade  
  
Installing runtime  
Generating runtime native images  
  
Adding C:\Users\kgoytizolo\.dnx\runtimes\dnx-clr-win-x86.1.0.0-rc1-update2\bin to user PATH
```

> Dnvm upgrade

Parámetros de ejemplo:

[-x86][-x64] [-svr50][-svrc50] [-g] [-
global] [-proxy <ADDRESS>]



```
C:\Administrator: Command Prompt  
C:\WINDOWS\system32>dnvm upgrade  
Determining latest version  
Downloading dnx-clr-win-x86.1.0.0-rc1-update2 from https://www.nuget.org/api/v2  
Installing to C:\Users\kgoytizolo\.dnx\runtimes\dnx-clr-win-x86.1.0.0-rc1-update2  
Adding C:\Users\kgoytizolo\.dnx\runtimes\dnx-clr-win-x86.1.0.0-rc1-update2\bin to process PATH  
Adding C:\Users\kgoytizolo\.dnx\runtimes\dnx-clr-win-x86.1.0.0-rc1-update2\bin to user PATH  
Setting alias 'default' to 'dnx-clr-win-x86.1.0.0-rc1-update2'  
  
C:\WINDOWS\system32>
```



¿Por donde empezar? Instalando ASP.NET 5 y nuevos conceptos

11. Dnvm Upgrade actualizó el último runtime y lo configuró por defecto:

The screenshot illustrates the process of upgrading the .NET runtime using the command prompt and the resulting file structure.

Administrator: Command Prompt

```
C:\WINDOWS\system32>dnvm upgrade
Determining latest version
Downloading dnx-clr-win-x86.1.0.0-rc1-update2 from https://www.nuget.org/api/v2
Installing to C:\Users\kgoytizolo\.dnx\runtimes\dnx-clr-win-x86.1.0.0-rc1-update2
Adding C:\Users\kgoytizolo\.dnx\runtimes\dnx-clr-win-x86.1.0.0-rc1-update2
Adding C:\Users\kgoytizolo\.dnx\runtimes\dnx-clr-win-x86.1.0.0-rc1-update2
Setting alias 'default' to 'dnx-clr-win-x86.1.0.0-rc1-update2'

C:\WINDOWS\system32>dnvm list
```

Active	Version	Runtime	Architecture	Location	Alias
	1.0.0-beta8	clr	x86	C:\Users\kgoytizolo\.dnx\runtimes	
	1.0.0-beta8	coreclr	x86	C:\Users\kgoytizolo\.dnx\runtimes	
*	1.0.0-rc1-update2	clr	x86	C:\Users\kgoytizolo\.dnx\runtimes	default

File Explorer

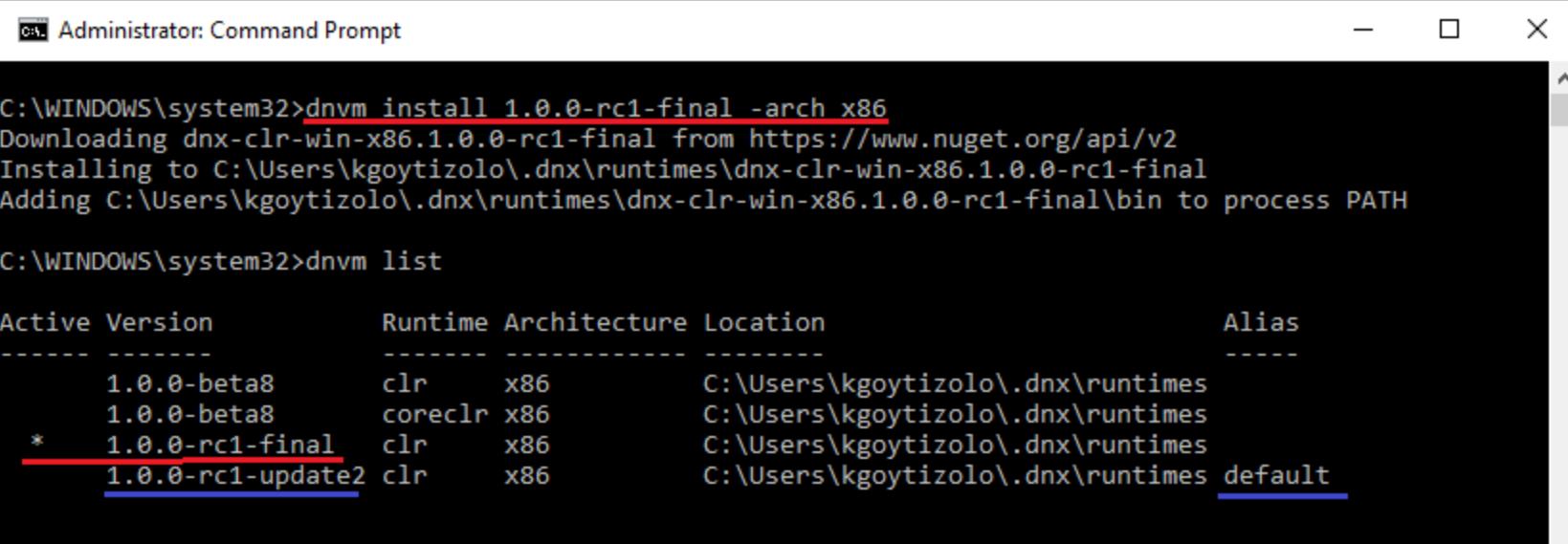
This PC > OS (C:) > Users > kgoytizolo > .dnx > runtimes

Name	Date modified	Type
dnx-clr-win-x86.1.0.0-beta8	4/19/2016 10:33 AM	File folder
dnx-clr-win-x86.1.0.0-rc1-update2	5/18/2016 2:38 PM	File folder
dnx-coreclr-win-x86.1.0.0-beta8	4/19/2016 10:34 AM	File folder

A circular arrow icon in the bottom right corner indicates a continuation or next step.

¿Por donde empezar? Instalando ASP.NET 5 y nuevos conceptos

12. En caso requieran instalar otras versiones del runtime, digitar:



```
Administrator: Command Prompt

C:\WINDOWS\system32>dnvm install 1.0.0-rc1-final -arch x86
Downloading dnx-clr-win-x86.1.0.0-rc1-final from https://www.nuget.org/api/v2
Installing to C:\Users\kgoytizolo\.dnx\runtimes\dnx-clr-win-x86.1.0.0-rc1-final
Adding C:\Users\kgoytizolo\.dnx\runtimes\dnx-clr-win-x86.1.0.0-rc1-final\bin to process PATH

C:\WINDOWS\system32>dnvm list

Active Version      Runtime Architecture Location          Alias
----- -----
  1.0.0-beta8      clr      x86      C:\Users\kgoytizolo\.dnx\runtimes
  1.0.0-beta8      coreclr  x86      C:\Users\kgoytizolo\.dnx\runtimes
 * 1.0.0-rc1-final clr      x86      C:\Users\kgoytizolo\.dnx\runtimes
1.0.0-rc1-update2 clr      x86      C:\Users\kgoytizolo\.dnx\runtimes default
```

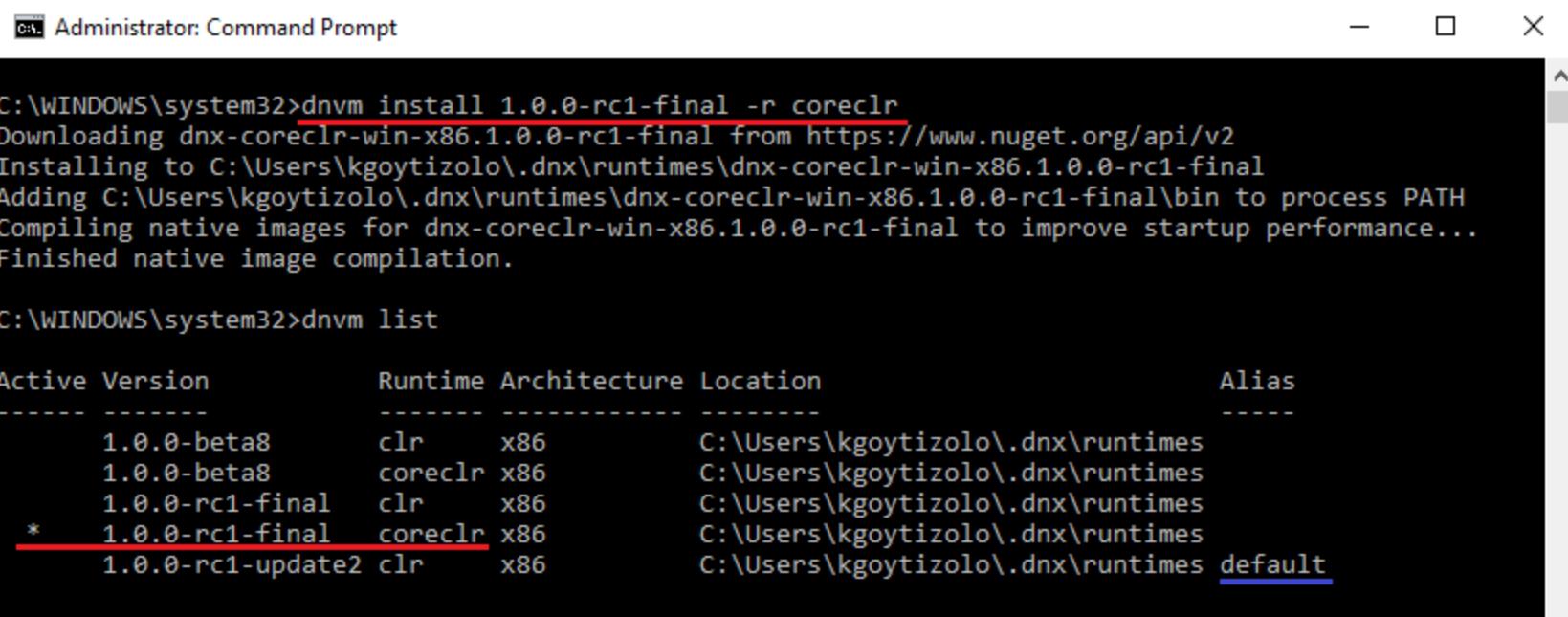
> Dnvm install

- Parámetros ejemplo:
 - Versión: 1.0.0-rc1-final
 - Arquitectura: -arch x86
- dnvm install
 - <semver> | <alias> | <nupkg> | latest [-x86][-x64] [-svr50][-svrc50] [-a] [-alias <alias>] [-g] [-global] [-f] [-force]



¿Por donde empezar? Instalando ASP.NET 5 y nuevos conceptos

13. En caso requieran instalar otras versiones del runtime, digitar:



```
Administrator: Command Prompt

C:\WINDOWS\system32>dnvm install 1.0.0-rc1-final -r coreclr
Downloading dnx-coreclr-win-x86.1.0.0-rc1-final from https://www.nuget.org/api/v2
Installing to C:\Users\kgoytizolo\.dnx\runtimes\dnx-coreclr-win-x86.1.0.0-rc1-final
Adding C:\Users\kgoytizolo\.dnx\runtimes\dnx-coreclr-win-x86.1.0.0-rc1-final\bin to process PATH
Compiling native images for dnx-coreclr-win-x86.1.0.0-rc1-final to improve startup performance...
Finished native image compilation.

C:\WINDOWS\system32>dnvm list

Active Version      Runtime Architecture Location          Alias
----- -----      ----- -----
  1.0.0-beta8       clr      x86      C:\Users\kgoytizolo\.dnx\runtimes
  1.0.0-beta8       coreclr   x86      C:\Users\kgoytizolo\.dnx\runtimes
  1.0.0-rc1-final    clr      x86      C:\Users\kgoytizolo\.dnx\runtimes
* 1.0.0-rc1-final   coreclr   x86      C:\Users\kgoytizolo\.dnx\runtimes default
  1.0.0-rc1-update2 clr      x86      C:\Users\kgoytizolo\.dnx\runtimes
```

> Dnvm install

- Parámetros ejemplo:
 - Versión: 1.0.0-rc1-final
 - Runtime: -r coreclr
- dnvm install
 - <semver> | <alias> | <nupkg> | latest [-x86][-x64] [-svr50][-svrc50] [-a] [-alias <alias>] [-g] [-global] [-f] [-force]



* Este Runtime (CoreCLR) demora un poco más en instalar por la compilación y librerías adicionales por no requerir .NET en la máquina



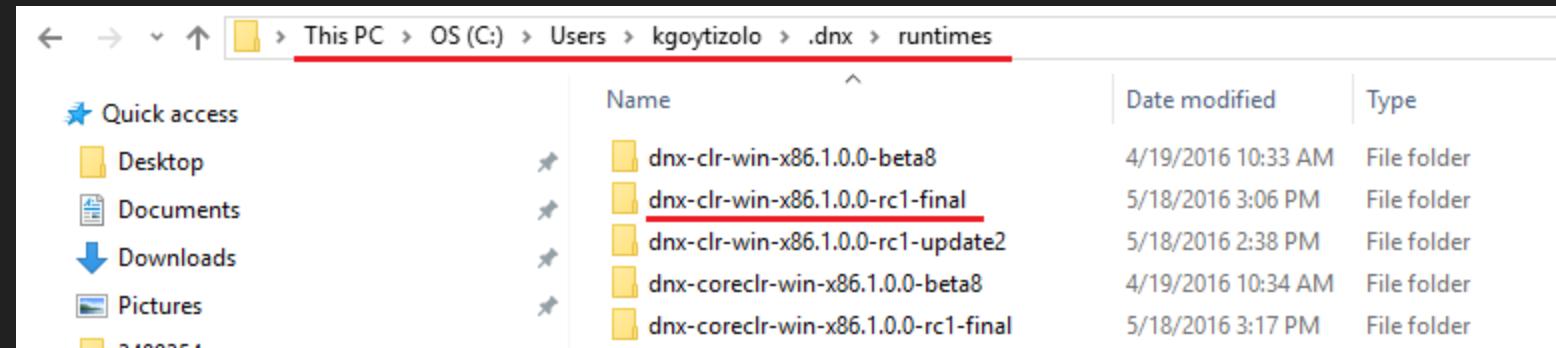
¿Por donde empezar? Instalando ASP.NET 5 y nuevos conceptos

14. Para definir el runtime a utilizar, digitar:

```
C:\WINDOWS\system32>dnvm use 1.0.0-rc1-final -r clr -arch x86
Adding C:\Users\kgoytizolo\.dnx\runtimes\dnx-clr-win-x86.1.0.0-rc1-final\bin to process PATH

C:\WINDOWS\system32>dnvm list

Active Version      Runtime Architecture Location          Alias
----- -----
  1.0.0-beta8       clr      x86      C:\Users\kgoytizolo\.dnx\runtimes
  1.0.0-beta8       coreclr  x86      C:\Users\kgoytizolo\.dnx\runtimes
* 1.0.0-rc1-final   clr      x86      C:\Users\kgoytizolo\.dnx\runtimes
  1.0.0-rc1-final   coreclr  x86      C:\Users\kgoytizolo\.dnx\runtimes
  1.0.0-rc1-update2 clr      x86      C:\Users\kgoytizolo\.dnx\runtimes default
```



> Dnvm use

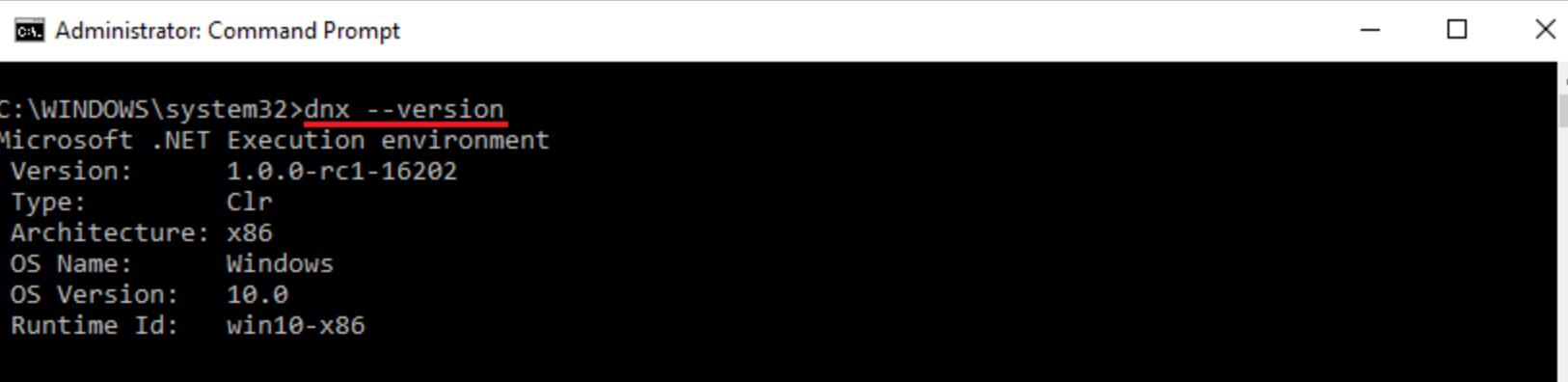
- Parámetros ejemplo:
 - Versión: 1.0.0-rc1-final
 - Runtime: -r clr
 - Arquitectura: -arch x86
- dnvm use
<semver> | <alias> | none [-x86][-x64] [-svr50][-svrc50] [-p | -persistent] [-g | -global]

* En este punto, añade una carpeta BIN del Runtime para los procesos PATH



¿Por donde empezar? Instalando ASP.NET 5 y nuevos conceptos

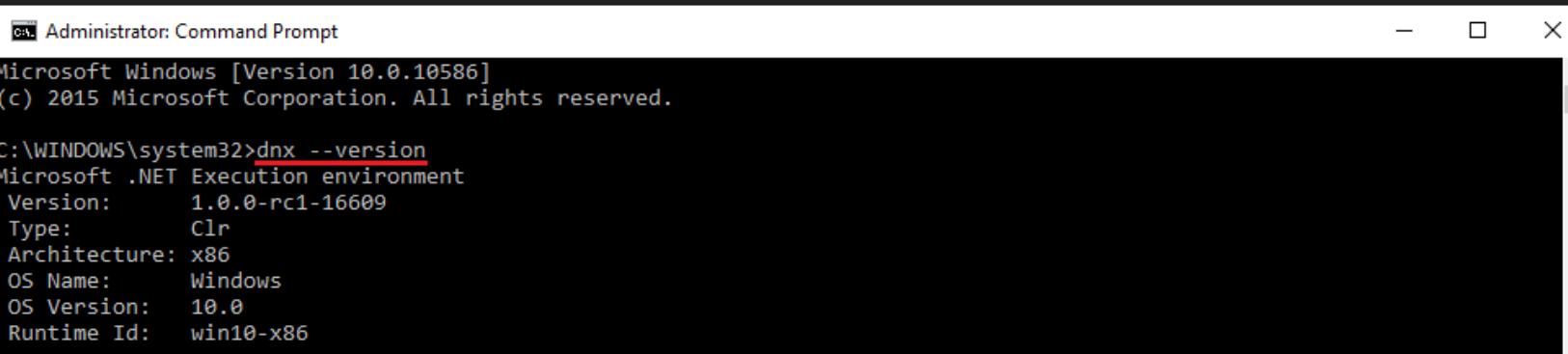
15. Finalmente, verificamos la versión DNX a utilizar, digitando:



```
C:\WINDOWS\system32>dnx --version
Microsoft .NET Execution environment
Version: 1.0.0-rc1-16202
Type: Clr
Architecture: x86
OS Name: Windows
OS Version: 10.0
Runtime Id: win10-x86
```

> Dnx --version

* Importante: La definición del Dnx Activo es por instancia / sesión.



```
Microsoft Windows [Version 10.0.10586]
(c) 2015 Microsoft Corporation. All rights reserved.

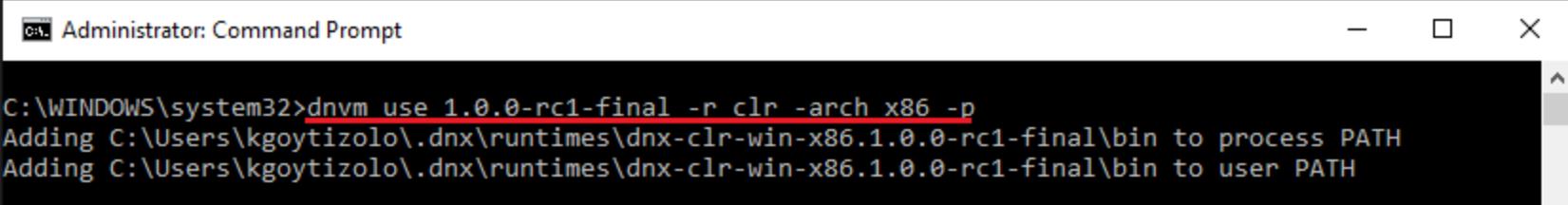
C:\WINDOWS\system32>dnx --version
Microsoft .NET Execution environment
Version: 1.0.0-rc1-16609
Type: Clr
Architecture: x86
OS Name: Windows
OS Version: 10.0
Runtime Id: win10-x86
```

* Si se abre otra pantalla de comandos, trabajará con el Dnx definido por defecto.



¿Por donde empezar? Instalando ASP.NET 5 y nuevos conceptos

16. Para definir el runtime a utilizar de manera persistente:



```
C:\WINDOWS\system32>dnvm use 1.0.0-rc1-final -r clr -arch x86 -p
Adding C:\Users\kgoytizolo\.dnx\runtimes\dnx-clr-win-x86.1.0.0-rc1-final\bin to process PATH
Adding C:\Users\kgoytizolo\.dnx\runtimes\dnx-clr-win-x86.1.0.0-rc1-final\bin to user PATH
```

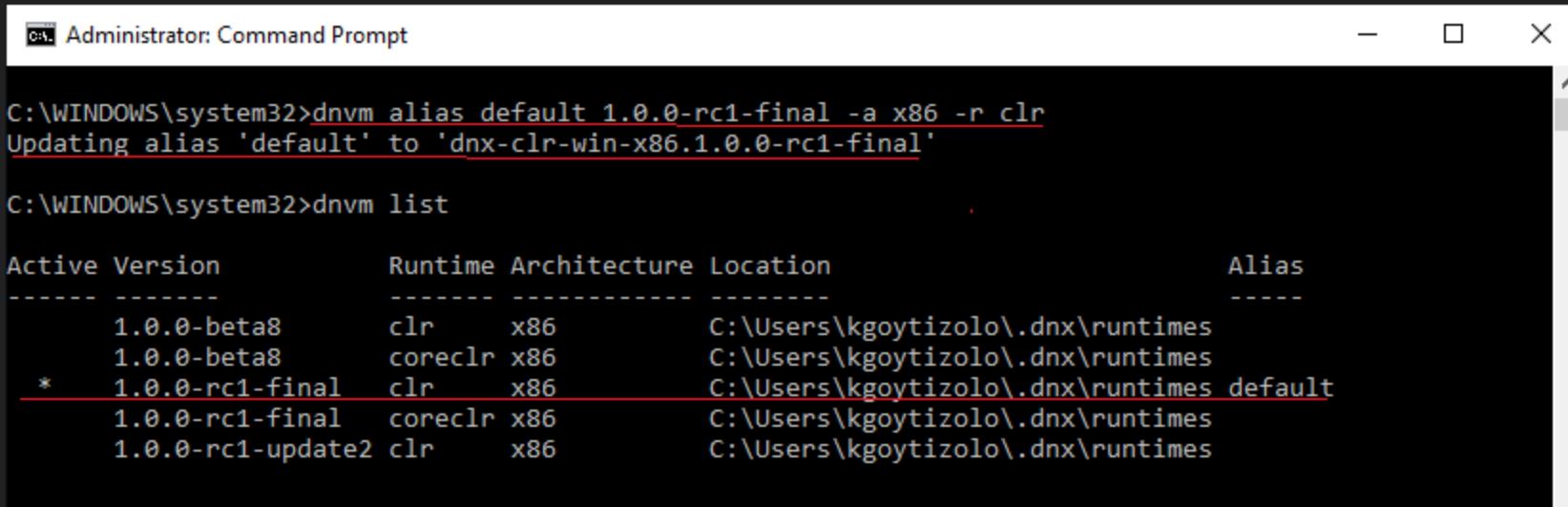
> Dnvm use

- Parámetros ejemplo:
 - Versión: 1.0.0-rc1-final
 - Runtime: -r clr
 - Arquitectura: -arch x86
 - Persistencia: -p
- dnvm use
<semver> | <alias> | none [-x86][-x64] [-svr50][-svrc50] [-p | -persistent] [-g | -global]



¿Por donde empezar? Instalando ASP.NET 5 y nuevos conceptos

17. Para definir el alias de un runtime a utilizar, digitar:



```
C:\WINDOWS\system32>dnvm alias default 1.0.0-rc1-final -a x86 -r clr
Updating alias 'default' to 'dnx-clr-win-x86.1.0.0-rc1-final'.

C:\WINDOWS\system32>dnvm list

Active Version      Runtime Architecture Location          Alias
----- -----      ----- -----
  1.0.0-beta8       clr      x86    C:\Users\kgoytizolo\.dnx\runtimes
  1.0.0-beta8       coreclr  x86    C:\Users\kgoytizolo\.dnx\runtimes
* 1.0.0-rc1-final   clr      x86    C:\Users\kgoytizolo\.dnx\runtimes default
  1.0.0-rc1-final   coreclr  x86    C:\Users\kgoytizolo\.dnx\runtimes
  1.0.0-rc1-update2 clr      x86    C:\Users\kgoytizolo\.dnx\runtimes
```

> Dnvm alias

- Parámetros ejemplo:
 - Nombre: default
 - Versión: 1.0.0-rc1-final
 - Arquitectura: -a x86
 - Runtime: -r clr
- dnvm alias <alias> <semver> [-x86][-x64] [-svr50][-svrc50]

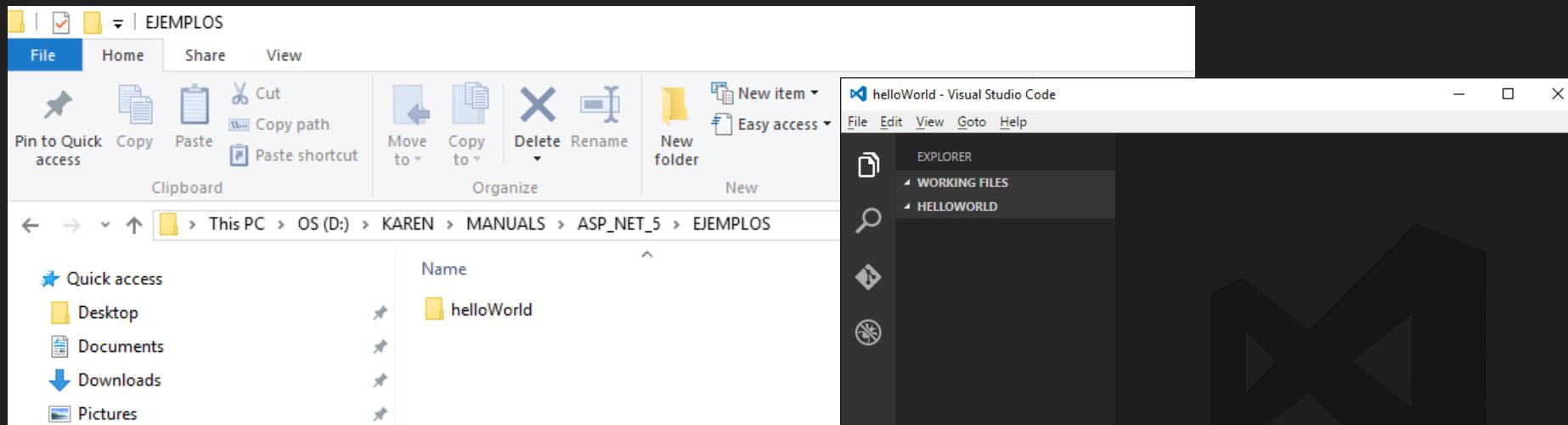


* Importante sobretodo para usar en Visual Studio Code.



Hola Mundo! Mi primera app con ASP.NET 5 y Visual Studio Code

1. Crea una carpeta llamada “helloWorld” y a ese nivel abrir con Visual Studio Code:



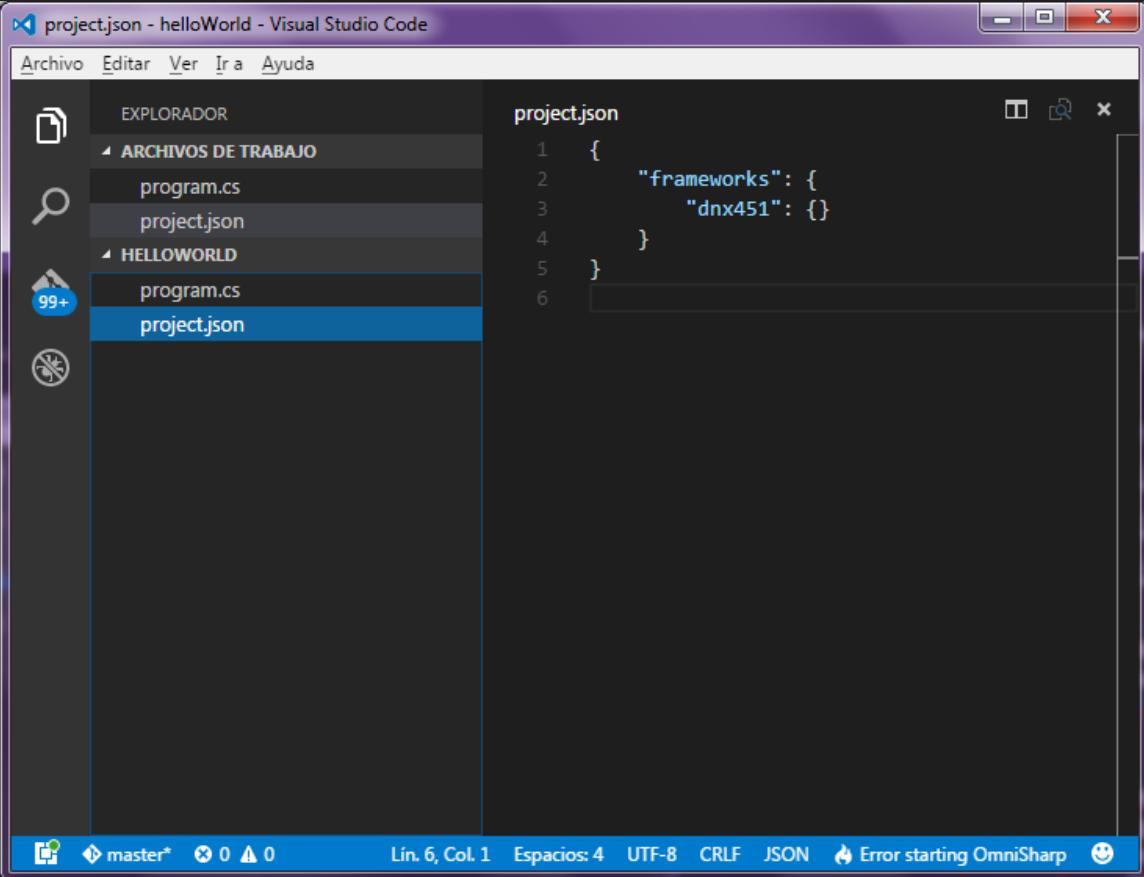
También se puede
abrir desde el mismo
Visual Studio Code en:

File > Open Folder



Hola Mundo! Mi primera app con ASP.NET 5 y Visual Studio Code

2. Crear un archivo “Project.json” asociado al proyecto “helloWorld”:



The screenshot shows the Visual Studio Code interface with the title bar "project.json - helloWorld - Visual Studio Code". The left sidebar shows a file tree with "EXPLORADOR", "ARCHIVOS DE TRABAJO" containing "program.cs" and "project.json", and a "HELLOWORLD" folder containing "program.cs" and "project.json". The "project.json" file in the "HELLOWORLD" folder is selected and open in the main editor area. The code in the editor is:

```
1 {  
2     "frameworks": {  
3         "dnx451": {}  
4     }  
5 }  
6
```

The status bar at the bottom shows "master*" and "Error starting OmniSharp".

- Contiene información del proyecto:
 - Framework a utilizar
 - Dependencias
 - Comandos

* Añadir framework dnx451 dentro del proyecto

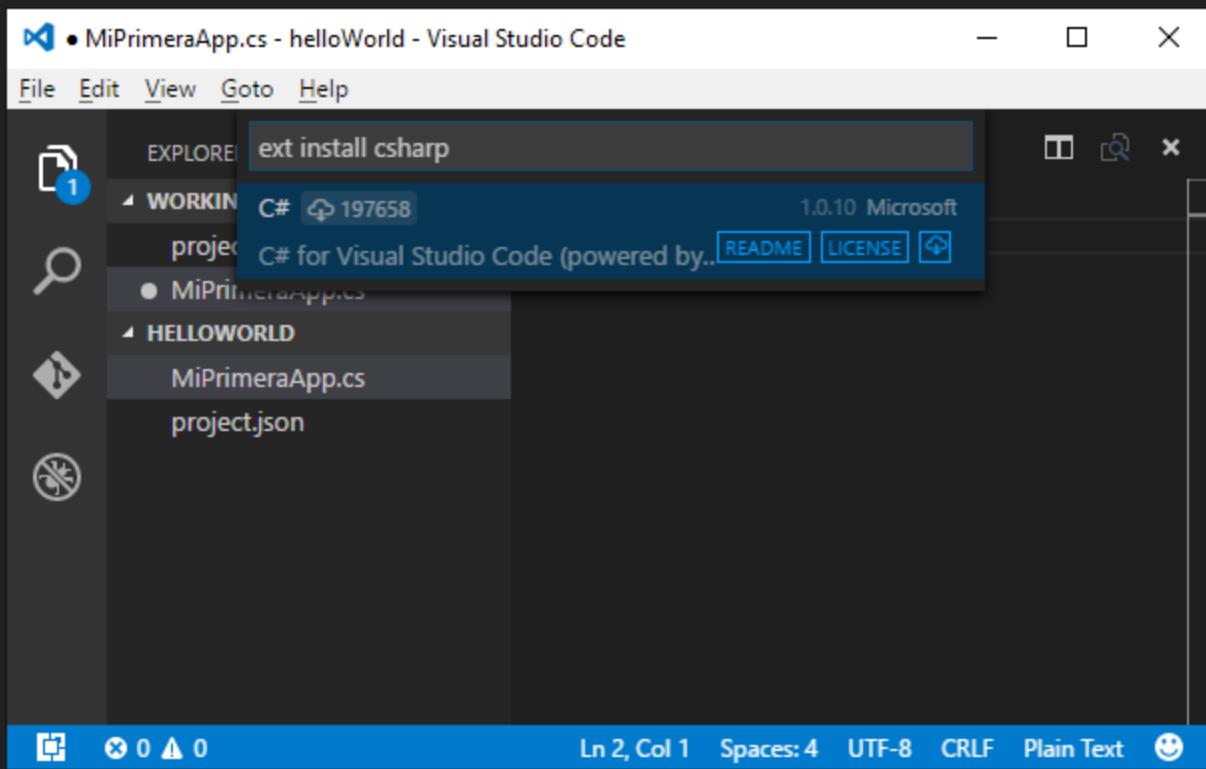
Importante:

Project.json es un archivo de configuración requerido como mínimo para una aplicación ASP.NET 5 en adelante



Hola Mundo! Mi primera app con ASP.NET 5 y Visual Studio Code

3. Antes de proseguir, puedes añadir extensiones para el editor Visual Studio Code (por ejemplo, para sintaxis de lógica C#):



CTRL + P >
ext install csharp

Fuentes de extensiones para Visual Studio Code:
(Marketplace):
<https://marketplace.visualstudio.com/VSCode>



Hola Mundo! Mi primera app con ASP.NET 5 y Visual Studio Code

4. Crear un archivo “program.cs” asociado al proyecto “helloWorld”

The screenshot shows the Visual Studio Code interface. The title bar says "program.cs - helloWorld - Visual Studio Code". The menu bar includes "Archivo", "Editar", "Ver", "Ir a", and "Ayuda". The left sidebar has icons for "EXPLORADOR", "ARCHIVOS DE TRABAJO" (with "program.cs" and "project.json" listed), and "HELLOWORLD" (with "program.cs" and "project.json" listed). The "program.cs" file in the "HELLOWORLD" folder is open in the main editor area. The code is:

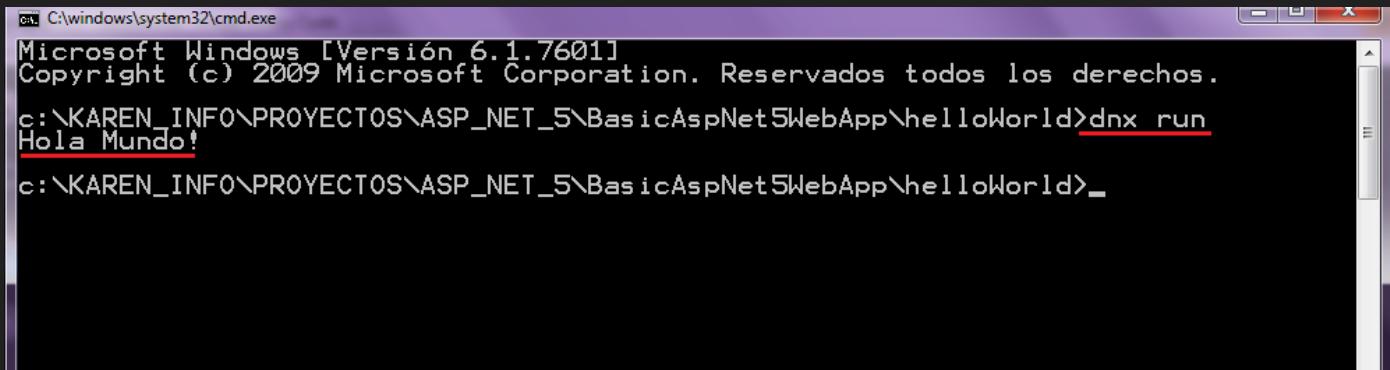
```
1  using System;
2
3  public class Program
4  {
5      public void Main()
6      {
7          Console.WriteLine("Hola Mundo!");
8      }
9  }
10
```

- Contiene la lógica en lenguaje C#.
- Crea un tipo de aplicación de consola, el mínimo para considerar como aplicación.
- Usar la clase estándar Program.cs para cargar aplicaciones de consola.



Hola Mundo! Mi primera app con ASP.NET 5 y Visual Studio Code

5. En el archivo “`program.cs`”, click derecho > Abrir en línea de comandos para utilizar la herramienta runtime DNX. Digitar:



```
C:\windows\system32\cmd.exe
Microsoft Windows [Versión 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Reservados todos los derechos.
c:\KAREN_INFO\PROYECTOS\ASP_NET_5\BasicAspNet5WebApp\helloWorld>dnx run
Hola Mundo!
c:\KAREN_INFO\PROYECTOS\ASP_NET_5\BasicAspNet5WebApp\helloWorld>
```

> `Dnx run`

- DNX es el entorno de ejecución de .NET para aplicaciones ASP.NET 5 en adelante.
- A partir de este framework, los proyectos son basados en DNX.
- DNX permite construir y ejecutar aplicaciones multiplataforma.
- `Dnx run` ejecuta la aplicación, hosting, debug, etc.

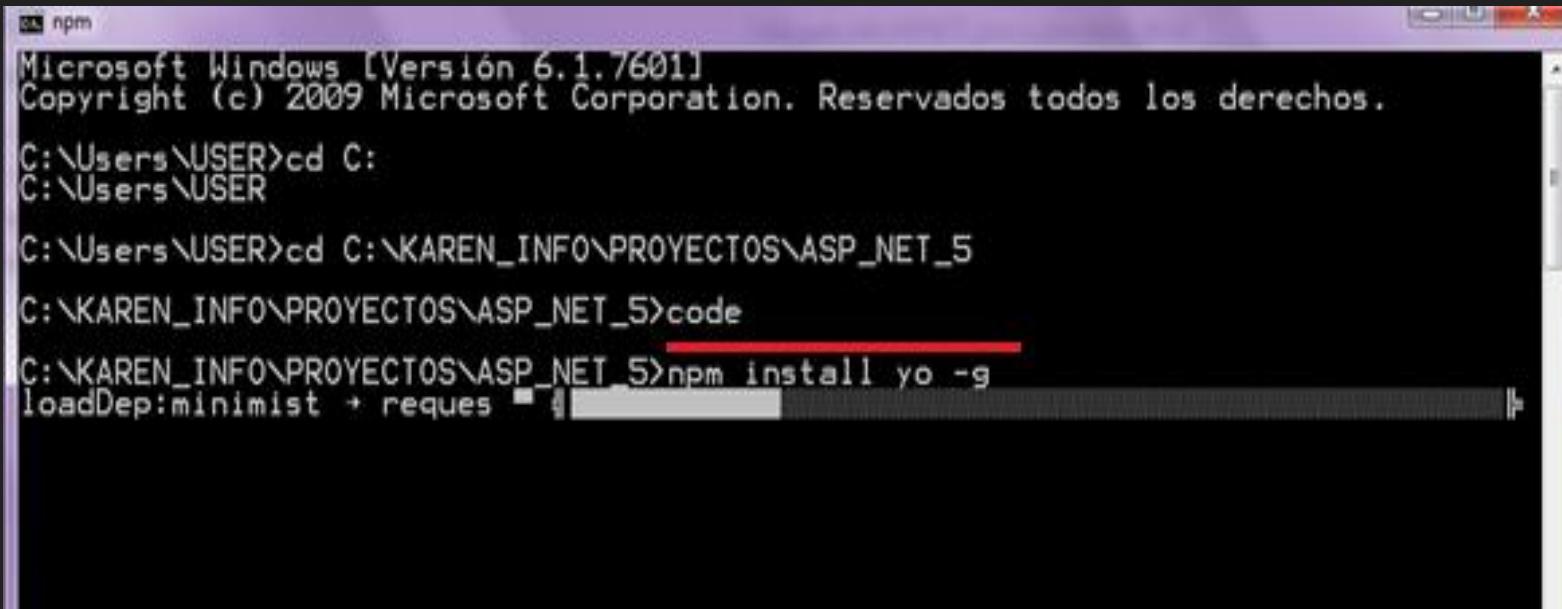


!Listo! Ya está!



Usando aplicaciones MVC plantilla con DNX y Visual Studio Code

1. Descargar e instalar Yeoman usando la herramienta NPM y Node JS:



```
npm
Microsoft Windows [Versión 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Reservados todos los derechos.
C:\Users\USER>cd C:
C:\Users\USER>
C:\Users\USER>cd C:\KAREN_INFO\PROYECTOS\ASP_NET_5
C:\KAREN_INFO\PROYECTOS\ASP_NET_5>code
C:\KAREN_INFO\PROYECTOS\ASP_NET_5>npm install yo -g
loadDep:minimist + reques
```

> Npm install
yo -g

Parámetro -g (Indica que se instale globalmente para que esté disponible en cualquier parte de la máquina).

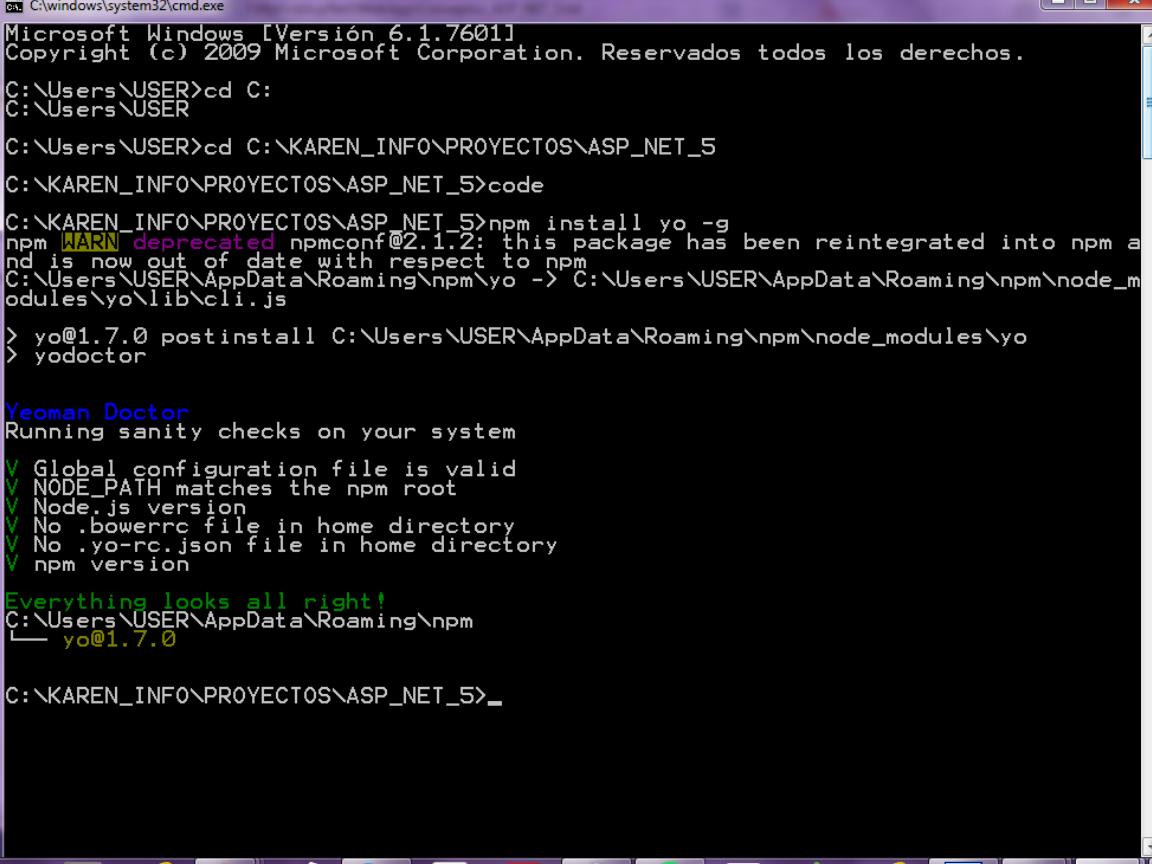
Yeoman:

Permite utilizar diferentes plantillas, generación de código y soporte de diferentes partes de la aplicación.



Usando aplicaciones MVC plantilla con DNX y Visual Studio Code

1. Descargar e instalar Yeoman usando la herramienta NPM y Node JS:



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Versión 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Reservados todos los derechos.

C:\Users\USER>cd C:
C:\Users\USER>cd C:\KAREN_INFO\PROYECTOS\ASP_NET_5
C:\KAREN_INFO\PROYECTOS\ASP_NET_5>code

C:\KAREN_INFO\PROYECTOS\ASP_NET_5>npm install yo -g
npm [WARN] deprecated npmcnfig@2.1.2: this package has been reintegrated into npm and is now out of date with respect to npm
C:\Users\USER\AppData\Roaming\npm\yo -> C:\Users\USER\AppData\Roaming\npm\node_modules\yo\lib\cli.js

> yo@1.7.0 postinstall C:\Users\USER\AppData\Roaming\npm\node_modules\yo
> yodoctor

Yeoman Doctor
Running sanity checks on your system
  ✓ Global configuration file is valid
  ✓ NODE_PATH matches the npm root
  ✓ Node.js version
  ✓ No .bowerrc file in home directory
  ✓ No .yo-rc.json file in home directory
  ✓ npm version

Everything looks all right!
C:\Users\USER\AppData\Roaming\npm
└─ yo@1.7.0

C:\KAREN_INFO\PROYECTOS\ASP_NET_5>
```

> Npm install

yo -g

Parámetro -g (Indica que se instale globalmente para que esté disponible en cualquier parte de la máquina).

Yeoman:

Permite utilizar diferentes plantillas, generación de código y soporte de diferentes partes de la aplicación.



Usando aplicaciones MVC plantilla con DNX y Visual Studio Code

1. Descargar e instalar Yeoman usando la herramienta NPM y Node JS:

```
C:\windows\system32\cmd.exe
C:\KAREN_INFO\PROJECT05\ASP_NET_5>tree
└─┬ run-async@2.2.0
   ├─┬ ls-promise@2.1.0
   │   └─┬ rx@4.1.0
   │       └─┬ minimist@1.2.0
   │           └─┬ npmlog@2.0.4
   │               ├─┬ ansi@0.3.1
   │               │   └─┬ are-we-there-yet@1.1.2
   │               │       └─┬ delegates@1.0.0
   │               │           └─┬ gauge@1.2.7
   │               │               └─┬ has-unicode@2.0.0
   │               │                   └─┬ lodash.pad@4.4.0
   │               │                       └─┬ lodash._baseslice@4.0.0
   │               │                           └─┬ lodash._basetoString@4.12.0
   │               │                               └─┬ lodash.tostring@4.1.3
   │               │                                   └─┬ lodash.padend@4.5.0
   │               │                                       └─┬ lodash.padstart@4.5.0
   │               └─┬ yeoman-doctor@2.1.0
   │                   └─┬ bin-version-check@2.1.0
   │                       ├─┬ lodash._baseSlice@4.0.0
   │                       │   └─┬ lodash._baseToString@4.12.0
   │                       │       └─┬ lodash.tostring@4.1.3
   │                       └─┬ lodash.padend@4.5.0
   │                           └─┬ lodash.padstart@4.5.0
   └─┬ yeoman-environment@1.6.1
      ├─┬ brace-expansion@1.1.4
      │   └─┬ balanced-match@0.4.1
      └─┬ globby@4.1.0
          ├─┬ glob@6.0.4
          │   └─┬ inflight@1.0.5
          │       └─┬ inquirer@1.0.2
          │           └─┬ cli-width@2.1.0
          │               └─┬ mute-stream@0.0.6
          │                   └─┬ run-async@2.2.0
          │                       └─┬ lodash@4.12.0
          │                           └─┬ mem-fs@1.1.3
          │                               └─┬ through2@2.0.1
          │                                   └─┬ readable-stream@2.0.6
          │                                       └─┬ vinyl-file@2.0.0
          │                                           └─┬ strip-bom-stream@2.0.0
          │                                               └─┬ first-chunk-stream@2.0.0
          └─┬ yosay@1.1.1
              └─┬ taketalk@1.0.0
                  └─┬ lodash._baseSlice@4.0.0
                      └─┬ lodash._baseToString@4.12.0
                          └─┬ lodash.tostring@4.1.3
                          └─┬ lodash.padend@4.5.0
                          └─┬ lodash.padstart@4.5.0
```

```
> Npm install  
yo -g
```

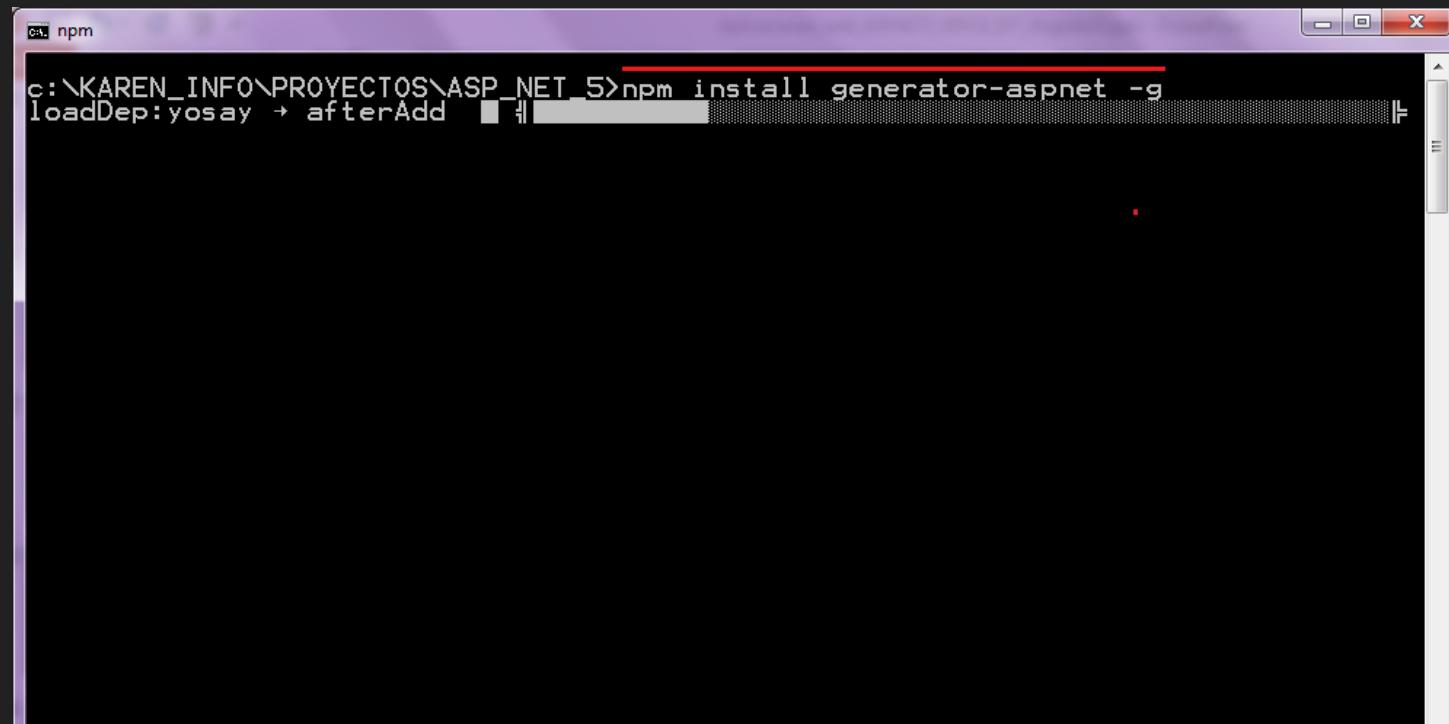
Parámetro –g (Indica que se instale globalmente para que esté disponible en cualquier parte de la máquina).

Yeoman:

Permite utilizar diferentes plantillas, generación de código y soporte de diferentes partes de la aplicación.

Usando aplicaciones MVC plantilla con DNX y Visual Studio Code

2. Instalamos un ASP.NET template vía NPM usando el generador aspnet:



```
c:\KAREN_INFO\PROYECTOS\ASP_NET_5>npm install generator-aspnet -g  
loadDep:yosay → afterAdd
```

> Npm install
generator-
aspnet -g

Hay muchos generadores soportados
por Yeoman. Empiezan con el
generador “dash”



Usando aplicaciones MVC plantilla con DNX y Visual Studio Code

2. Instalamos un ASP.NET template vía NPM usando el generador aspnet:

```
c:\windows\system32\cmd.exe
C:\KAREN_INFO\PROYECTOS\ASP_NET_5> npm install -g generator-aspnet
[...]
c:\KAREN_INFO\PROYECTOS\ASP_NET_5>
```

> Npm install
generator-
aspnet -g

Hay muchos generadores soportados
por Yeoman. Empiezan con el
generador “dash”



Usando aplicaciones MVC plantilla con DNX y Visual Studio Code

3. Ejecutamos la herramienta Yeoman, asociada al generador ASP.NET:

```
cmd yo
yeoman-welcome@1.0.1
yosay@1.1.1
ansi-regex@2.0.0
pad-component@0.0.1
repeating@2.0.1
is-finite@1.0.1
string-width@1.0.1
code-point-at@1.0.0
is-fullwidth-code-point@1.0.0
taketalk@1.0.0
minimist@1.2.0
wrap-ansi@2.0.0

npm [WARN] ENOENT ENOENT: no such file or directory, open 'C:\KAREN_INFO\PROYECTOS\ASP.NET_5\package.json'
npm [WARN] EPACKAGEJSON ASP.NET_5 No description
npm [WARN] EPACKAGEJSON ASP.NET_5 No repository field.
npm [WARN] EPACKAGEJSON ASP.NET_5 No README data
npm [WARN] EPACKAGEJSON ASP.NET_5 No license field.

C:\KAREN_INFO\PROYECTOS\ASP.NET_5>yo aspnet
[---(o)---]
[---(U---)
 ( - A - )
 / \ T \ / Y \
[-----]
[-----] Welcome to the
[-----] marvellous ASP.NET 5
[-----] generator!
[-----]

? What type of application do you want to create? Web Application
? What's the name of your ASP.NET application? (WebApplication)
```

> Yo aspnet



Usando aplicaciones MVC plantilla con DNX y Visual Studio Code

4. Seleccionamos el tipo de aplicación ASP.NET 5 del generador (En este caso, Web Application [opción Nro 3]) y el Nombre de tu aplicación web:

```
cmd: yo
yosay@1.1.1
ansi-regex@2.0.0
pad-component@0.0.1
repeating@2.0.1
is-finite@1.0.1
string-width@1.0.1
code-point-at@1.0.0
is-fullwidth-code-point@1.0.0
taketalk@1.0.0
minimist@1.2.0
wrap-ansi@2.0.0

npm [WARN] ENOENT ENOENT: no such file or directory, open 'C:\KAREN_INFO\PROYECTOS\ASP.NET_5\package.json'
npm [WARN] EPACKAGEJSON ASP.NET_5 No description
npm [WARN] EPACKAGEJSON ASP.NET_5 No repository field.
npm [WARN] EPACKAGEJSON ASP.NET_5 No README data
npm [WARN] EPACKAGEJSON ASP.NET_5 No license field.

C:\KAREN_INFO\PROYECTOS\ASP.NET_5>yo aspnet

  _--(o)--_
  |  U   |
  ( - A - )
  \  T   /
    Y   /`-.

  Welcome to the
  marvellous ASP.NET 5
  generator!

? What type of application do you want to create? Web Application
? What's the name of your ASP.NET application? (WebApplication)
```

> Yo aspnet



Usando aplicaciones MVC plantilla con DNX y Visual Studio Code

5. Y listo! La aplicación MVC 6 web plantilla fue creada satisfactoriamente:

```
C:\Windows\system32\cmd.exe
create BasicAspNet5WebApp\Views\Account\Lockout.cshtml
create BasicAspNet5WebApp\Views\Account\Login.cshtml
create BasicAspNet5WebApp\Views\Account\Register.cshtml
create BasicAspNet5WebApp\Views\Account\ResetPassword.cshtml
create BasicAspNet5WebApp\Views\Account\ResetPasswordConfirmation.cshtml
create BasicAspNet5WebApp\Views\Account\SendCode.cshtml
create BasicAspNet5WebApp\Views\Account\VerifyCode.cshtml
create BasicAspNet5WebApp\Views\Home\About.cshtml
create BasicAspNet5WebApp\Views\Home>Contact.cshtml
create BasicAspNet5WebApp\Views\Home\Index.cshtml
create BasicAspNet5WebApp\Views\Manage\AddPhoneNumber.cshtml
create BasicAspNet5WebApp\Views\Manage\ChangePassword.cshtml
create BasicAspNet5WebApp\Views\Manage\Index.cshtml
create BasicAspNet5WebApp\Views\Manage\ManageLogins.cshtml
create BasicAspNet5WebApp\Views\Manage\SetPassword.cshtml
create BasicAspNet5WebApp\Views\Manage\VerifyPhoneNumber.cshtml
create BasicAspNet5WebApp\Views\Shared\_Layout.cshtml
create BasicAspNet5WebApp\Views\Shared\_LoginPartial.cshtml
create BasicAspNet5WebApp\Views\Shared\ValidationScriptsPartial.cshtml
create BasicAspNet5WebApp\Views\Shared>Error.cshtml
create BasicAspNet5WebApp\wwwroot\css\site.css
create BasicAspNet5WebApp\wwwroot\css\site.min.css
create BasicAspNet5WebApp\wwwroot\favicon.ico
create BasicAspNet5WebApp\wwwroot\images\ASP-NET-Banners-01.png
create BasicAspNet5WebApp\wwwroot\images\ASP-NET-Banners-02.png
create BasicAspNet5WebApp\wwwroot\images\Banner-01-Azure.png
create BasicAspNet5WebApp\wwwroot\images\Banner-02-VS.png
create BasicAspNet5WebApp\wwwroot\js\site.js
create BasicAspNet5WebApp\wwwroot\js\site.min.js
create BasicAspNet5WebApp\wwwroot\web.config

Your project is now created, you can use the following commands to get going
  cd "BasicAspNet5WebApp"
  dnu restore
  dnu build (optional, build will also happen when it's run)
  dnx web

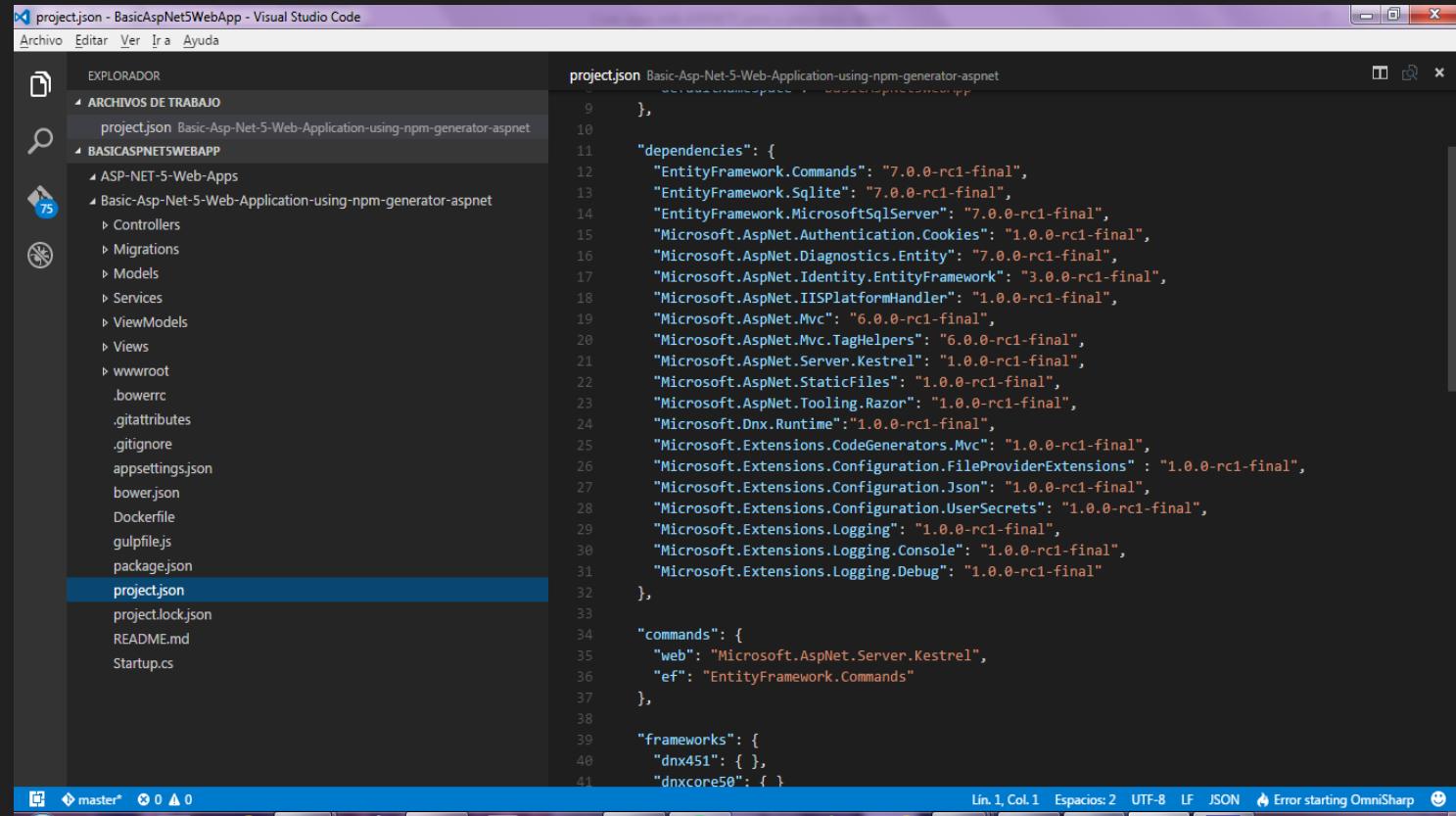
C:\KAREN_INFO\PROYECTOS\ASP_NET_5>
```

> Yo aspnet



Usando aplicaciones MVC plantilla con DNX y Visual Studio Code

6. Abrir la aplicación generada por Yeoman con Visual Studio Code:



The screenshot shows the Visual Studio Code interface with the title bar "project.json - BasicAspNet5WebApp - Visual Studio Code". The left sidebar is the "EXPLORADOR" (Explorer) showing the file structure of the "BASICASPNET5WEBAPP" project, which includes files like .bowerrc, .gitignore, Dockerfile, gulpfile.js, package.json, project.json (which is currently selected), project.lock.json, README.md, and Startup.cs. The main editor area displays the content of the project.json file, which defines dependencies, commands, and frameworks for the application.

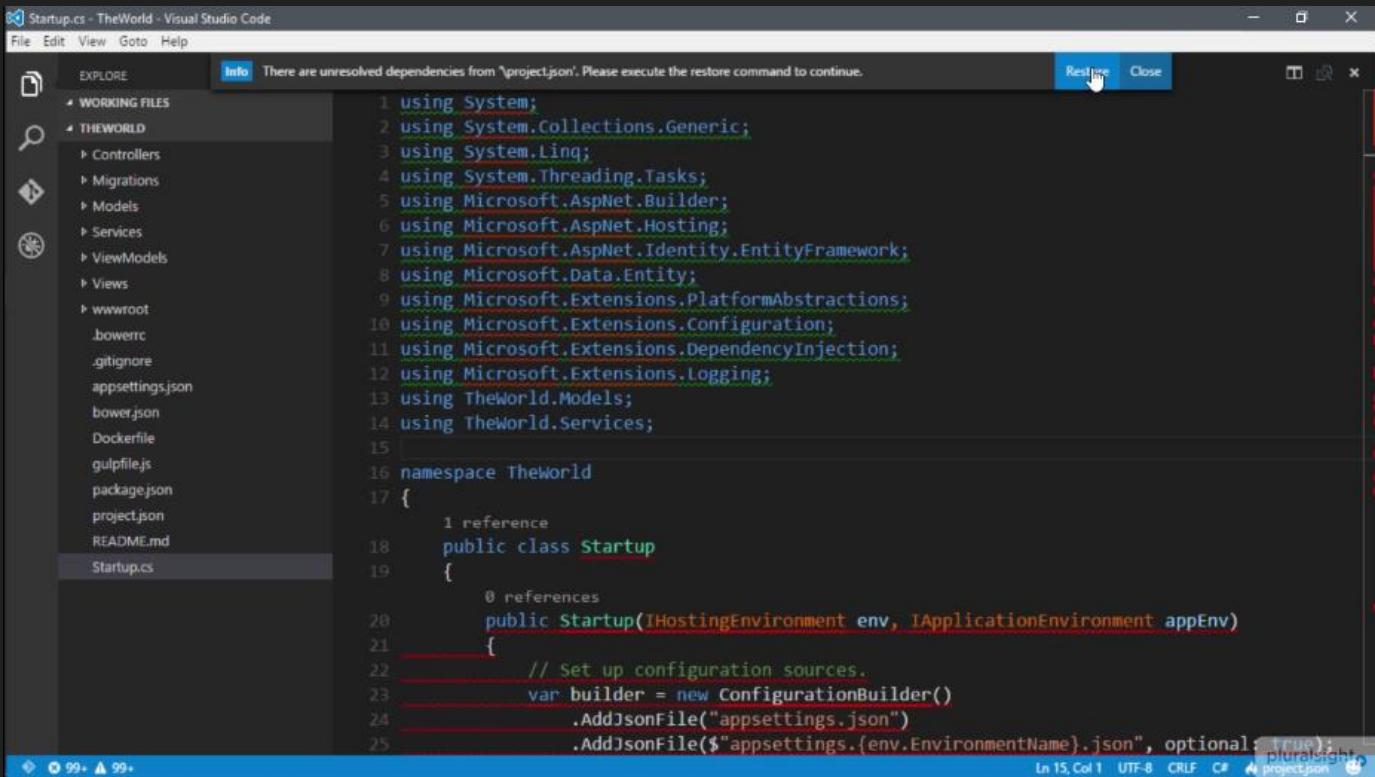
```
project.json
{
  "name": "Basic-Asp-Net-5-Web-Application-using-npm-generator-aspnet",
  "version": "0.0.1",
  "private": true,
  "dependencies": {
    "EntityFramework.Commands": "7.0.0-rc1-final",
    "EntityFramework.SQLite": "7.0.0-rc1-final",
    "EntityFramework.MicrosoftSqlServer": "7.0.0-rc1-final",
    "Microsoft.AspNet.Authentication.Cookies": "1.0.0-rc1-final",
    "Microsoft.AspNet.Diagnostics.Entity": "7.0.0-rc1-final",
    "Microsoft.AspNet.Identity.EntityFramework": "3.0.0-rc1-final",
    "Microsoft.AspNet.IISPlatformHandler": "1.0.0-rc1-final",
    "Microsoft.AspNet.Mvc": "6.0.0-rc1-final",
    "Microsoft.AspNet.Mvc.TagHelpers": "6.0.0-rc1-final",
    "Microsoft.AspNet.Server.Kestrel": "1.0.0-rc1-final",
    "Microsoft.AspNet.StaticFiles": "1.0.0-rc1-final",
    "Microsoft.AspNet.Tooling.Razor": "1.0.0-rc1-final",
    "Microsoft.Dnx.Runtime": "1.0.0-rc1-final",
    "Microsoft.Extensions.CodeGenerators.Mvc": "1.0.0-rc1-final",
    "Microsoft.Extensions.Configuration.FileProviderExtensions": "1.0.0-rc1-final",
    "Microsoft.Extensions.Configuration.Json": "1.0.0-rc1-final",
    "Microsoft.Extensions.Configuration.UserSecrets": "1.0.0-rc1-final",
    "Microsoft.Extensions.Logging": "1.0.0-rc1-final",
    "Microsoft.Extensions.Logging.Console": "1.0.0-rc1-final",
    "Microsoft.Extensions.Logging.Debug": "1.0.0-rc1-final"
  },
  "commands": {
    "web": "Microsoft.AspNet.Server.Kestrel",
    "ef": "EntityFramework.Commands"
  },
  "frameworks": {
    "dnx451": {},
    "dnxcore50": {}
  }
}
```

- Verificar archivos base de la aplicación web MVC:
 - Project.json
 - Startup.cs
 - Bower.json
- Verificar carpetas y archivos base de la aplicación web MVC:
 - Controllers
 - Models
 - ViewModels
 - Views
 - wwwroot



Usando aplicaciones MVC plantilla con DNX y Visual Studio Code

7. Si te aparece en Visual Studio Code líneas rojas, hay dependencias no resueltas que requieren ser resueltas por NuGet. Hacer click en “Restore”:



The screenshot shows the Visual Studio Code interface with the file 'Startup.cs' open. The code editor displays the following C# code:

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Threading.Tasks;
5 using Microsoft.AspNetCore.Builder;
6 using Microsoft.AspNetCore.Hosting;
7 using Microsoft.AspNetCore.Identity.EntityFrameworkCore;
8 using Microsoft.Data.Entity;
9 using Microsoft.Extensions.PlatformAbstractions;
10 using Microsoft.Extensions.Configuration;
11 using Microsoft.Extensions.DependencyInjection;
12 using Microsoft.Extensions.Logging;
13 using TheWorld.Models;
14 using TheWorld.Services;
15
16 namespace TheWorld
17 {
18     public class Startup
19     {
20         public Startup(IHostingEnvironment env, IApplicationBuilder appEnv)
21         {
22             // Set up configuration sources.
23             var builder = new ConfigurationBuilder()
24                 .AddJsonFile("appsettings.json")
25                 .AddJsonFile($"appsettings.{env.EnvironmentName}.json", optional: true);
26         }
27     }
28 }
```

A status bar at the top of the code editor says: "Info There are unresolved dependencies from 'project.json'. Please execute the restore command to continue." A blue "Restore" button is highlighted with a mouse cursor. The sidebar on the left shows the project structure with files like 'Startup.cs', 'appsettings.json', 'bower.json', 'Dockerfile', etc.

Internamente, el botón “Restore” ejecuta la herramienta runtime “dnu restore”

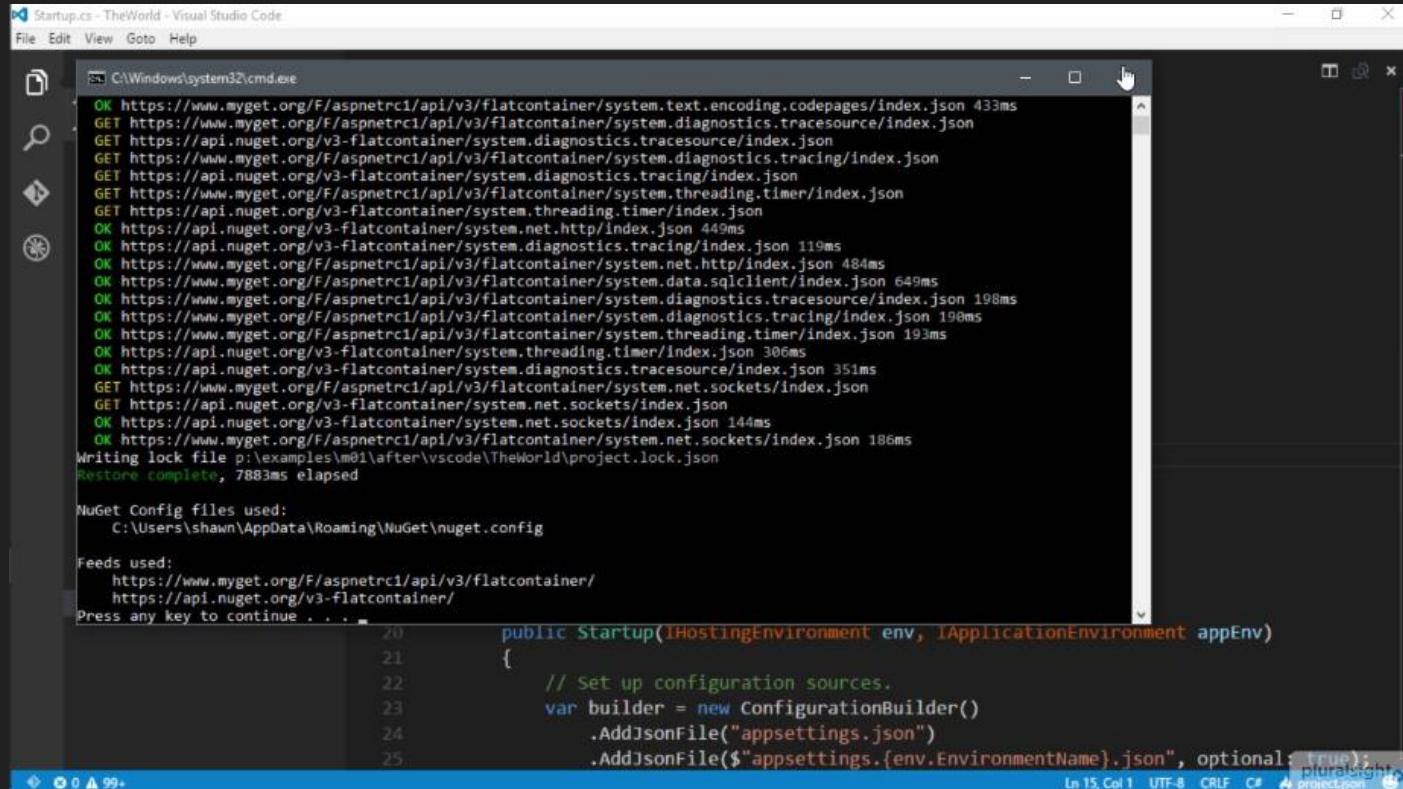
> Dnu restore

- DNU (Development Utilities) es otra herramienta de línea de comandos.
- Sirve de utilitario para administrar y restaurar paquetes de dependencias para aplicaciones ASP.NET 5 en adelante.



Usando aplicaciones MVC plantilla con DNX y Visual Studio Code

7. Si te aparece en Visual Studio Code líneas rojas, hay dependencias no resueltas que requieren ser resueltas por NuGet. Hacer click en “Restore”:



The terminal window shows the output of a NuGet restore command:

```
Startup.cs - TheWorld - Visual Studio Code
File Edit View Goto Help
C:\Windows\system32\cmd.exe
OK https://www.myget.org/F/aspnetrc1/api/v3/flatcontainer/system.text.encoding.codepages/index.json 433ms
GET https://www.myget.org/F/aspnetrc1/api/v3/flatcontainer/system.diagnostics.tracesource/index.json
GET https://api.nuget.org/v3-flatcontainer/system.diagnostics.tracesource/index.json
GET https://www.myget.org/F/aspnetrc1/api/v3/flatcontainer/system.diagnostics.tracing/index.json
GET https://api.nuget.org/v3-flatcontainer/system.diagnostics.tracing/index.json
GET https://www.myget.org/F/aspnetrc1/api/v3/flatcontainer/system.threading/thread/index.json
GET https://api.nuget.org/v3-flatcontainer/system.threading.thread/index.json
GET https://api.nuget.org/v3-flatcontainer/system.net.http/index.json 449ms
OK https://api.nuget.org/v3-flatcontainer/system.diagnostics.tracing/index.json 119ms
OK https://www.myget.org/F/aspnetrc1/api/v3/flatcontainer/system.net.http/index.json 484ms
OK https://www.myget.org/F/aspnetrc1/api/v3/flatcontainer/system.data.sqlclient/index.json 649ms
OK https://www.myget.org/F/aspnetrc1/api/v3/flatcontainer/system.diagnostics.tracesource/index.json 198ms
OK https://www.myget.org/F/aspnetrc1/api/v3/flatcontainer/system.diagnostics.tracing/index.json 190ms
OK https://www.myget.org/F/aspnetrc1/api/v3/flatcontainer/system.threading.timer/index.json 193ms
OK https://api.nuget.org/v3-flatcontainer/system.threading.timer/index.json 306ms
OK https://api.nuget.org/v3-flatcontainer/system.diagnostics.tracesource/index.json 351ms
GET https://www.myget.org/F/aspnetrc1/api/v3/flatcontainer/system.net.sockets/index.json
GET https://api.nuget.org/v3-flatcontainer/system.net.sockets/index.json
OK https://api.nuget.org/v3-flatcontainer/system.net.sockets/index.json 144ms
OK https://www.myget.org/F/aspnetrc1/api/v3/flatcontainer/system.net.sockets/index.json 186ms
Writing lock file p:\examples\m01\after\vscode\TheWorld\project.lock.json
Restore complete, 7883ms elapsed

NuGet Config files used:
  C:\Users\shawm\AppData\Roaming\NuGet\nuget.config

Feeds used:
  https://www.myget.org/F/aspnetrc1/api/v3/flatcontainer/
  https://api.nuget.org/v3-flatcontainer/
Press any key to continue . . .
```

The code editor window shows the `Startup.cs` file:

```
public Startup(IHostingEnvironment env, IApplicationBuilder appEnv)
{
    // Set up configuration sources.
    var builder = new ConfigurationBuilder()
        .AddJsonFile("appsettings.json")
        .AddJsonFile($"appsettings.{env.EnvironmentName}.json", optional: true);
    ...
```

Internamente, el botón “Restore” ejecuta la herramienta runtime “dnu restore”

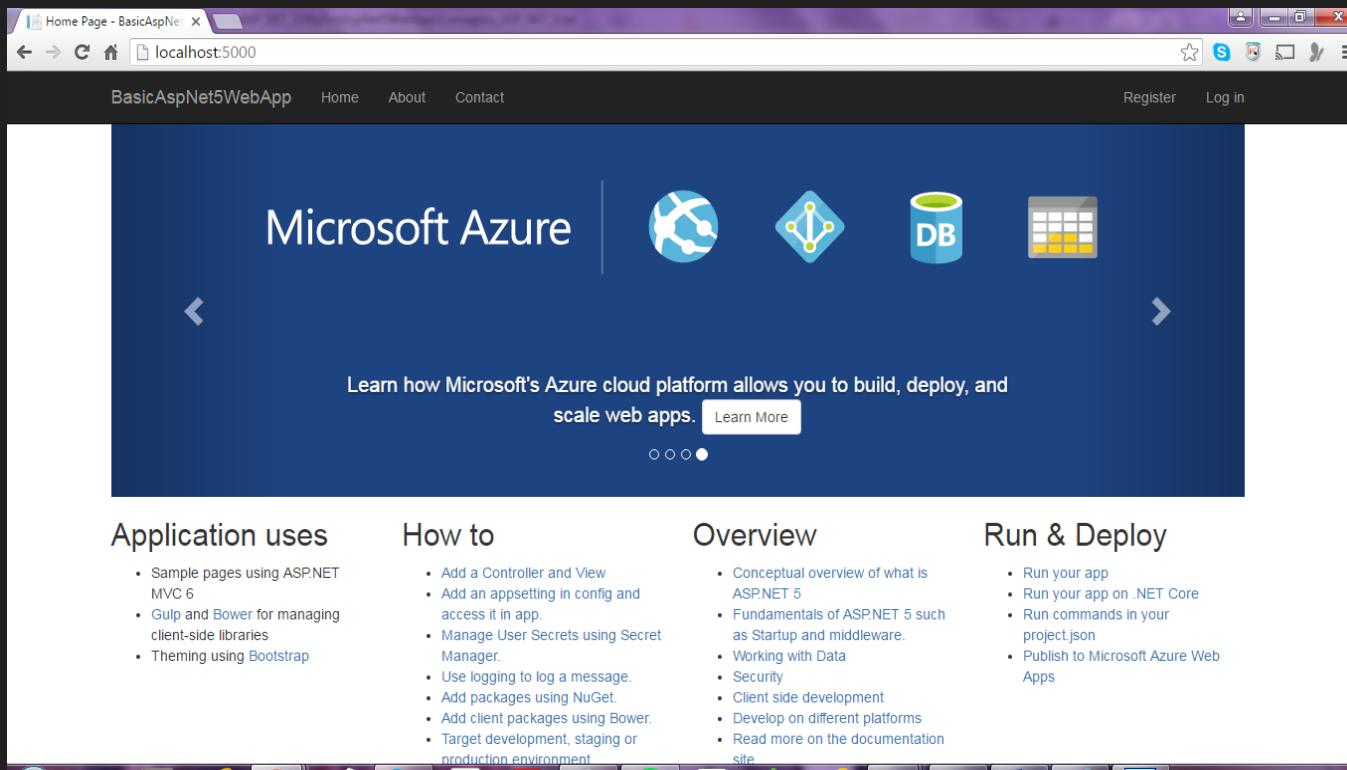
> Dnu restore

- DNU (Development Utilities) es otra herramienta de línea de comandos.
- Sirve de utilitario para administrar y restaurar paquetes de dependencias para aplicaciones ASP.NET 5 en adelante.



Usando aplicaciones MVC plantilla con DNX y Visual Studio Code

8. Finalmente, levantamos la aplicación web desde la ruta de la carpeta del proyecto con la siguiente instrucción:



> Dnx web

Con esta instrucción, levantamos el servidor de aplicaciones y el puerto correspondiente:

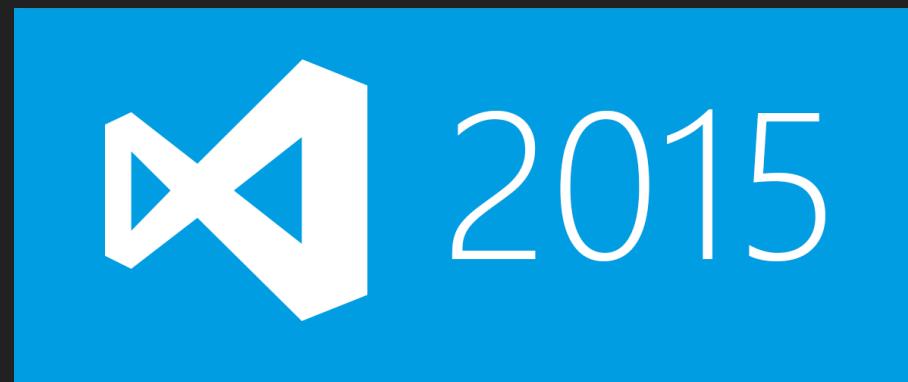
<http://localhost:5000>

(Este es el que trabaja Dnx web por defecto)

Tipeamos la URL en el navegador
Y se verá la aplicación plantilla

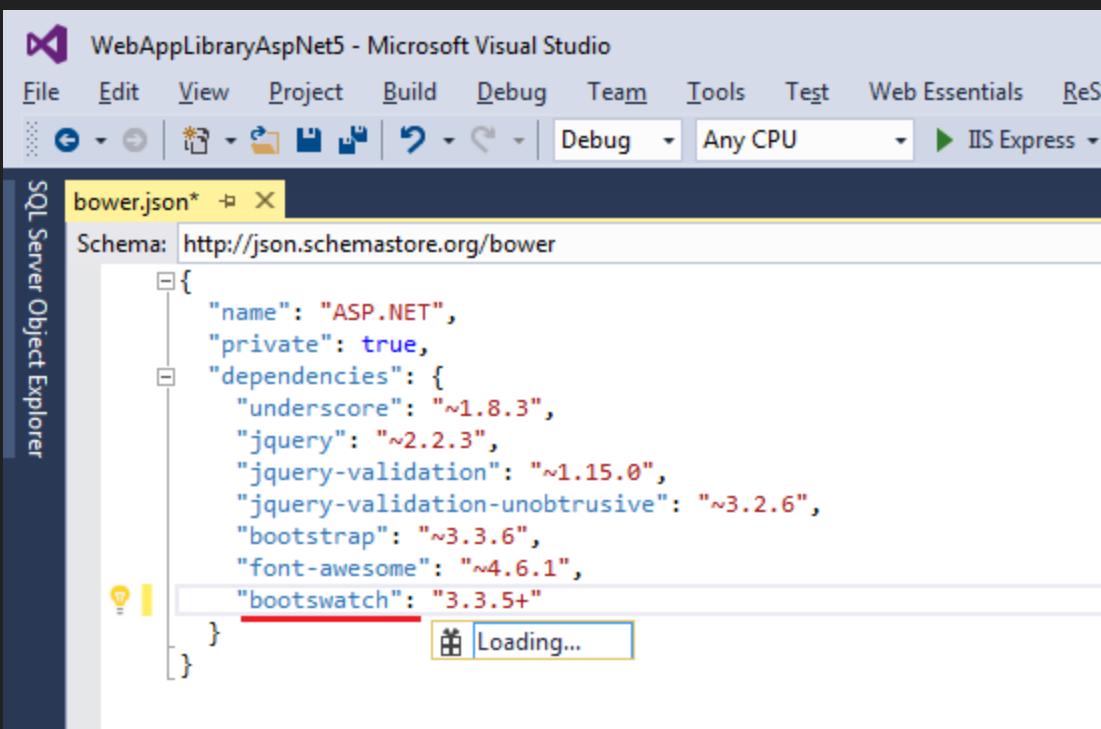


Y a partir de acá... todo es Visual Studio 2015

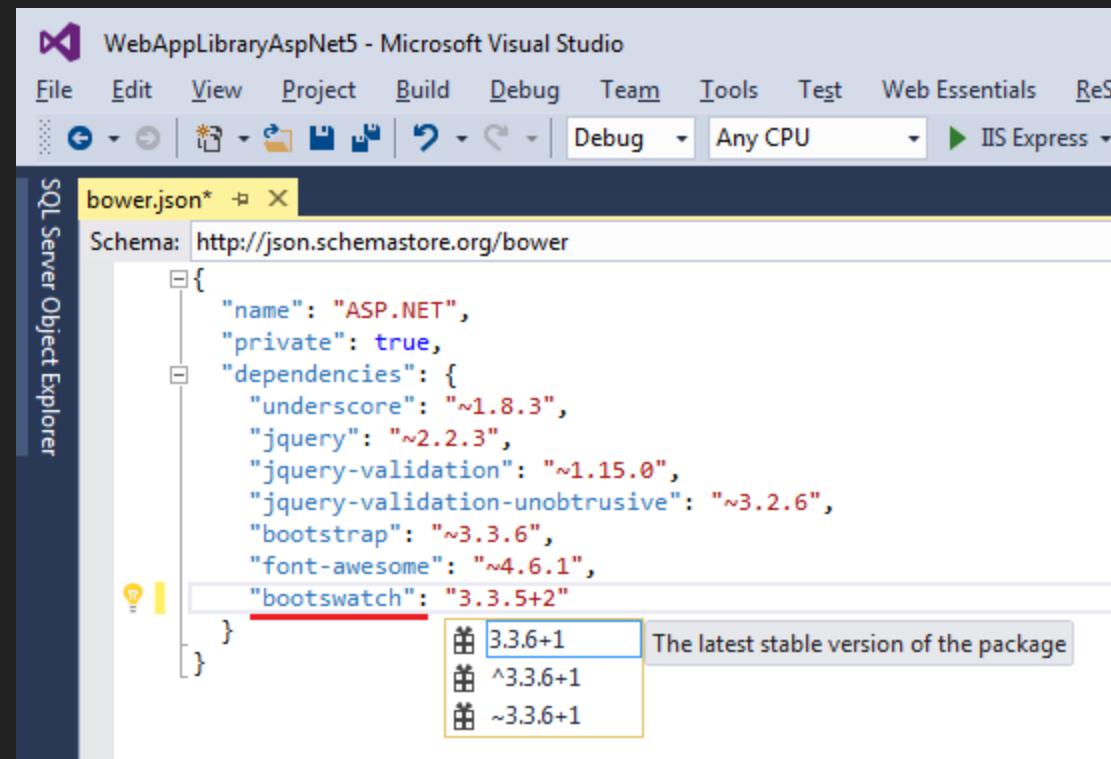


Visual Studio 2015

Intellisense + Bower: Adicionando dependencias por el lado del cliente



```
WebAppLibraryAspNet5 - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools Test Web Essentials ReShade
SQL Server Object Explorer
bower.json* -> X
Schema: http://json.schemastore.org/bower
{
  "name": "ASP.NET",
  "private": true,
  "dependencies": {
    "underscore": "~1.8.3",
    "jquery": "~2.2.3",
    "jquery-validation": "~1.15.0",
    "jquery-validation-unobtrusive": "~3.2.6",
    "bootstrap": "~3.3.6",
    "font-awesome": "~4.6.1",
    "bootswatch": "3.3.5+"
  }
}
Loading...
```



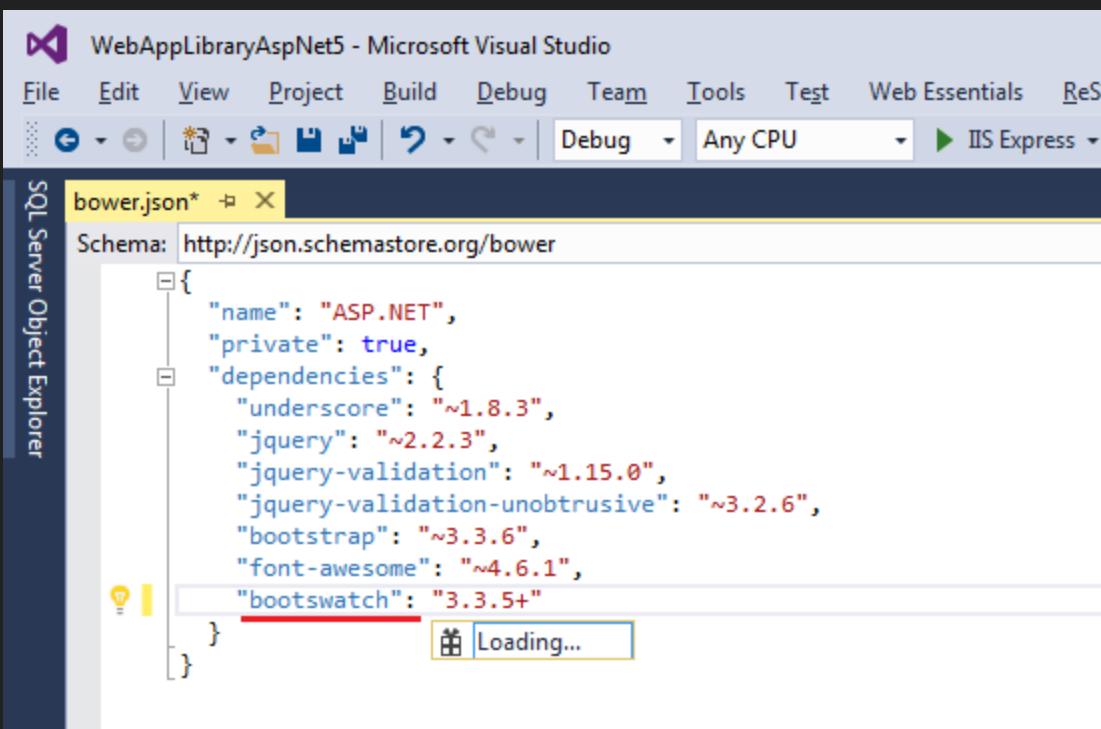
```
WebAppLibraryAspNet5 - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools Test Web Essentials ReShade
SQL Server Object Explorer
bower.json* -> X
Schema: http://json.schemastore.org/bower
{
  "name": "ASP.NET",
  "private": true,
  "dependencies": {
    "underscore": "~1.8.3",
    "jquery": "~2.2.3",
    "jquery-validation": "~1.15.0",
    "jquery-validation-unobtrusive": "~3.2.6",
    "bootstrap": "~3.3.6",
    "font-awesome": "~4.6.1",
    "bootswatch": "3.3.5+"
  }
}
3.3.6+1
^3.3.6+1
~3.3.6+1
The latest stable version of the package
```



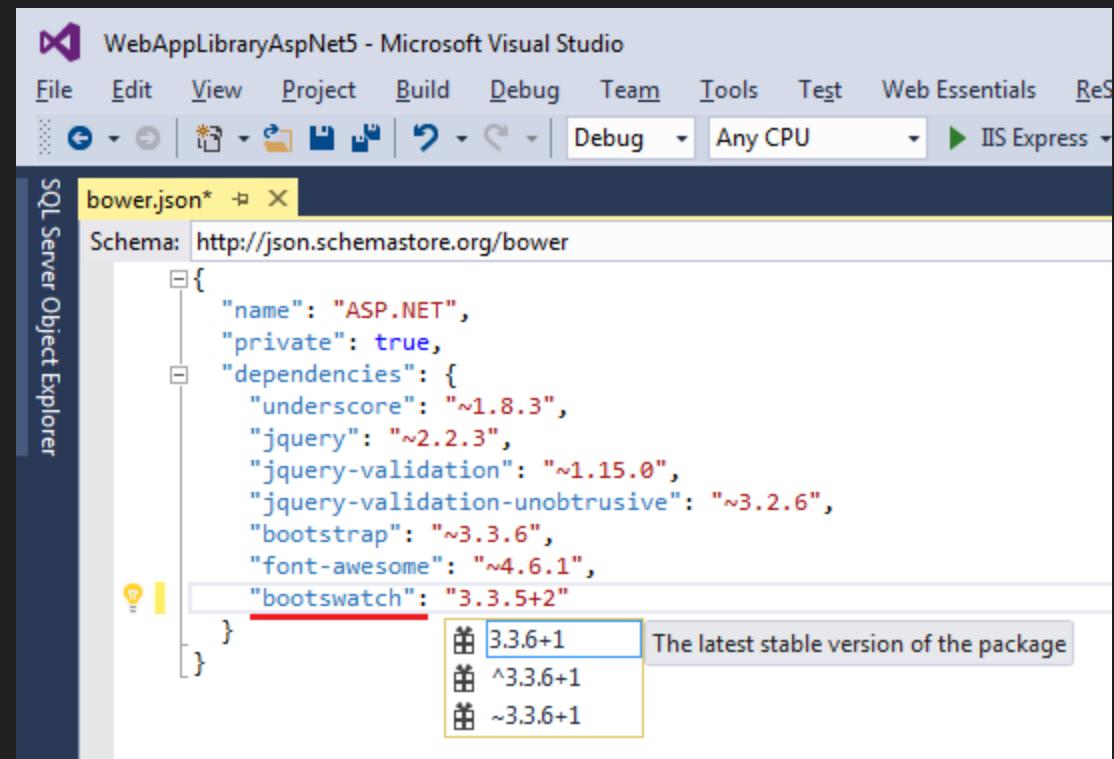
Microsoft



Intellisense + Bower: Adicionando dependencias por el lado del cliente



```
WebAppLibraryAspNet5 - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools Test Web Essentials ReShade
SQL Server Object Explorer
bower.json* -> X
Schema: http://json.schemastore.org/bower
{
  "name": "ASP.NET",
  "private": true,
  "dependencies": {
    "underscore": "~1.8.3",
    "jquery": "~2.2.3",
    "jquery-validation": "~1.15.0",
    "jquery-validation-unobtrusive": "~3.2.6",
    "bootstrap": "~3.3.6",
    "font-awesome": "~4.6.1",
    "bootswatch": "3.3.5+"
  }
}
Loading...
```



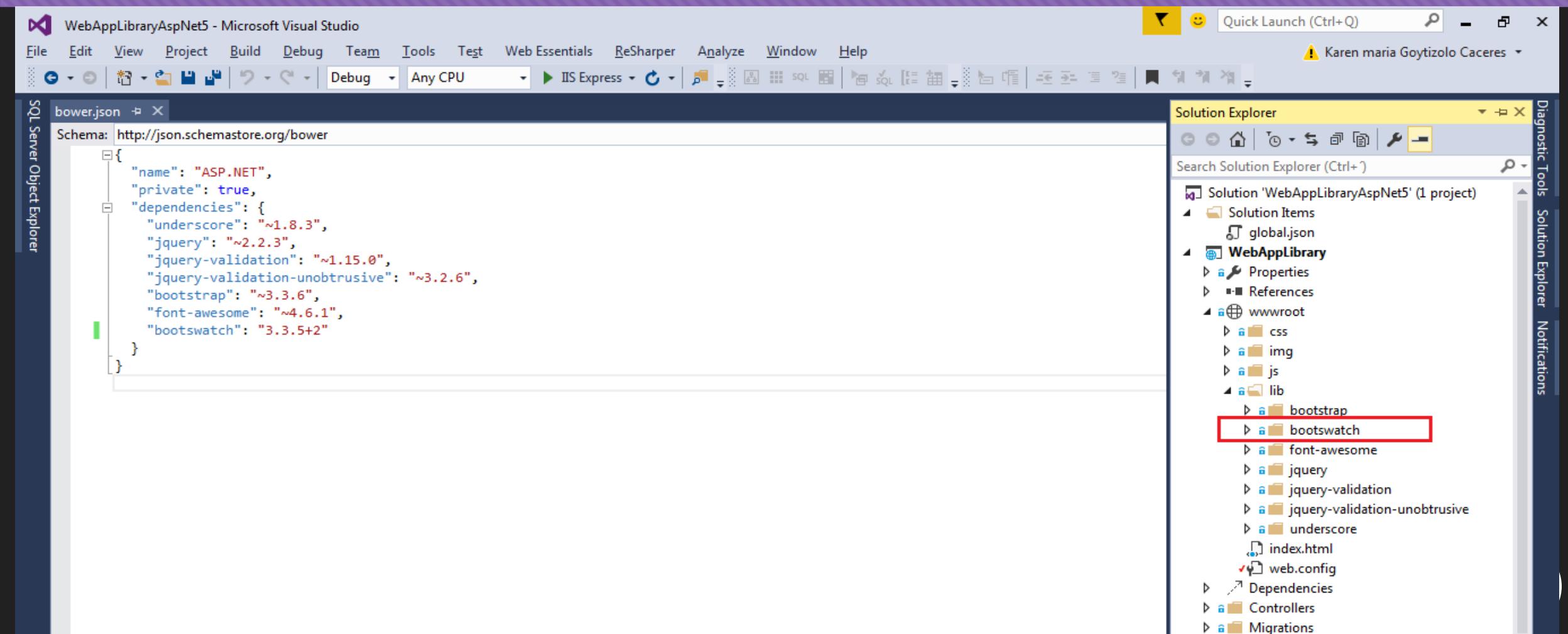
```
WebAppLibraryAspNet5 - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools Test Web Essentials ReShade
SQL Server Object Explorer
bower.json* -> X
Schema: http://json.schemastore.org/bower
{
  "name": "ASP.NET",
  "private": true,
  "dependencies": {
    "underscore": "~1.8.3",
    "jquery": "~2.2.3",
    "jquery-validation": "~1.15.0",
    "jquery-validation-unobtrusive": "~3.2.6",
    "bootstrap": "~3.3.6",
    "font-awesome": "~4.6.1",
    "bootswatch": "3.3.5+"
  }
}
3.3.6+1
^3.3.6+1
~3.3.6+1
The latest stable version of the package
```



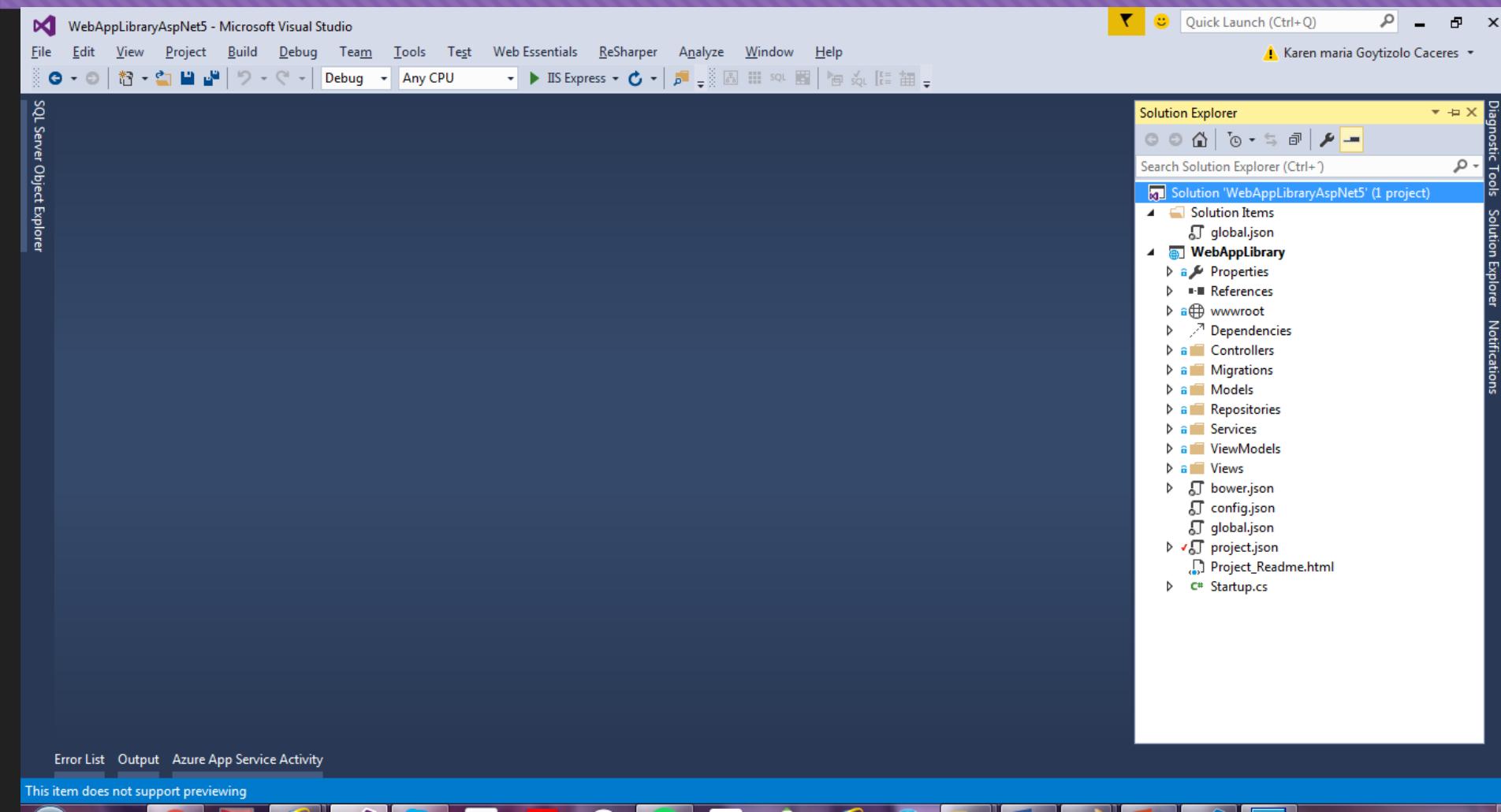
Microsoft



Intellisense + Bower: Adicionando dependencias por el lado del cliente



Estructura base de una aplicación web MVC 6



□ Jerarquía de aplicación

web MVC ASP.NET 5:

- Solución
- Proyecto
- Wwwwroot
- Controladores
 - Api
 - Web
- Repositories
- Services
- Migrations
- Models
- Viewmodels
- Views

Diseñando aplicaciones web y servicios web API RESTful con MVC 6



DEMO - Visual Studio 2015



□ Pasos para crear aplicaciones web MVC 6:

- Crea tus controladores.
- Crea tus vistas asociadas a los controladores.
- Crea una vista Shared (página compartida).
- Crea un layout.
- Uso de los tags (helpers).
- Uso de validaciones en el ViewModel
- Soportando métodos GET y POST
- Añade un servicio a tu método
- Personaliza tu formulario



Diseñando aplicaciones web y servicios web API RESTful con MVC 6



DEMO - Visual Studio 2015

□ Pasos para crear servicios web API en MVC 6:

- Crea tus controladores API.
- Verifica el retorno de datos con Postman.
- Crear métodos GET y POST
- Crea validaciones a nivel de View Models.
- Uso del AutoMapper (modelo > ViewModel).
- Implementa un servicio API externo



Aplicando Entity Framework 7 para armar tu base de datos



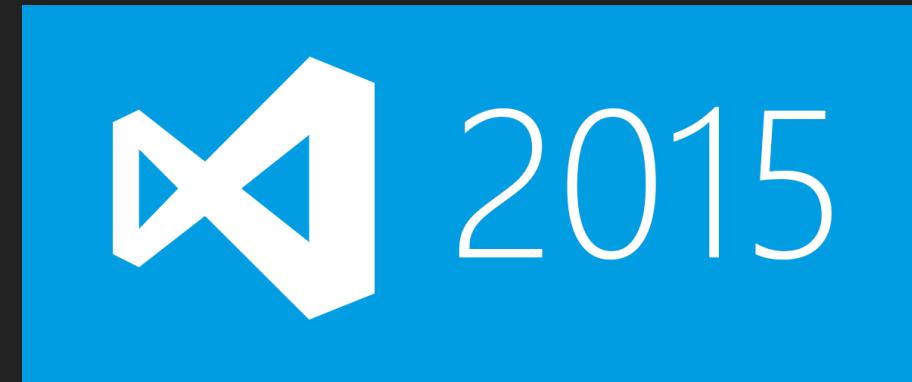
DEMO - Visual Studio 2015

□ Pasos para aplicar Entity Framework 7:

- Crea tus entidades dentro de la carpeta “Model”
- Las entidades se convertirán en tablas luego.
- Crea la clase DbContext
- Usa la clase DbContext dentro del controlador
- Usa las migraciones
- “Germina” la base de datos
- Uso del patrón Repository
- Uso extra de Logging



Uso de Angular JS para potenciar el lado del cliente



DEMO - Visual Studio 2015

Mejores prácticas para desarrollo de Aplicaciones Web ASP.NET

□ Aplicaciones multi-nivel:

- Entidades
- Servicios
- Datos (conexión)
- Lógica de negocio
- Integrador MVC (*)
- UI / front-end
- Testing back-end y front-end

- Inyección de dependencias
- Repositorios para Mockups / Testing
- Host de servicios no debe estar dentro de aplicación web. Esta solo la consume.
- Analizar cuando crear aplicaciones web con Web Forms, MVC, etc
- Analizar implementación de frameworks de diseño para front-end
- Bases de datos traen la data y realizan transacciones CRUD. Usar LINQ para aplicar lógica de negocio compleja.



Decídete por ASP.NET Web Forms si....

- Los usuarios de tu sistema no tienen sistemas operativos actualizados compatibles con las nuevas versiones de .NET Frameworks.
- Tu equipo es de pocos desarrolladores y requieren aplicaciones con entregables rápidos.
- Crear aplicaciones web con arquitectura robusta pero poco compleja



Decídete por ASP.NET MVC si....

- Soluciones de software más complejas a nivel de front-end y back-end
- Tu equipo es de varios desarrolladores y pueden cubrir requerimientos de aplicaciones escalables en el tiempo.
- Requieren realizar un manejo más profundo de pruebas (sobre todo TDD).
- Tu equipo está familiarizado con metodologías Scrum para los entregables de software.





Materiales de interés

Ejercicio del demo en GitHub:

<https://github.com/kgoytizolo/ASP-NET-5-Web-Apps>

Otros links:

- <https://john-dugan.com/visual-studio-code-vs-sublime-text/>
- <https://marketplace.visualstudio.com/search?term=Git&target=SCode&sortBy=Relevance>
- <https://code.visualstudio.com/docs?start=true>
- <https://code.visualstudio.com/Docs/?dv=win>
- <http://proudmonkey.azurewebsites.net/asp-net-5-getting-started-with-mvc-6/>
- <http://proudmonkey.azurewebsites.net/introducing-asp-net-5-the-new-asp-net-in-town/>
- <http://www.talkingdotnet.com/various-configuration-json-files-in-asp-net-5/>
- <http://www.talkingdotnet.com/whats-new-in-asp-net-5-rc-1/>

GRACIAS!!

