Crea aplicaciones web dinámicas con ASP.NET Core RC1, MVC 6, Entity Framework 7 y Angular JS

Microsoft

Por: Karen Goytizolo Cáceres

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.

Microsoft

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

Microsoft

¿Por qué ASP.NET?

Microsoft

¿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

Microsoft

¿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

Microsoft

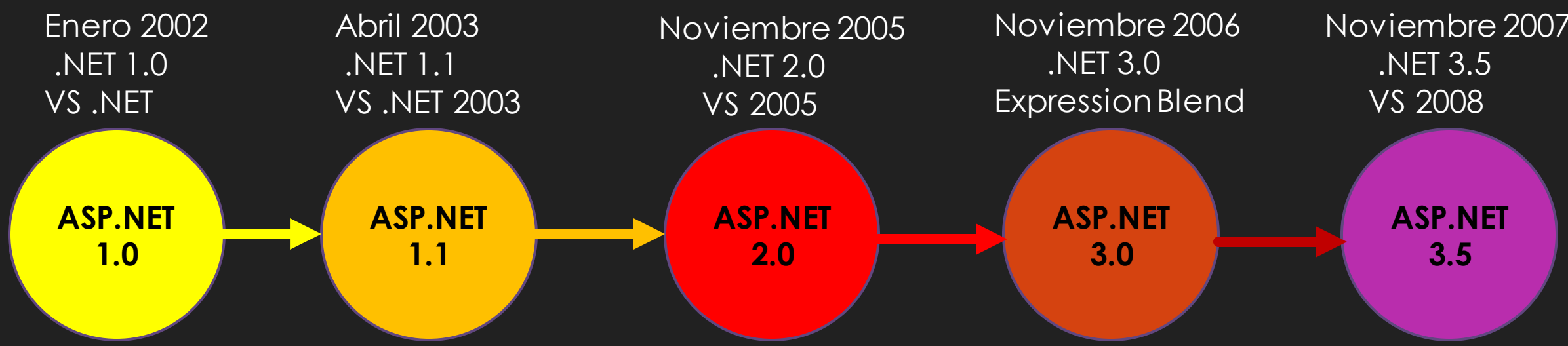
¿Por qué ASP.NET?

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

Microsoft

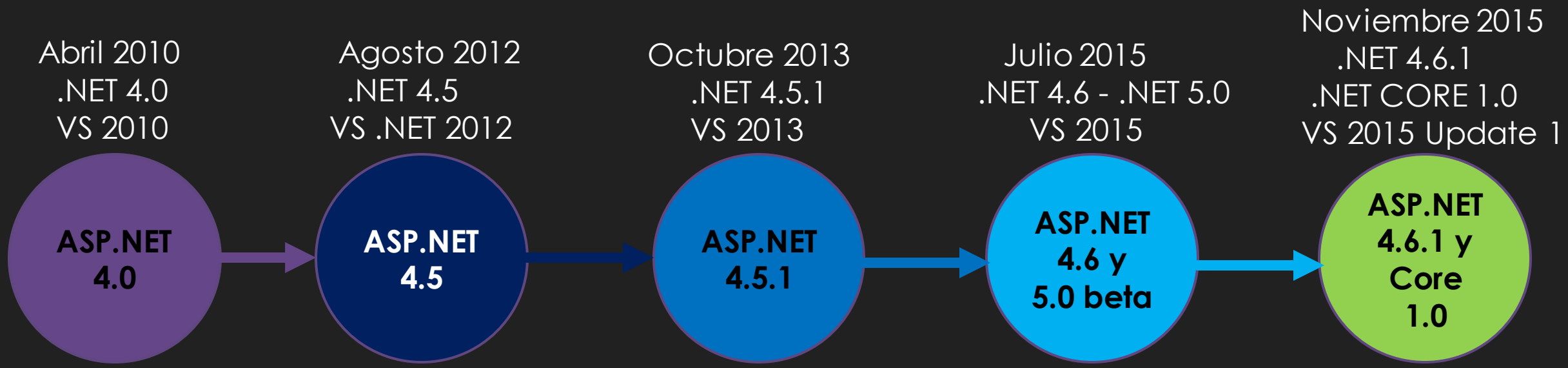
ASP.NET – 14 años de evolución



Microsoft

Fuente: https://en.wikipedia.org/wiki/.NET_Framework_version_history#cite_note-net45-48

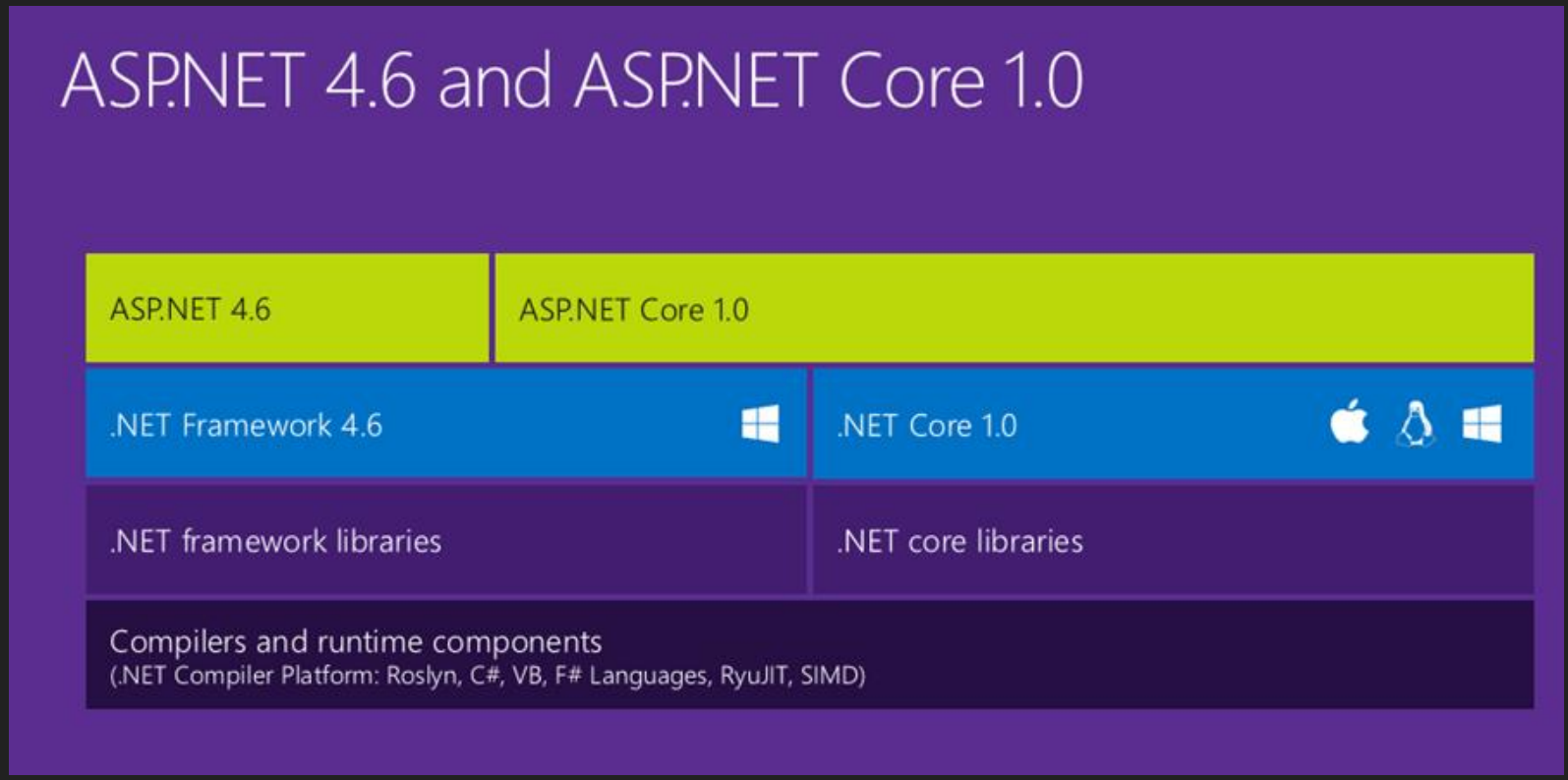
ASP.NET – 14 años de evolución



Microsoft

Fuente: https://en.wikipedia.org/wiki/.NET_Framework_version_history#cite_note-net45-48

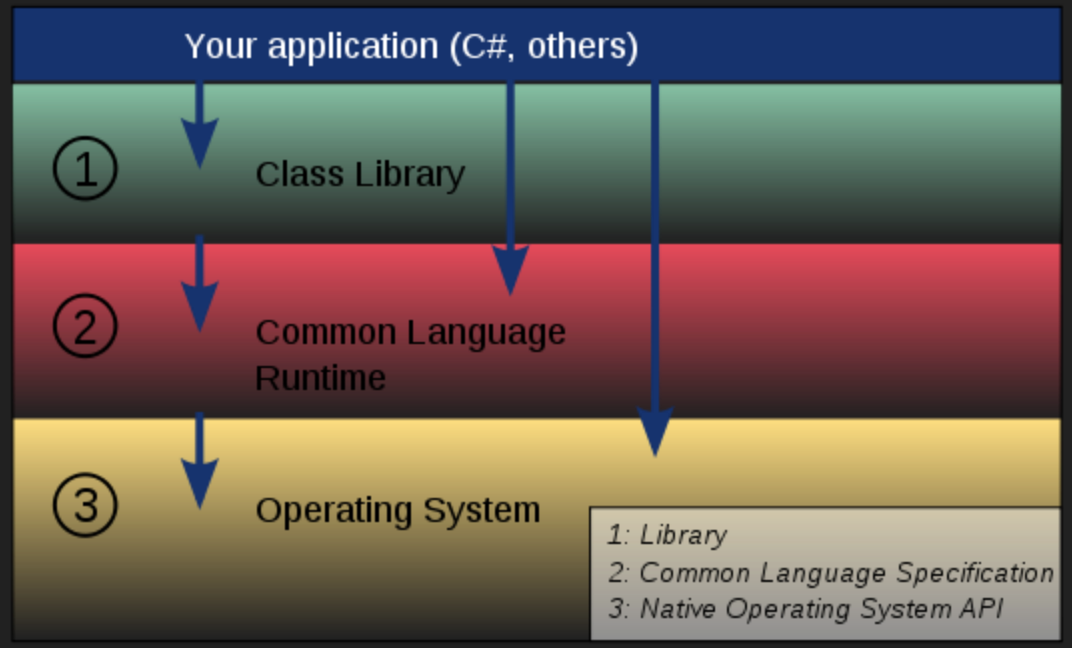
Comparando ASP.NET 4.6 y ASP.NET CORE 1.0



Microsoft

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

¿NET y ASPNET 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

Fuente: http://www.mono-project.com/docs/web/aspnet/

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

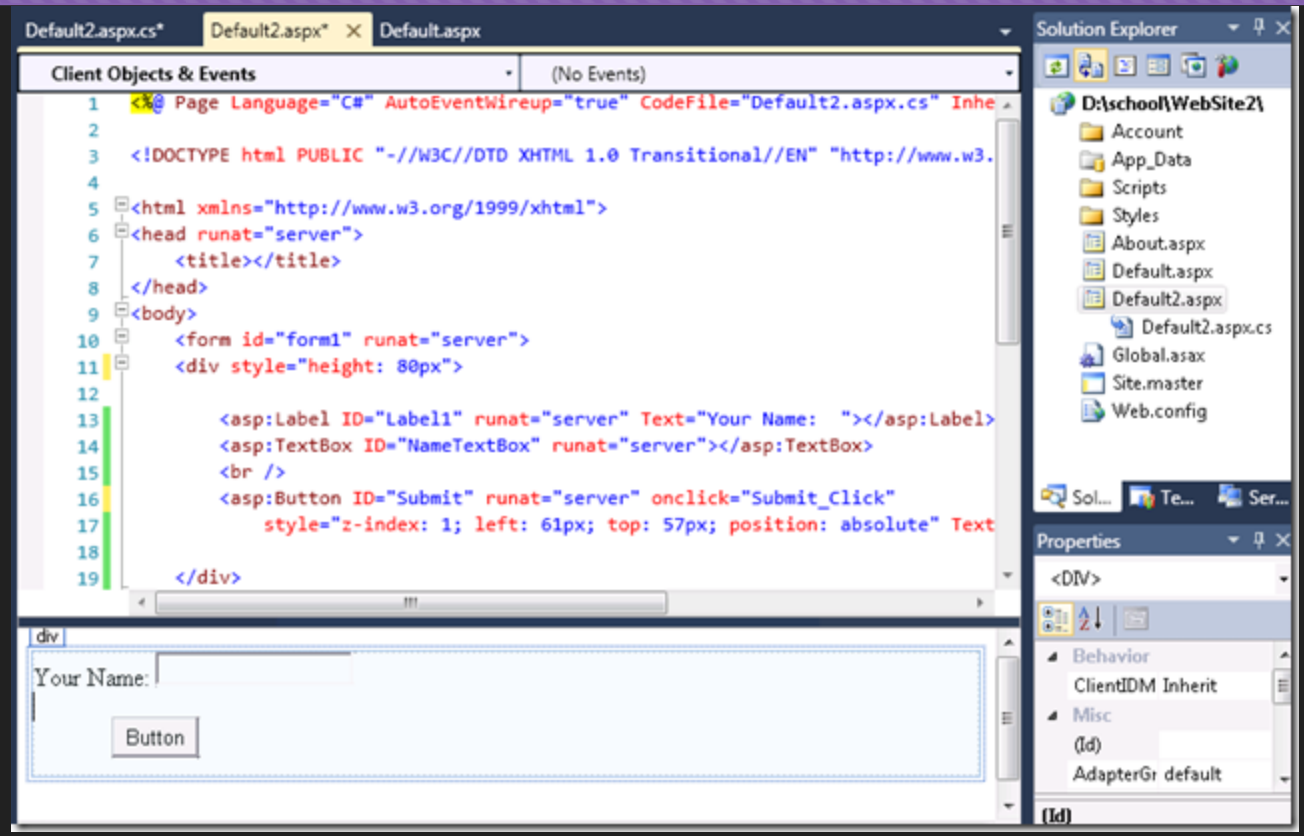
Extensiones

- 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



Microsoft

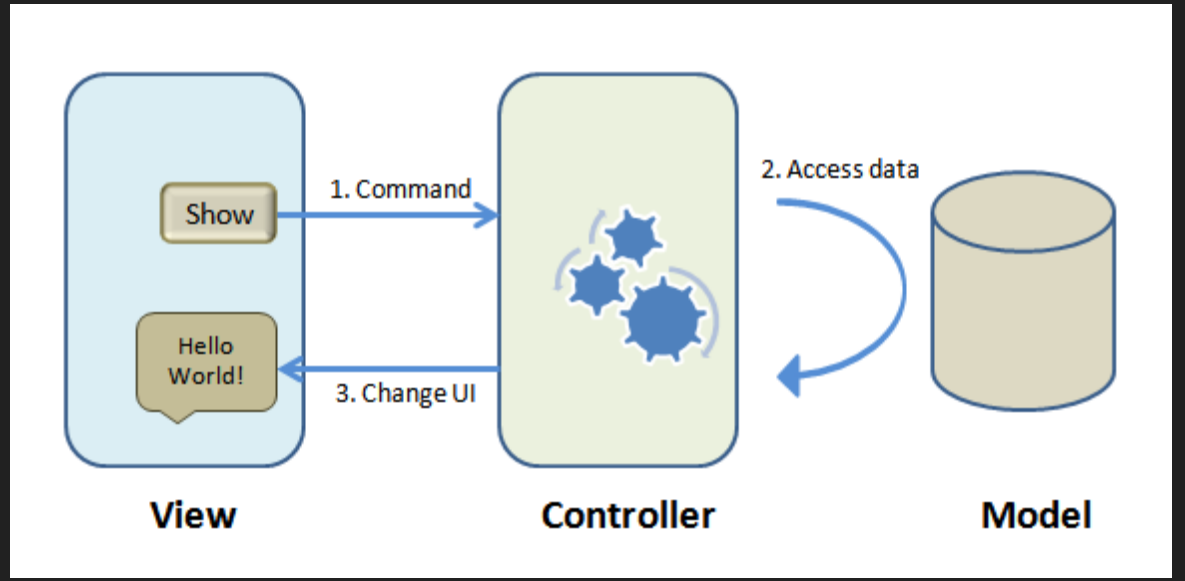
Trabajando con ASP.NET Web Forms

- 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

Microsoft

Trabajando con ASP.NET MVC

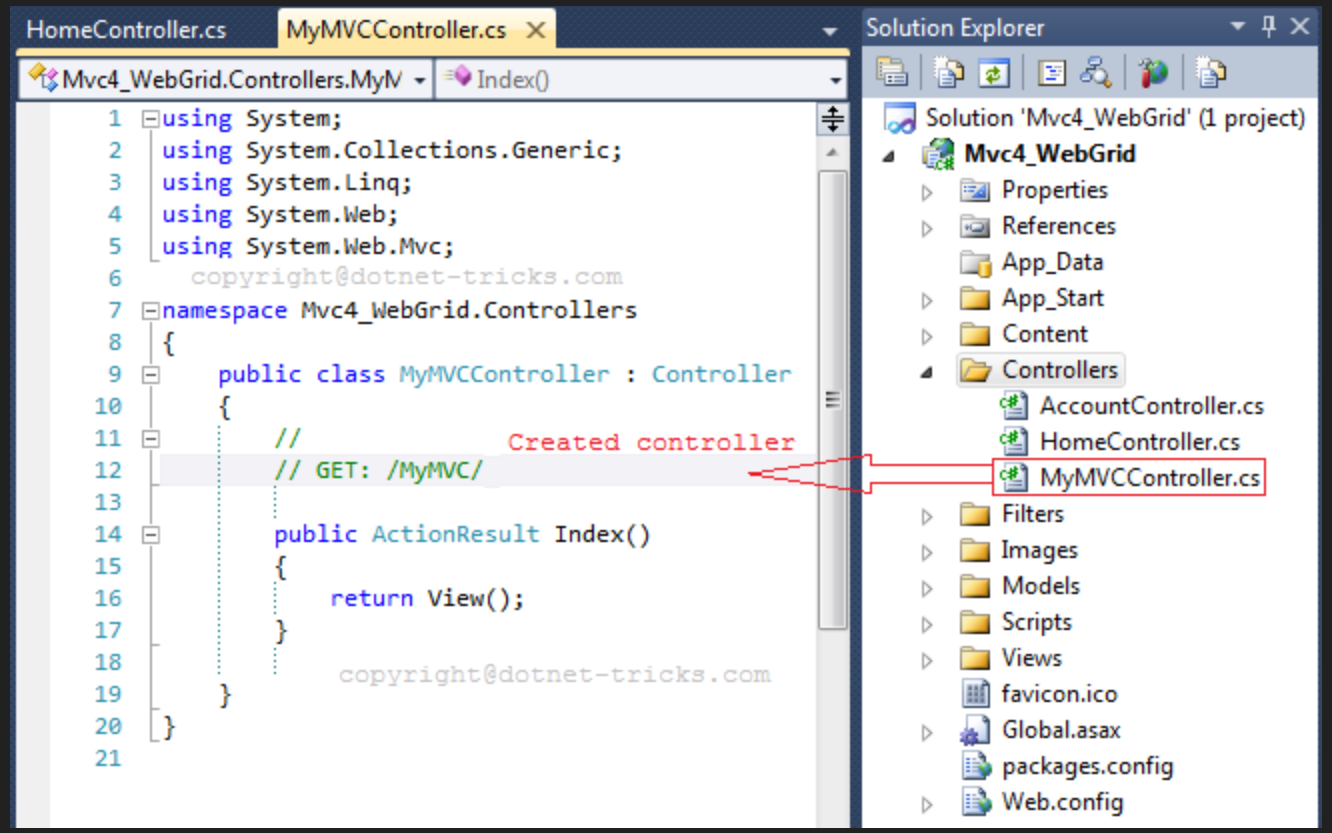
- Patrón Modelo – Vista - Controlador
- Modelo:
  - Reglas de negocio
  - Datos devueltos / Acceso a datos
- Vista: Presentación de la aplicación
- Controlador: Intermediario entre
  - Usuario > Modelo > Vistas



Microsoft

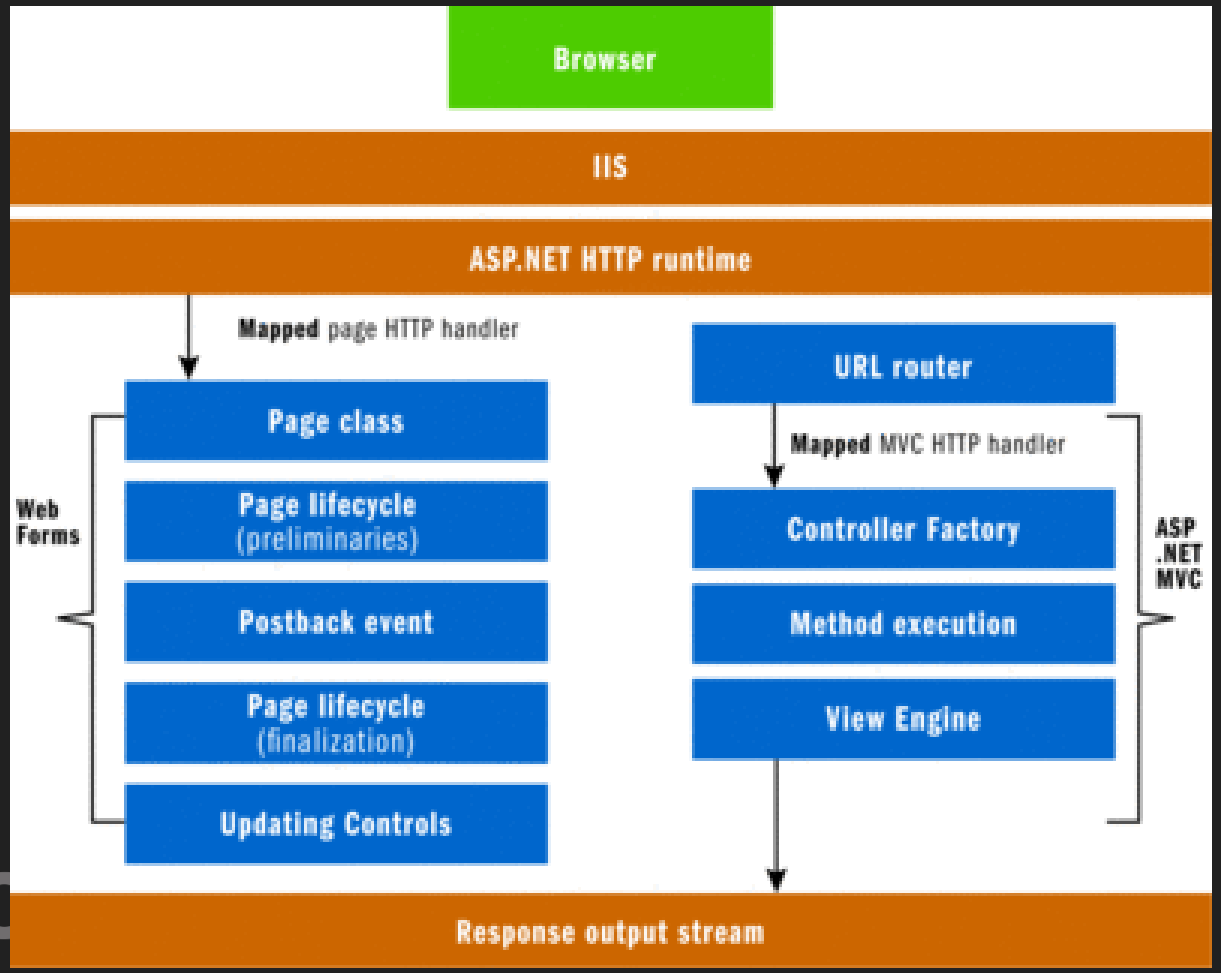
Trabajando con ASP.NET MVC

- 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



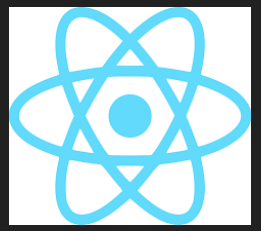
Microsoft

Arquitectura Web Forms vs MVC



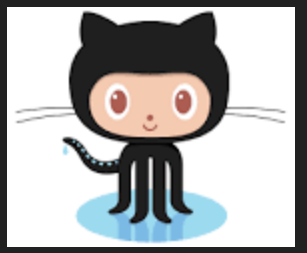
Sinergia entre ASP.NET web applications y Opensource

Front-end



Integración NPM y manejo de Dependencias

Repositorios



Bases de Datos No Sql

Microsoft

http://nosql-database.org/

Empezando a meter mano en el código (Demo)

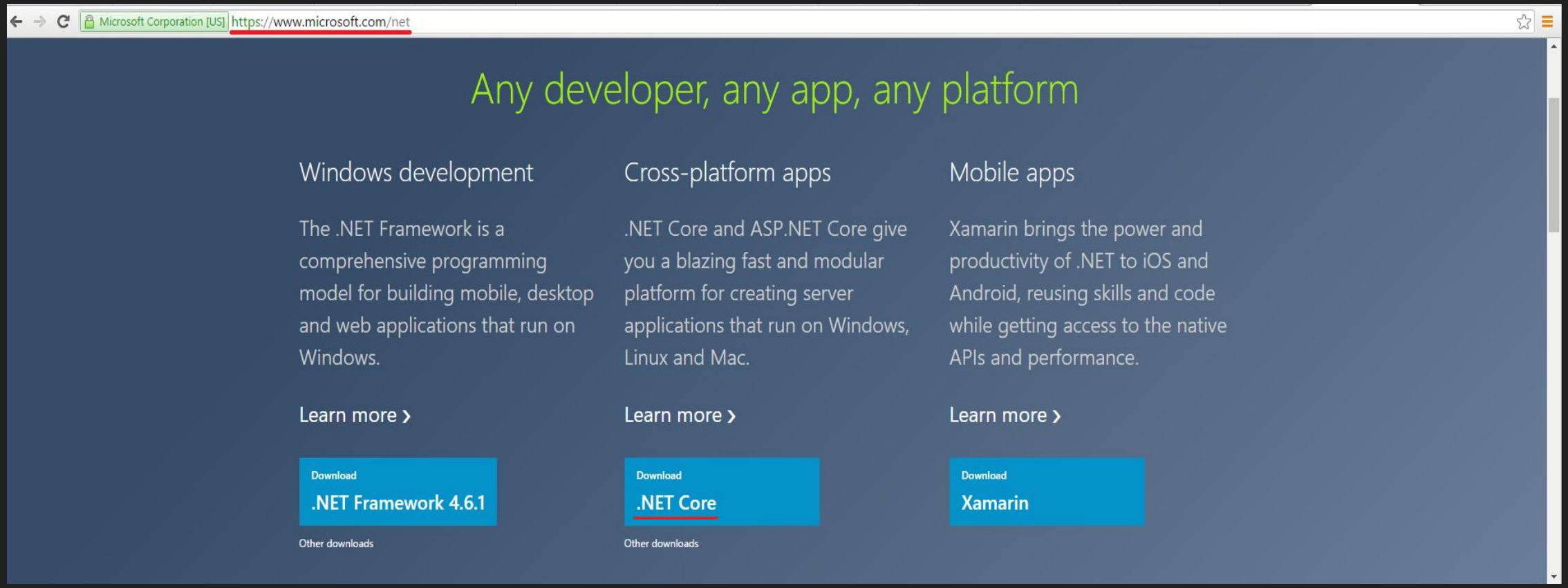
Visual Studio 2015

Visual Studio Code

Microsoft

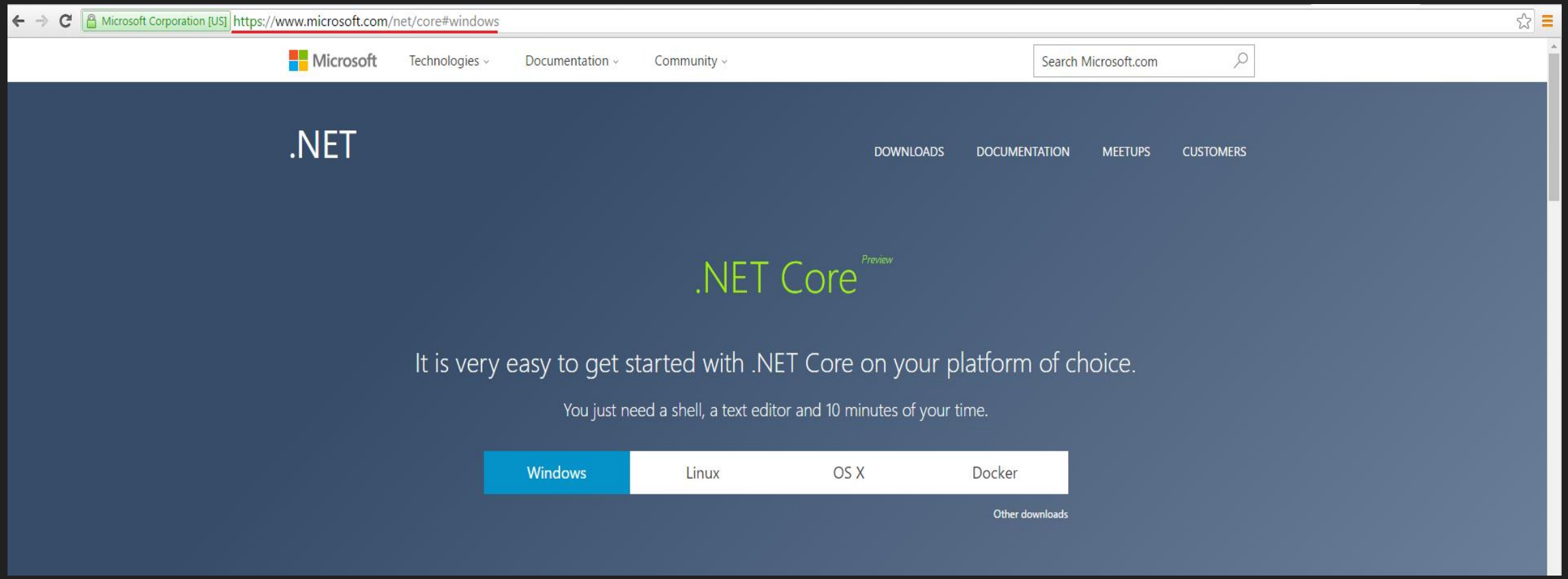
¿Por donde empezar? Instalando ASP.NET 5 y nuevos conceptos

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



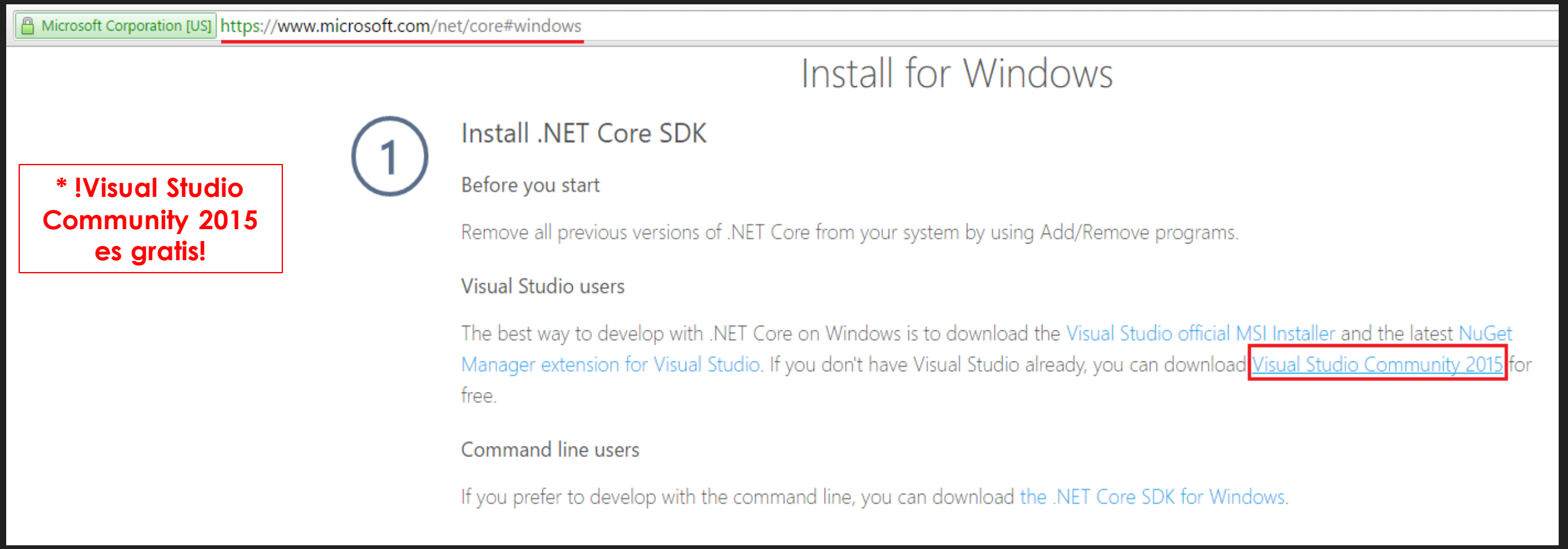
¿Por donde empezar? Instalando ASP.NET 5 y nuevos conceptos

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



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

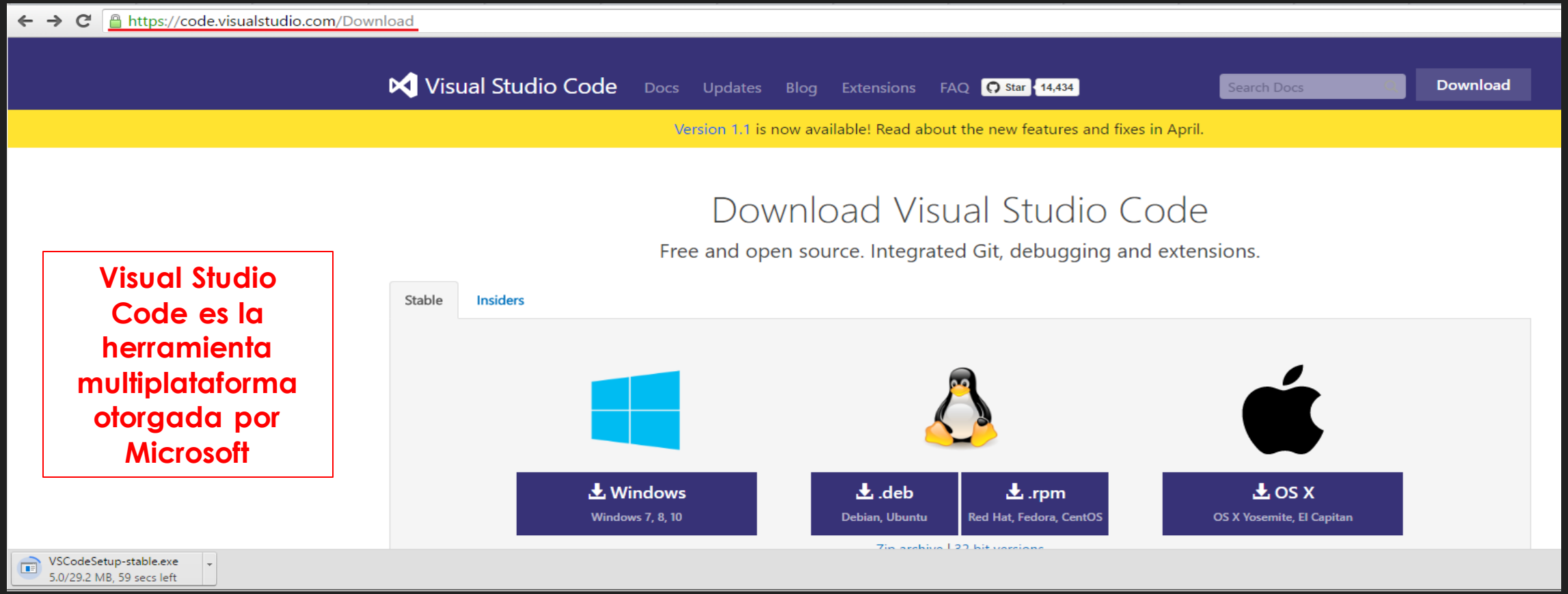
3. Si usas Windows, antes descarga e instala *Visual Studio Community 2015*:



*** !Visual Studio Community 2015 es gratis!**

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

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

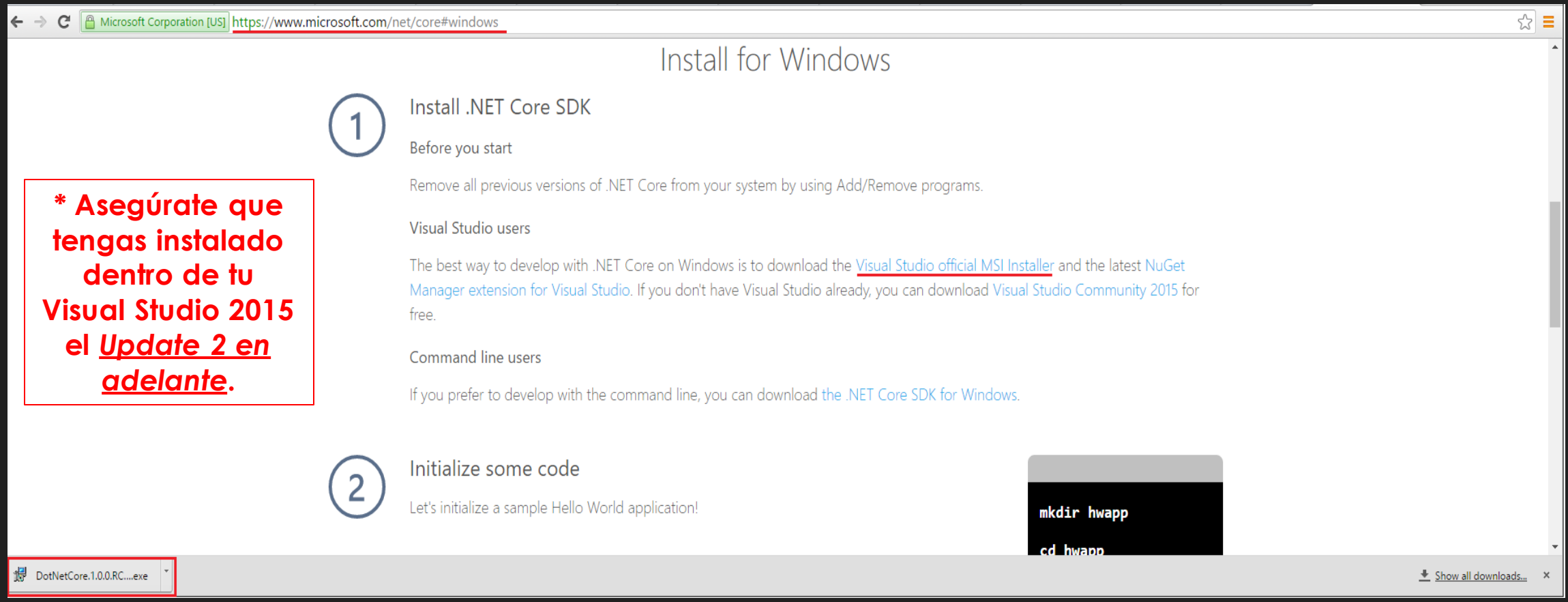


Visual Studio Code es la herramienta multiplataforma otorgada por Microsoft

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

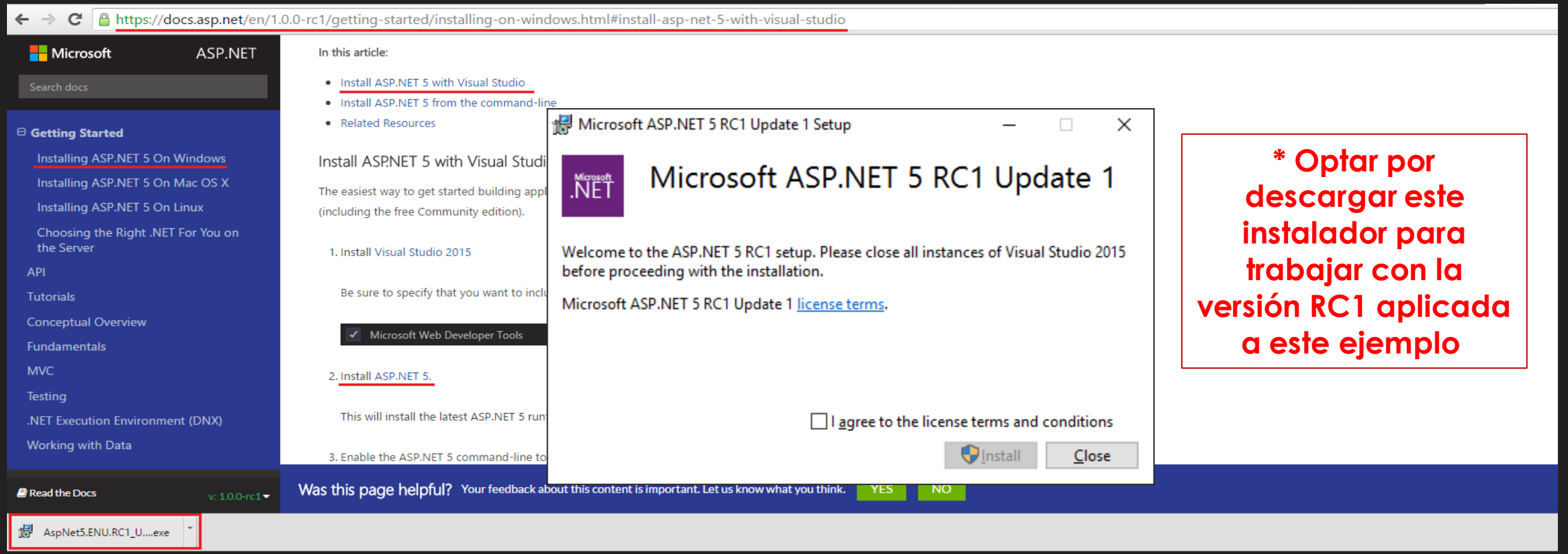
5. Descargar e instalar ASP.NET 5 (.NET Core) - versión actual RC2 :

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



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

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

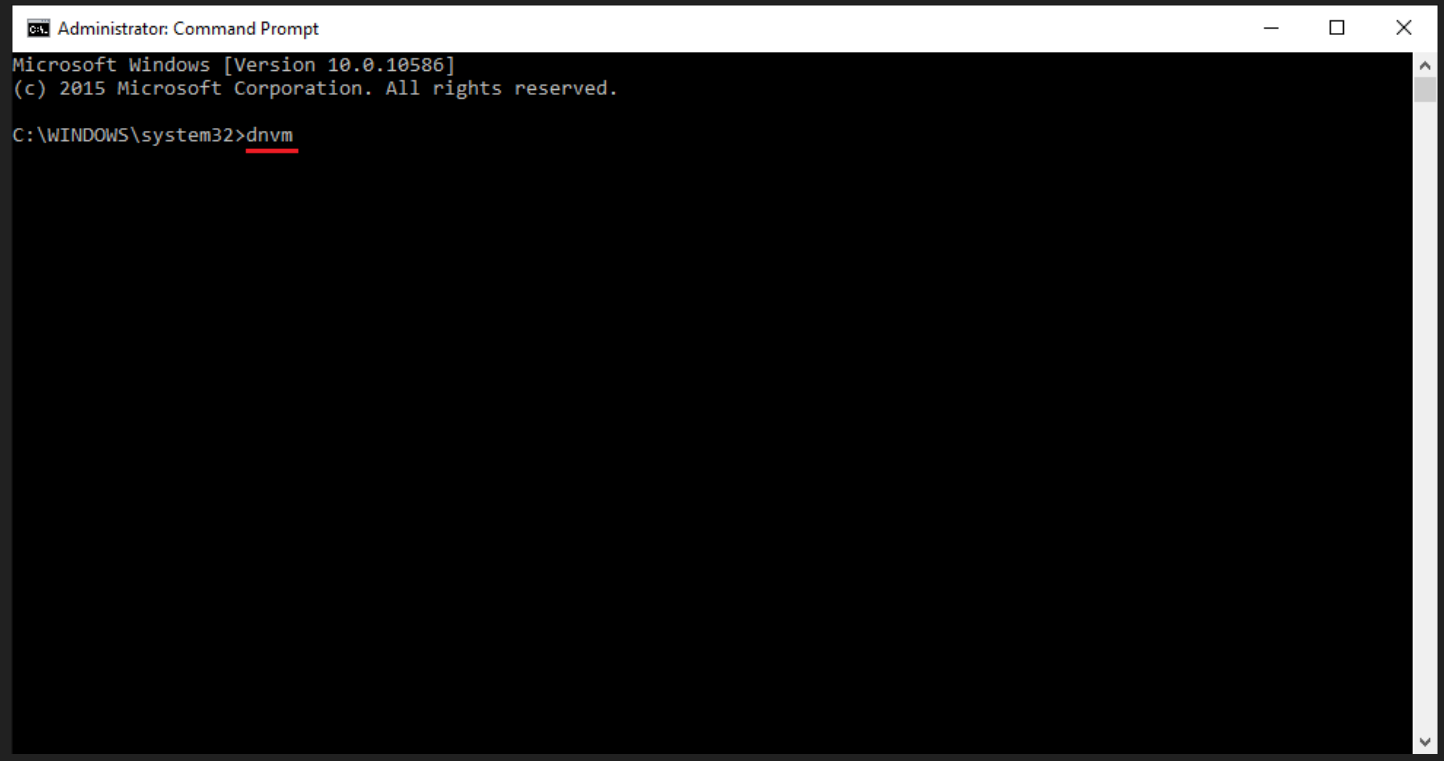


*** Optar por descargar este instalador para trabajar con la versión RC1 aplicada a este ejemplo**

¿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:

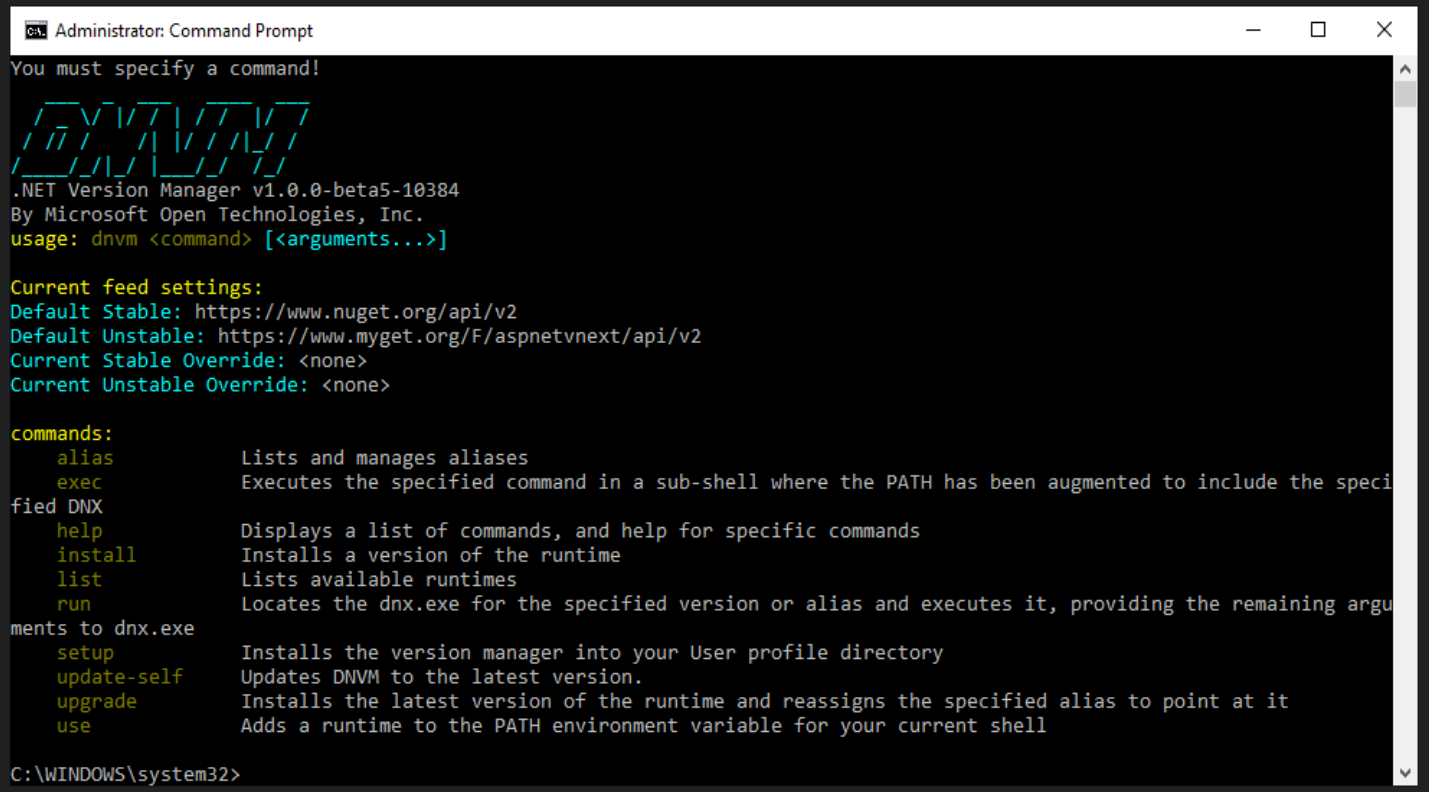
> dnvm



Microsoft

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

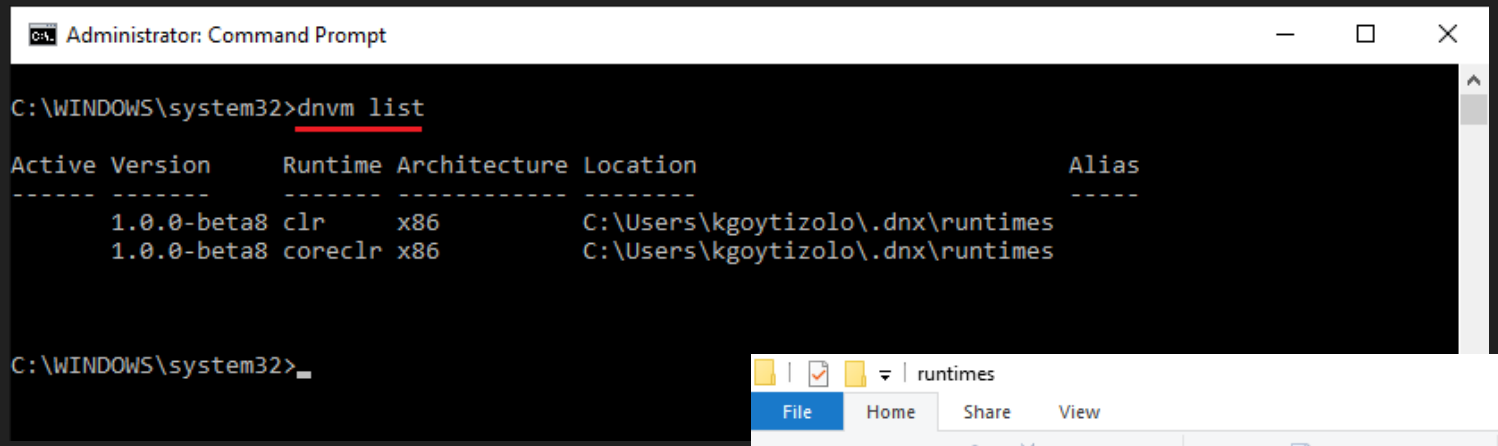
8. Esto es nuevo en el mundo de ASP.NET… ¿Qué es el DNVM?:



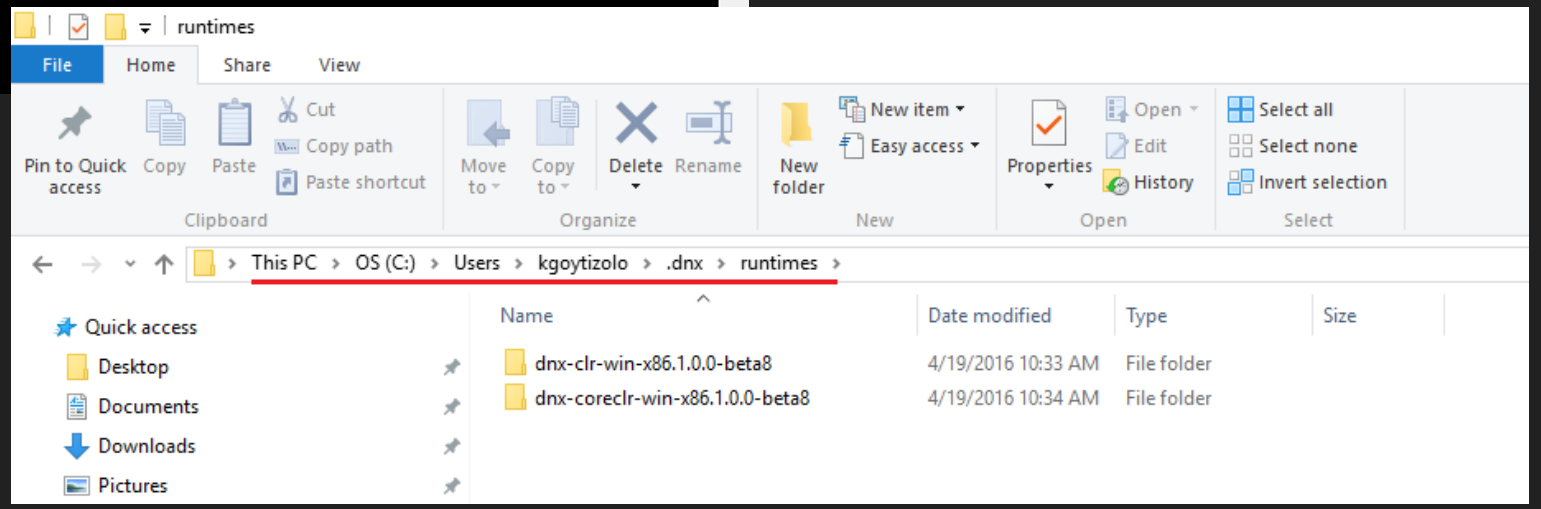
- DNVM es el Administrador de versiones de ASP.NET 5 DNX en adelante.
- Es un nuevo modelo de herramienta runtime (usando línea de comandos), junto con DNX y DNU.
- Permite seleccionar la versión de DNX .NET con la que se trabajará.
- Si te aparece esta pantalla, quiere decir que el *framework ASP.NET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



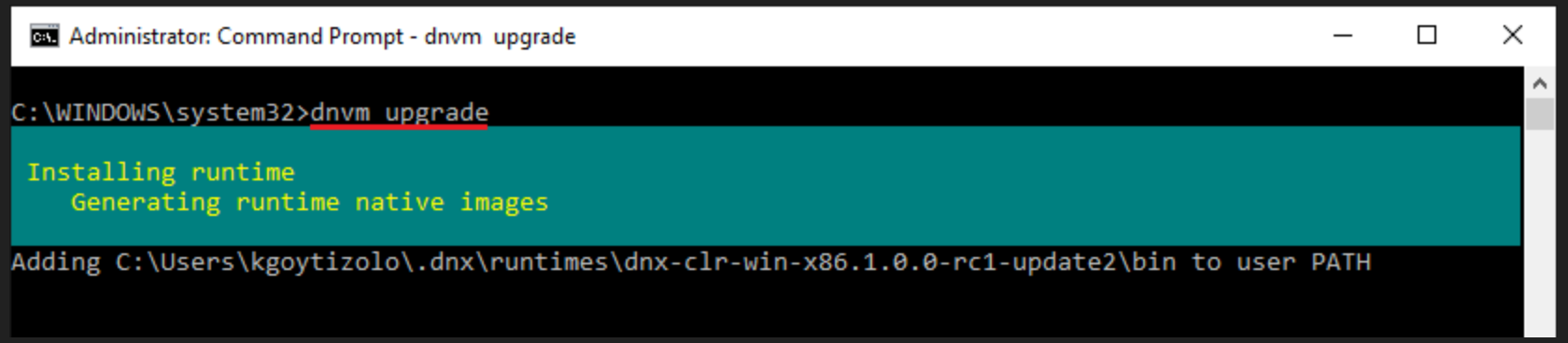
> Dnvm list

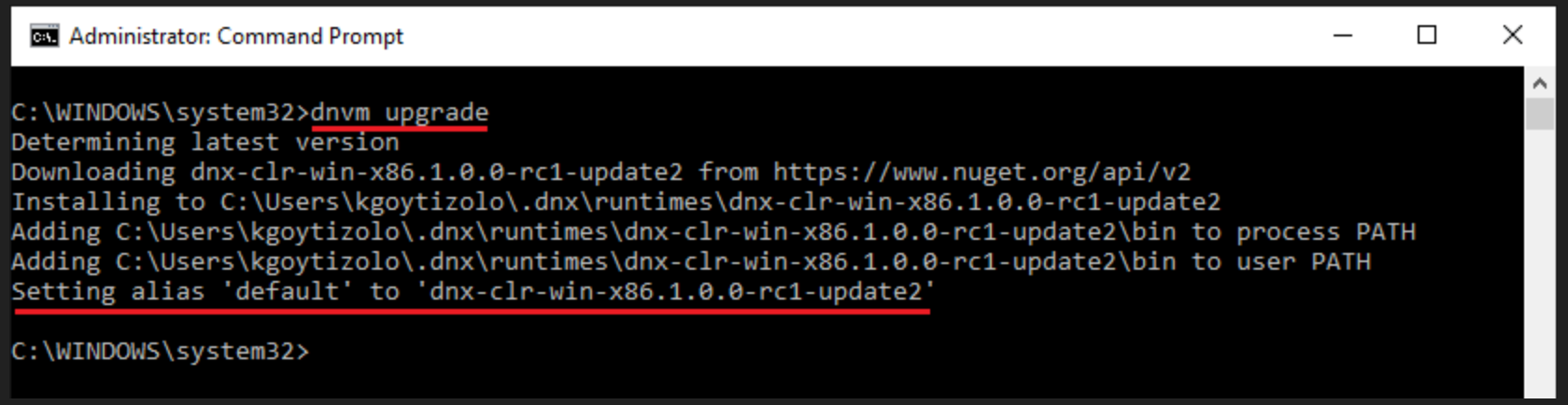


Microsoft

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

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





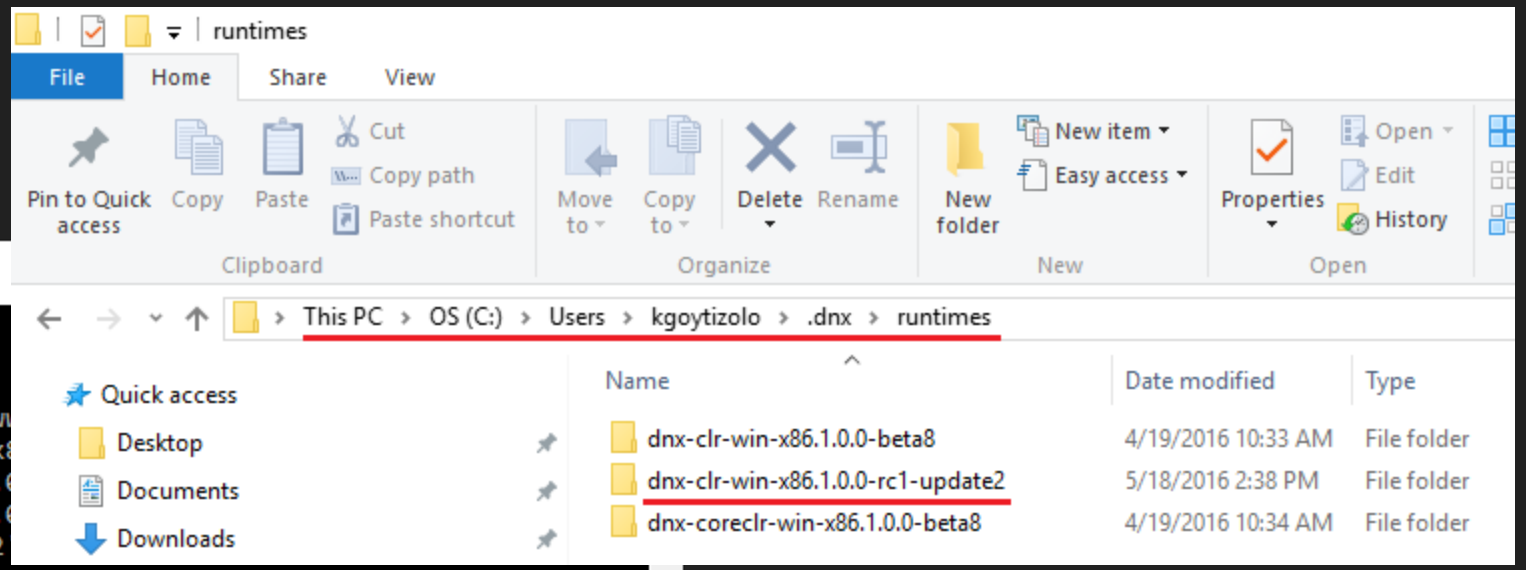
> Dnvm upgrade

Parámetros de ejemplo:

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

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

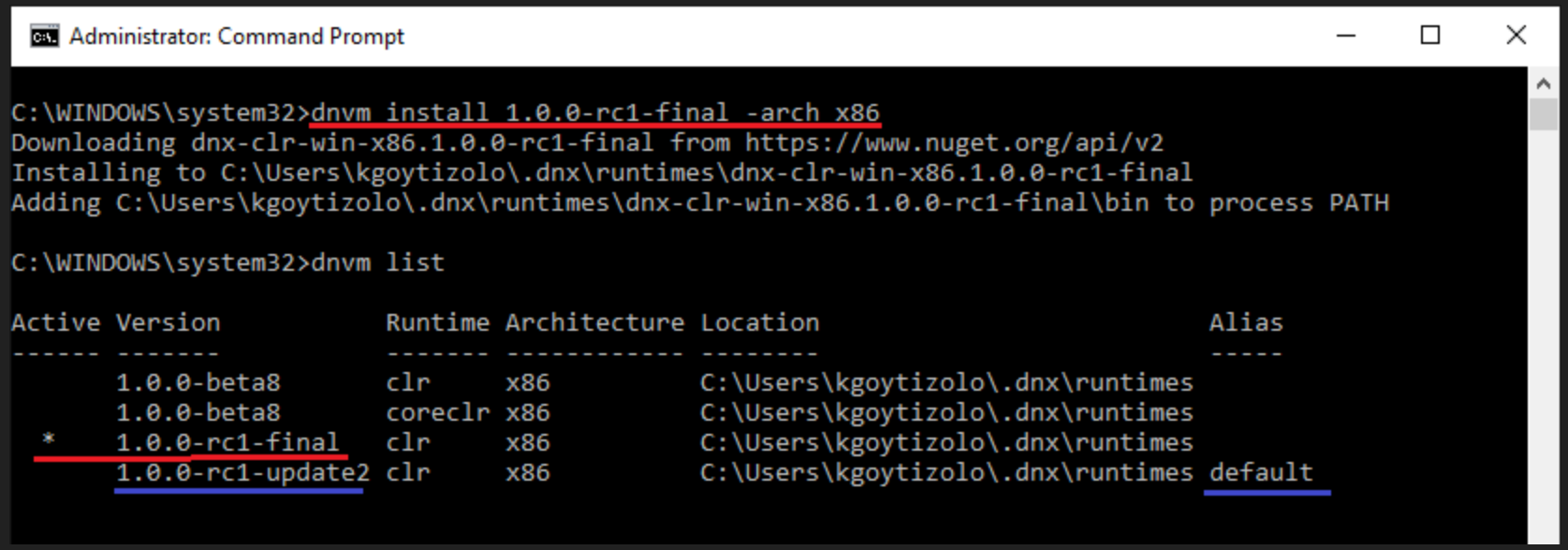
11. Dnvm Upgrade actualizó el último runtime y lo configuró por defecto:




```
Administrator: Command Prompt

C:\WINDOWS\system32>dnvm upgrade
Determining latest version
Downloading dnx-clr-win-x86.1.0.0-rc1-update2 from https://ww
Installing to C:\Users\kgoytizolo\.dnx\runtimes\dnx-clr-win-x
Adding C:\Users\kgoytizolo\.dnx\runtimes\dnx-clr-win-x86.1.0.
Adding C:\Users\kgoytizolo\.dnx\runtimes\dnx-clr-win-x86.1.0.
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
```

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

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



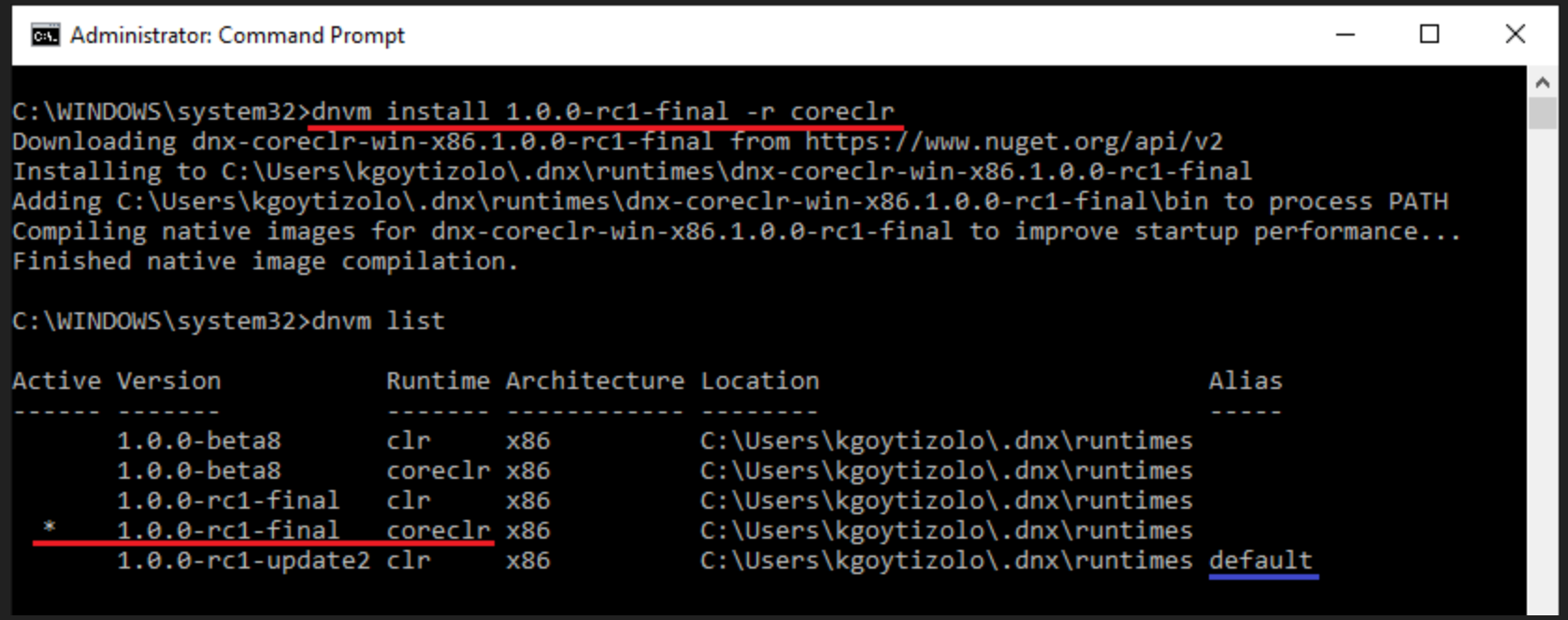
> Dnvm install

- 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]

Microsoft

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

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



> 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]

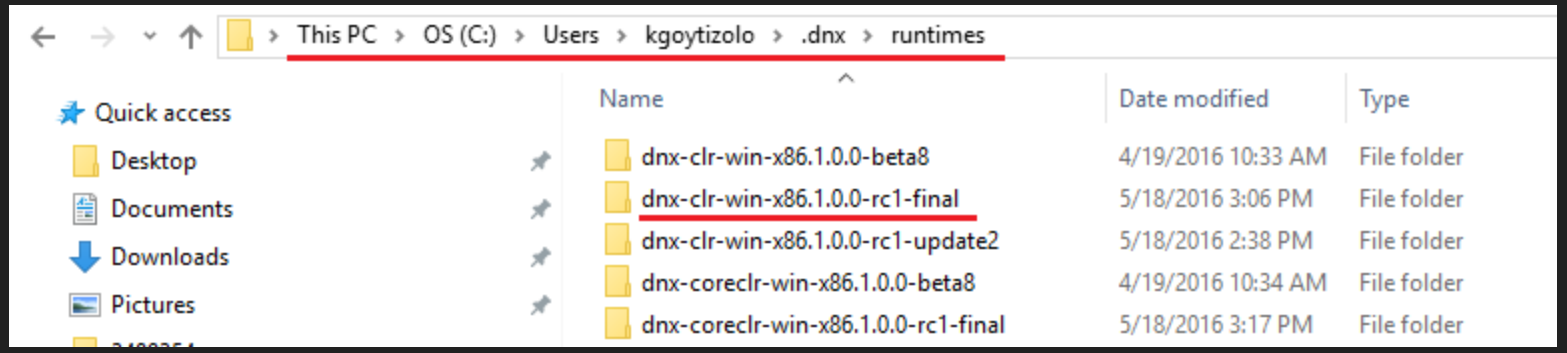
Microsoft

*** Este Runtime (CoreCRL) 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:

```
Administrator: Command Prompt

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
```



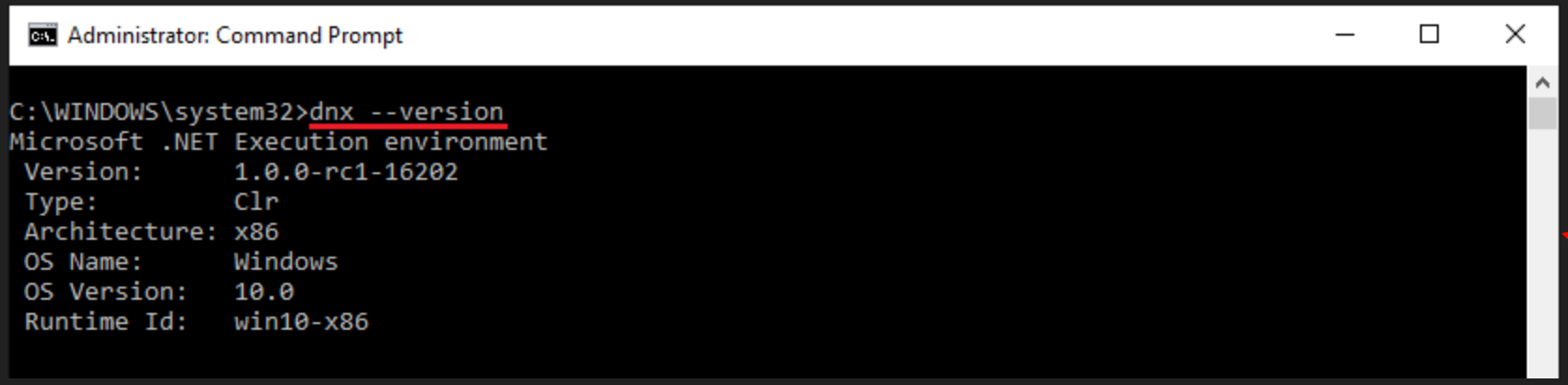
> Dnvm use

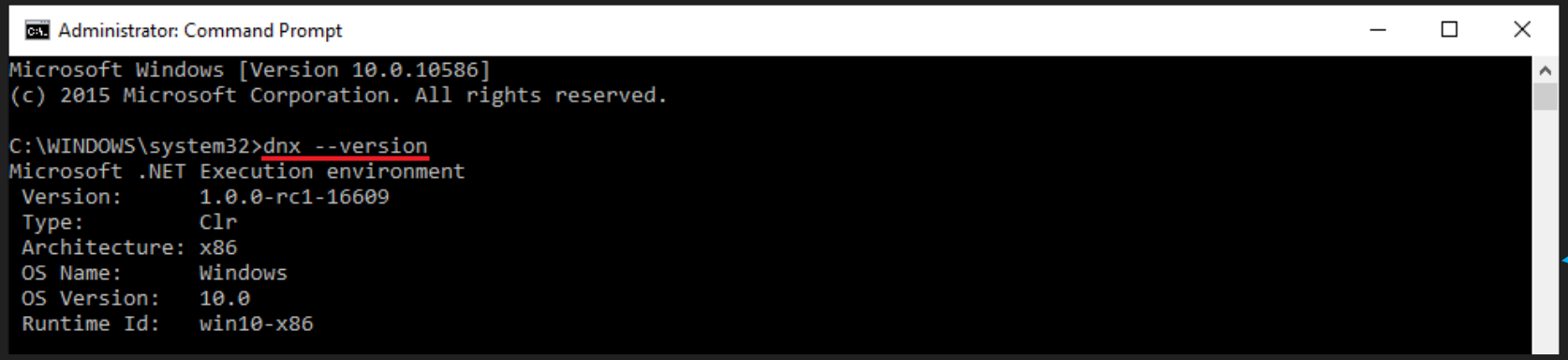
- Parámetros ejemplo:
  - Versión: 1.0.0-rc1-final
  - Runtime: -r clr
  - Arquitectura: -arch x86
- dnvm use <semver>|<alias>|none [-x86][-x64] [-svr50][-svrc50] [-p|-persistent] [-g|-global]

*** 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:





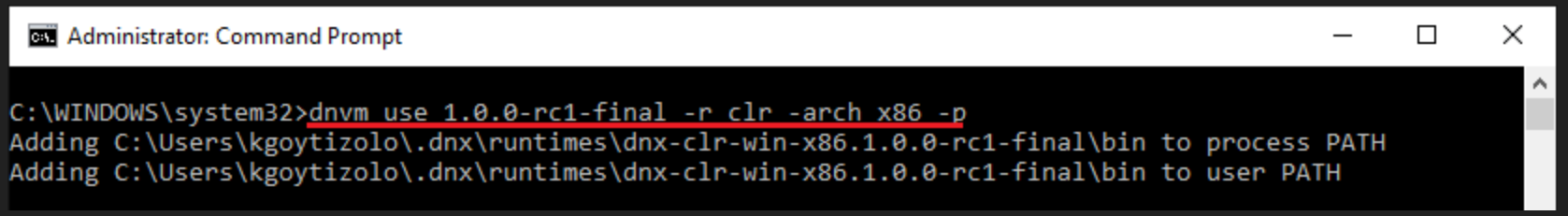
> Dnx --version

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

*** 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:



> Dnvm use

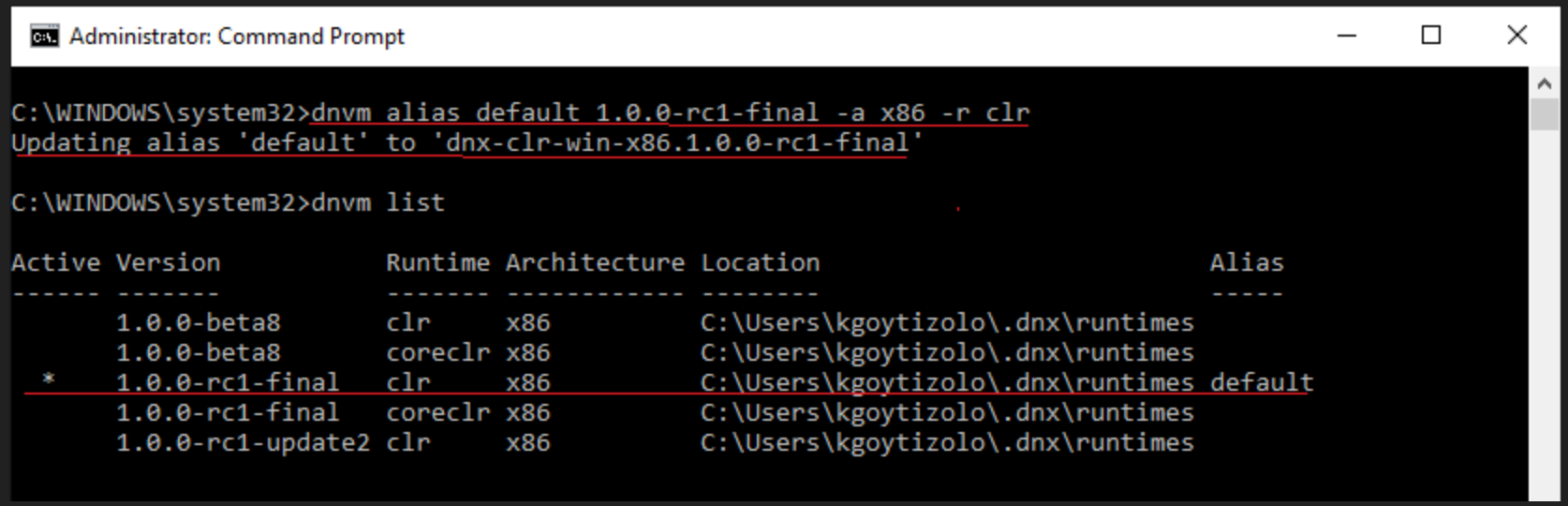
- Parámetros ejemplo:
  - Versión: 1.0.0-rc1-final
  - Runtime: -r clr
  - Arquitectura: -arch x86
  - Persistencia: -p
- dnvm use

<semver> | <alias> | none [-x86] [-x64] [-svr50] [-svrc50] [-p | -persistent] [-g | -global]

Microsoft

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

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



> Dnvm alias

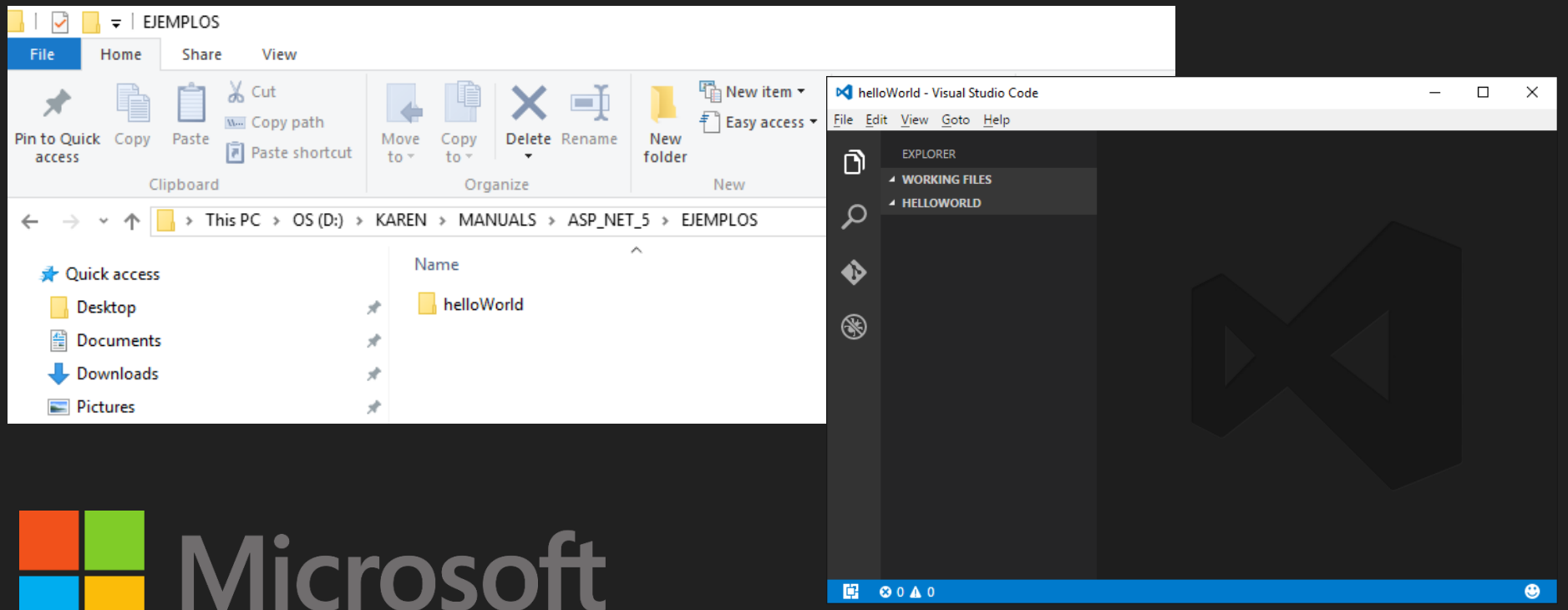
- Parámetros ejemplo:
  - Nombre: default
  - Versión: 1.0.0-rc1-final
  - Arquitectura: -a x86
  - Runtime: -r clr
- dnvm alias <alias> <semver> [-x86][-x64] [-svr50][-svrc50]

Microsoft

*** 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:



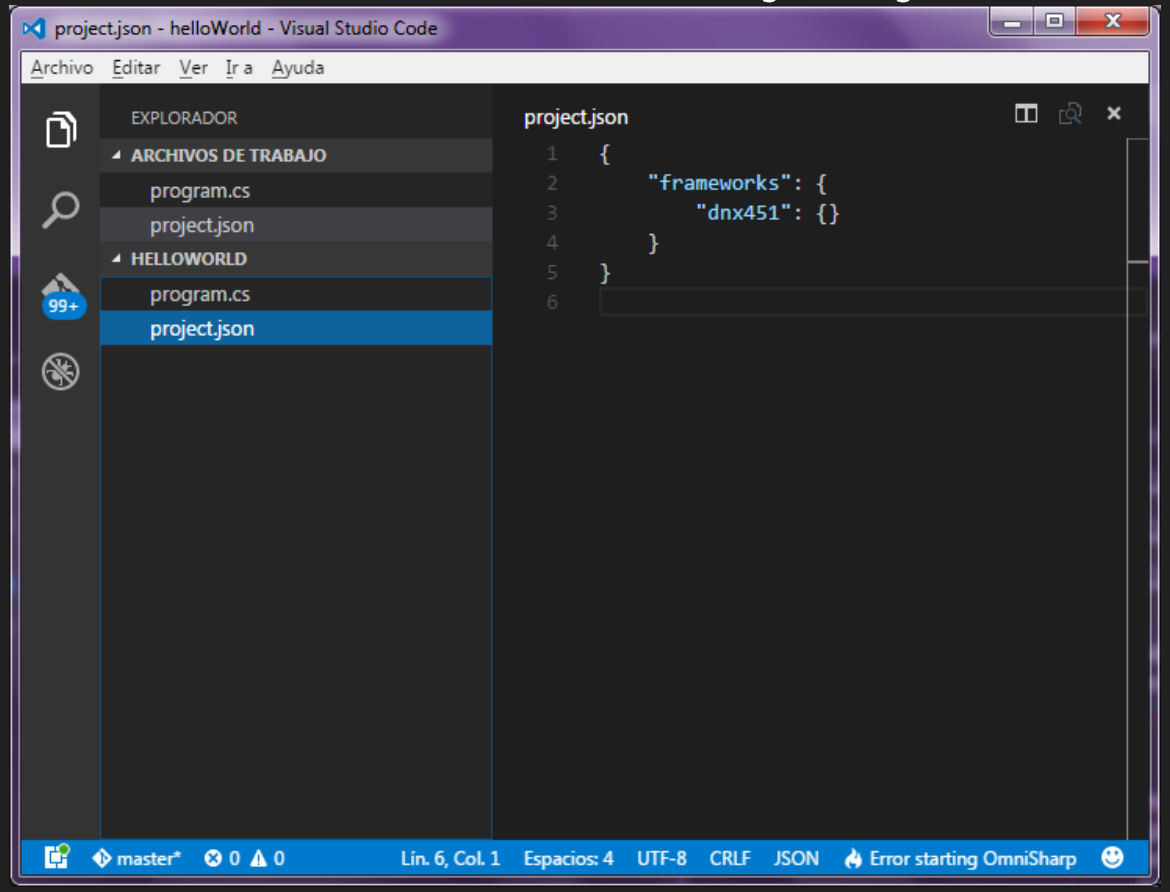
También se puede abrir desde el mismo Visual Studio Code en:

File > Open Folder

Microsoft

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

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



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

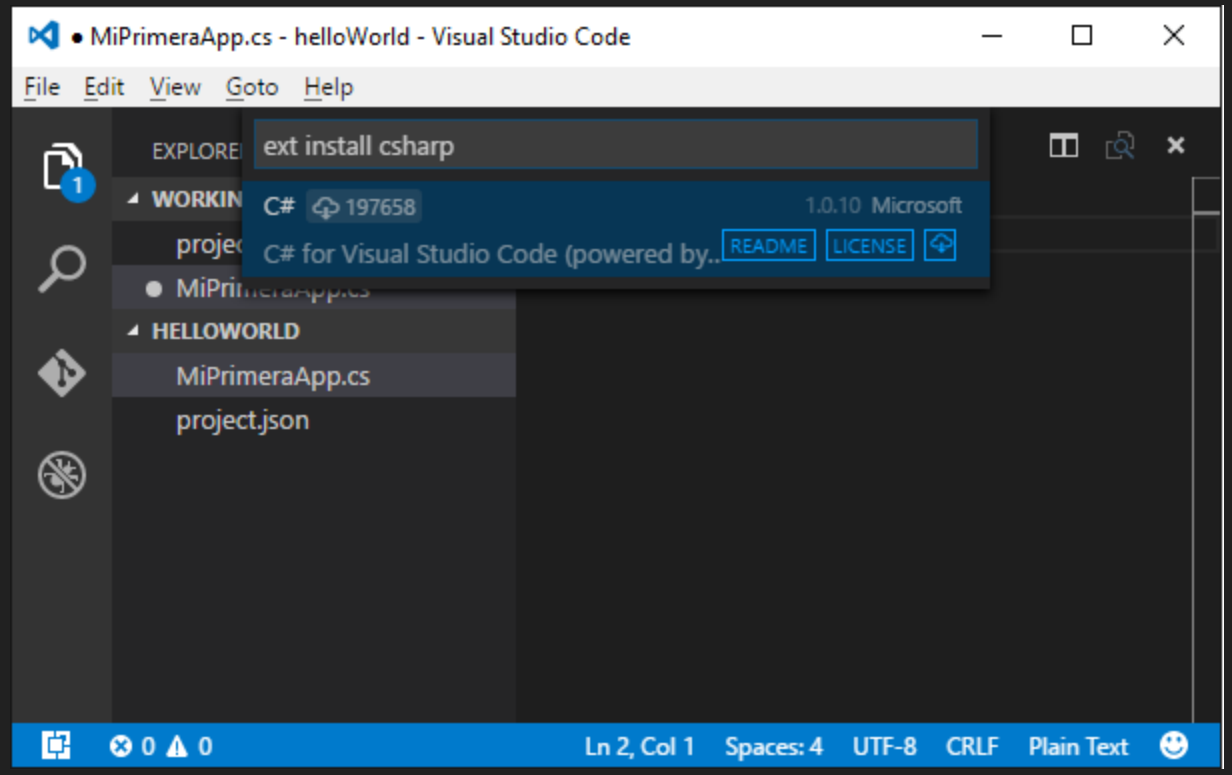
*** Añadir framework dnx451 dentro del proyecto**

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#):



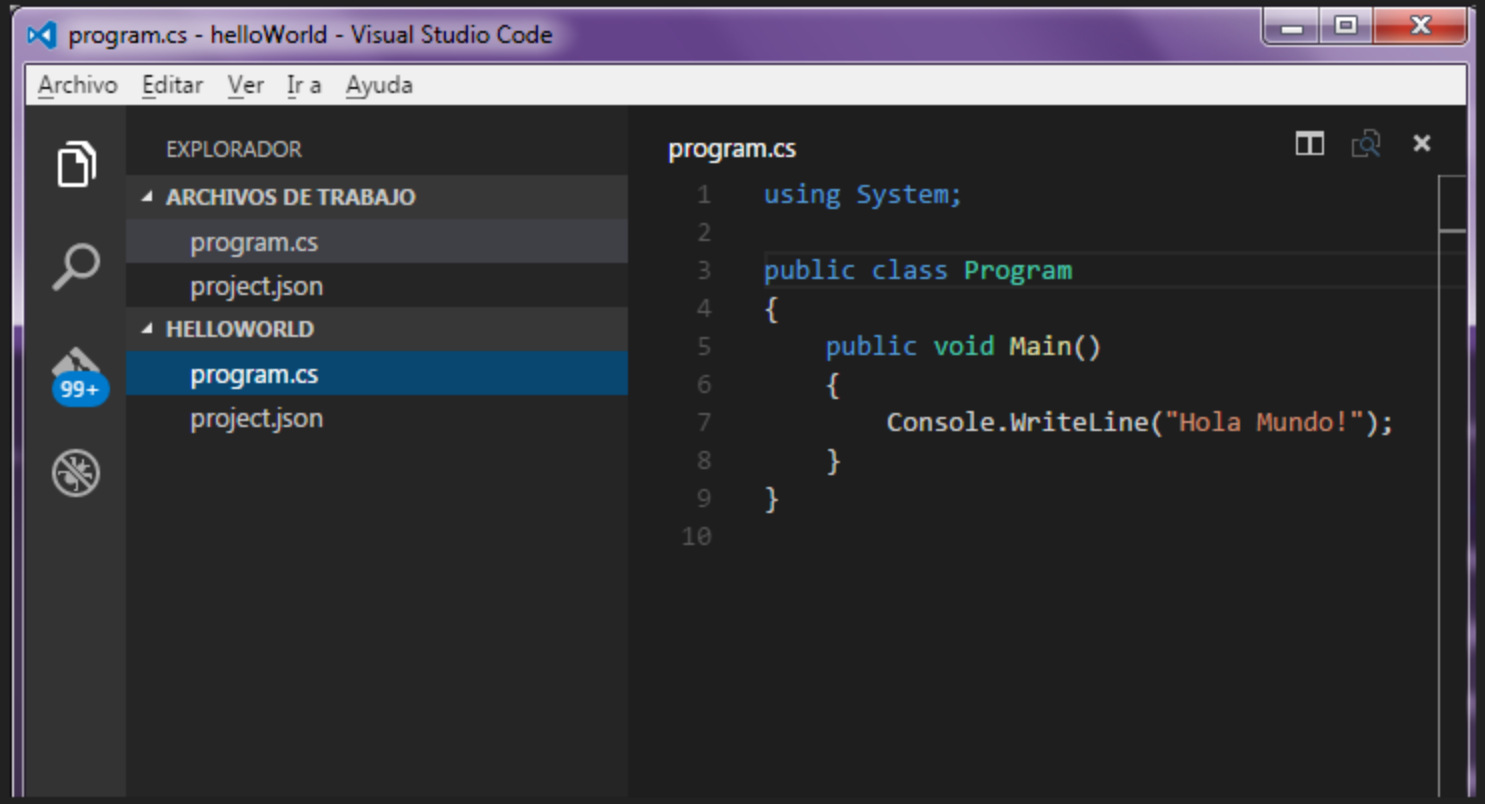
CTRL + P >

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

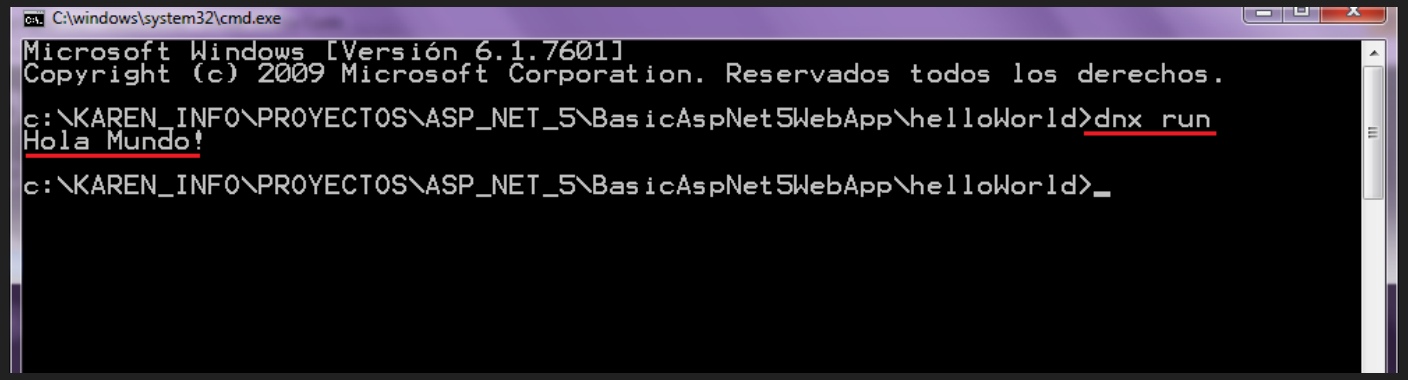
4. Crear un archivo “program.cs” asociado al proyecto “helloWorld”



- 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:



> Dnx run

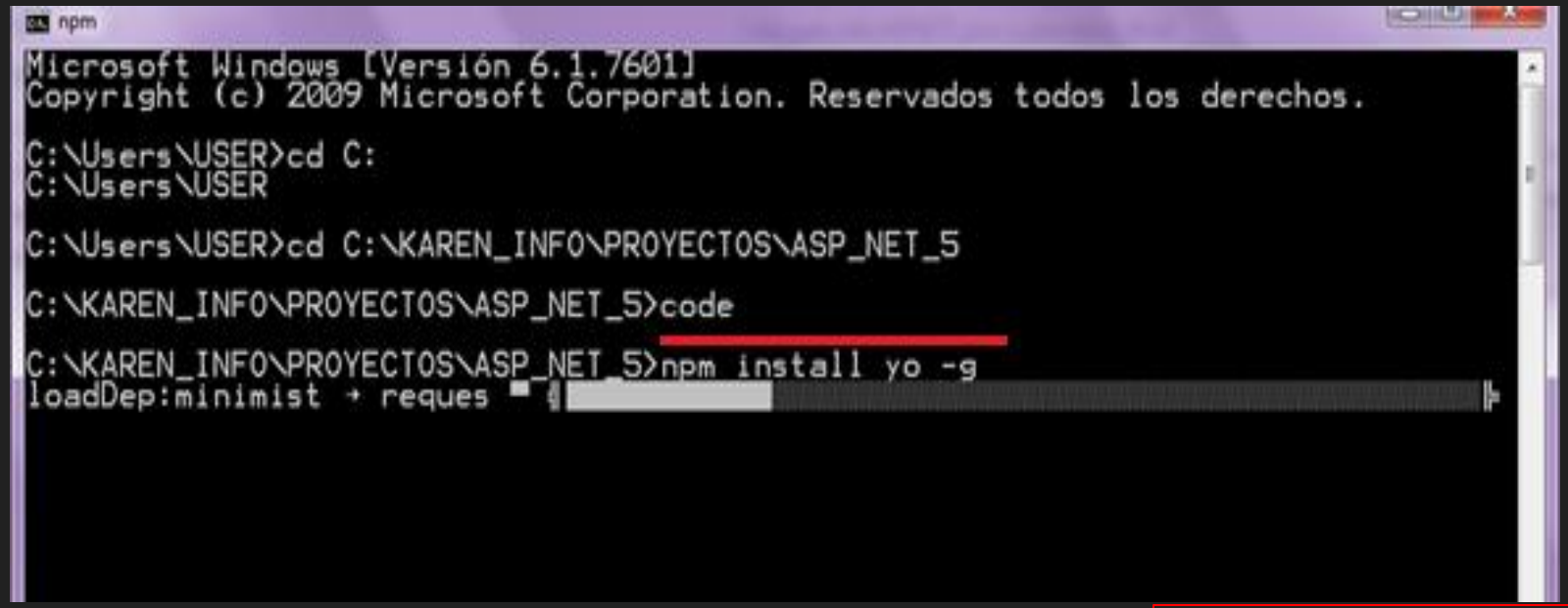
- DNX es el entorno de ejecución de .NET para aplicaciones ASP.NET 5 en adelante.
- A partir de este framework, los proyectos son basados en DNX.
- DNX permite construir y ejecutar aplicaciones multiplataforma.
- **Dnx run** ejecuta la aplicación, hosting, debug, etc.

Microsoft

!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 install

yo -g

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

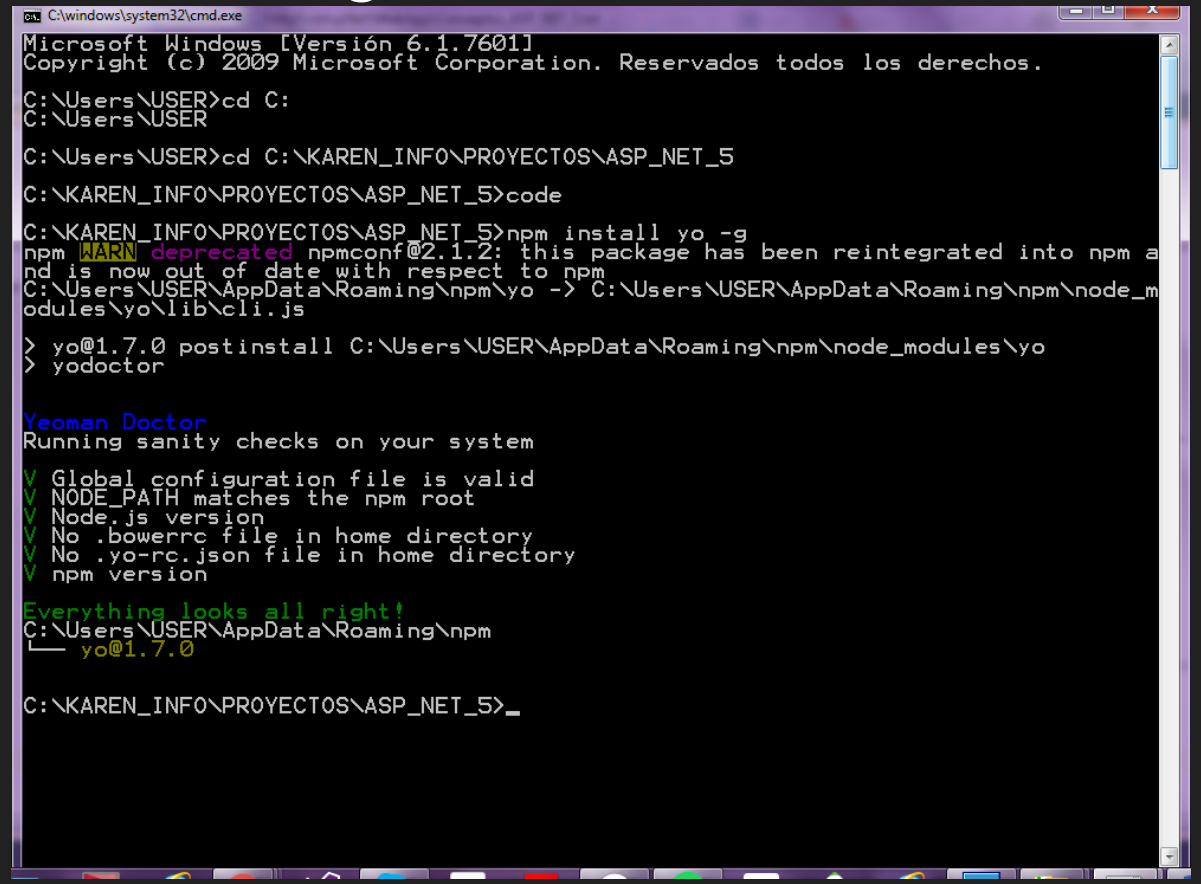
Microsoft

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:



> Npm install

yo -g

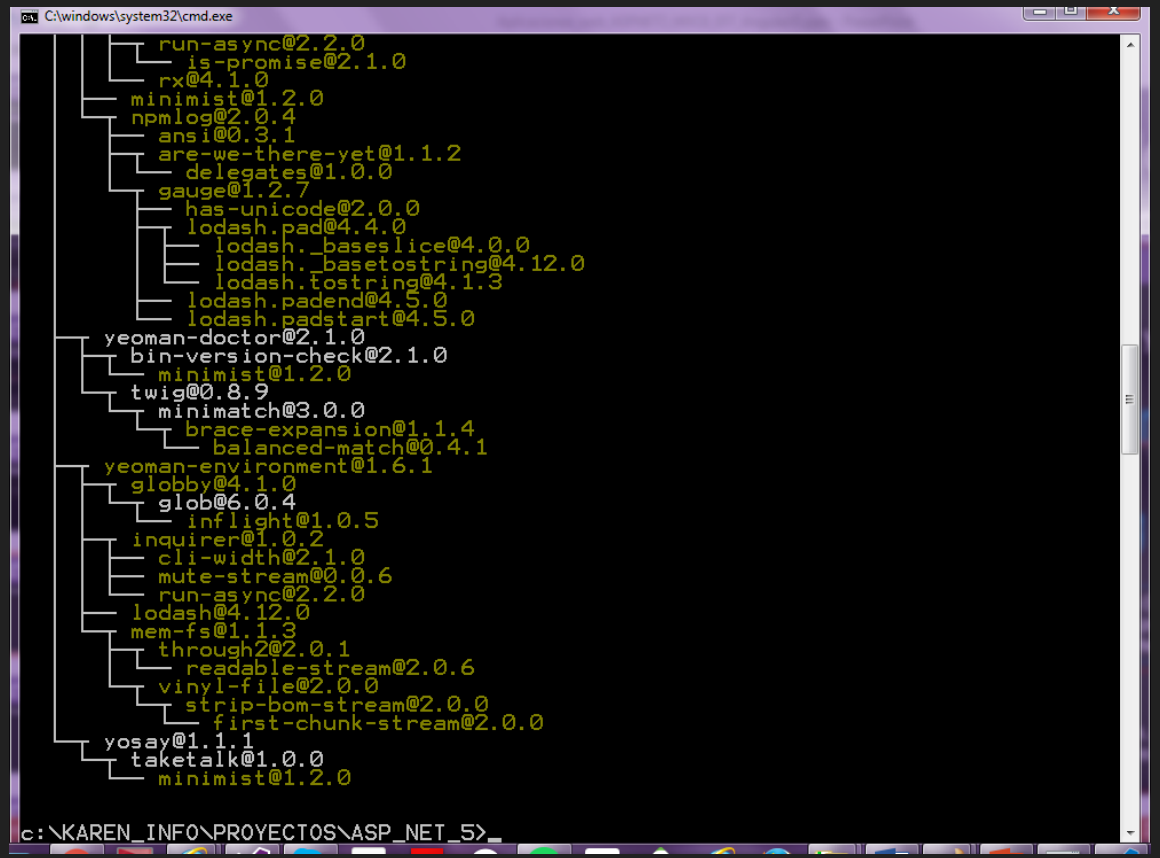
Parámetro –g (Indica que se instale globalmente para que esté disponible en cualquier parte de la máquina).

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:



> Npm install

yo -g

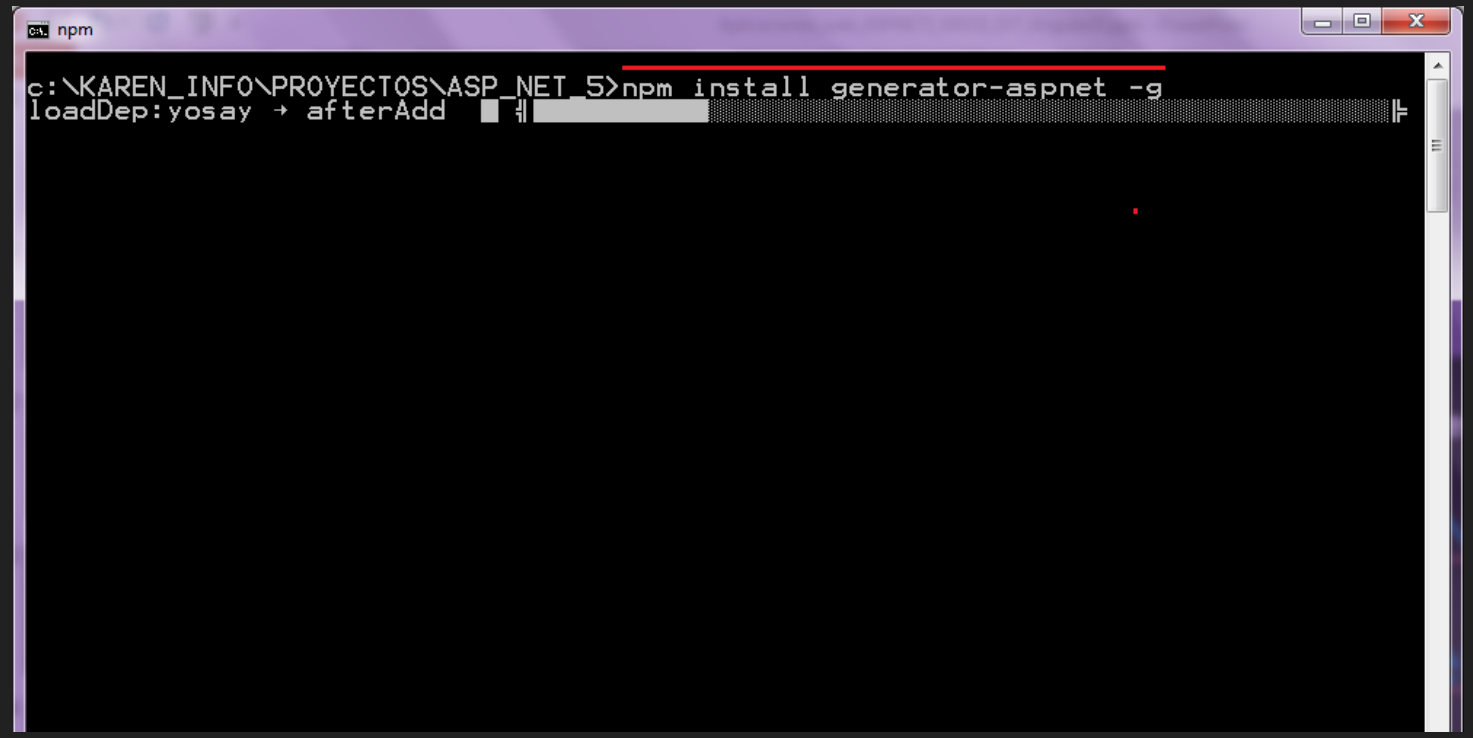
Parámetro –g (Indica que se instale globalmente para que esté disponible en cualquier parte de la máquina).

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:

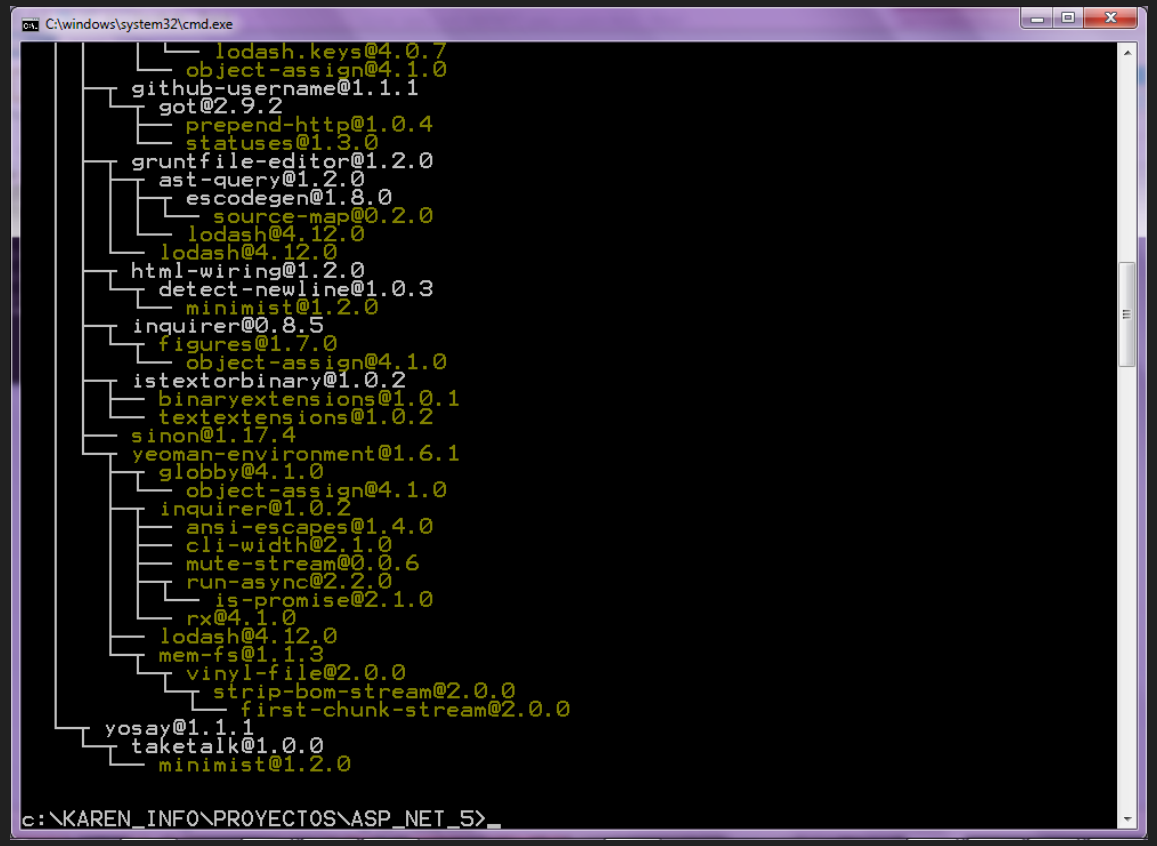


> Npm install generator-aspnet -g

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:

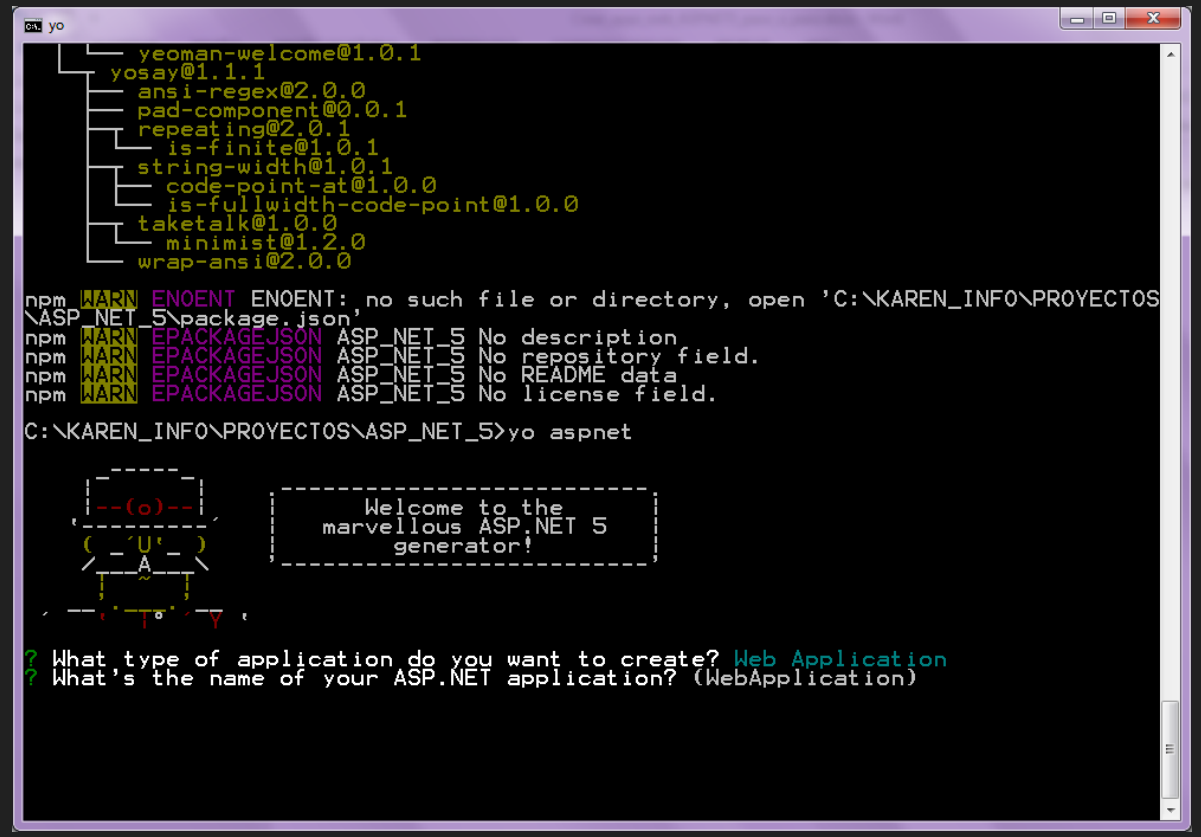


> Npm install generator-aspnet -g

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

Usando aplicaciones MVC plantilla con DNX y Visual Studio Code

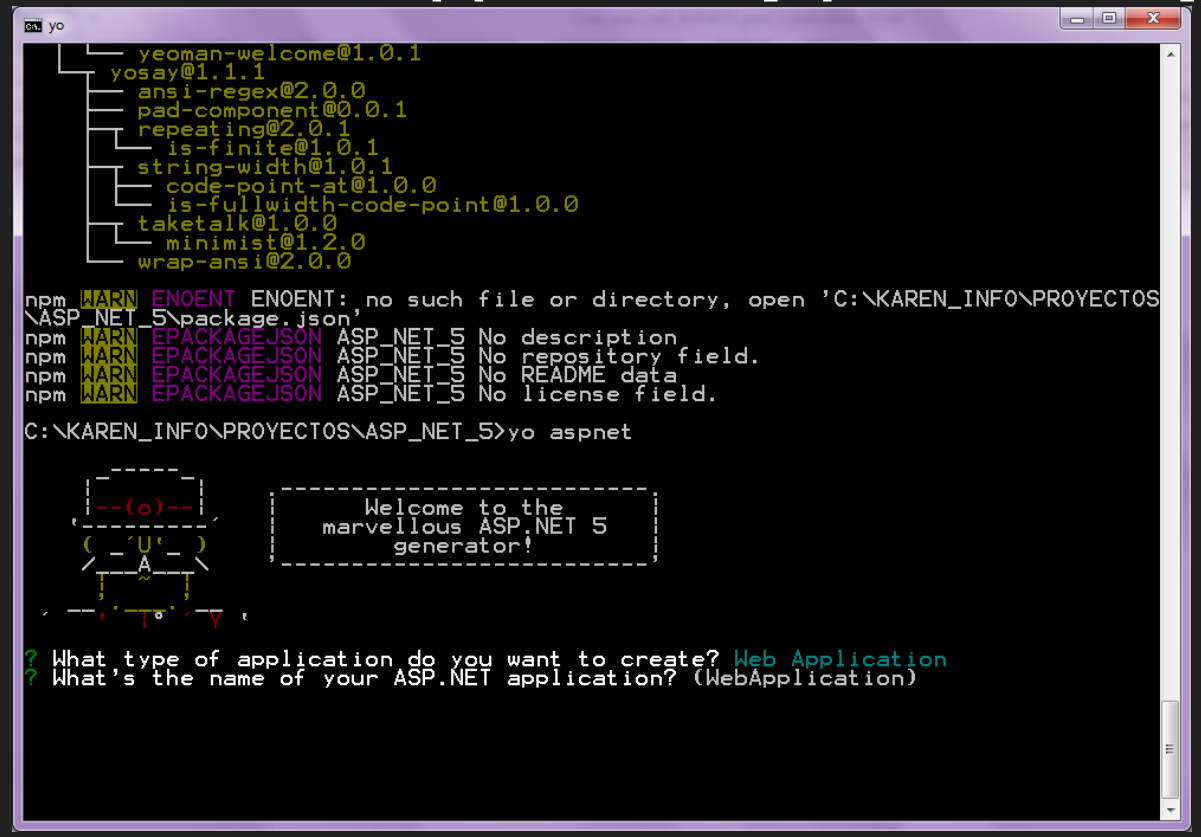
3. Ejecutamos la herramienta Yeoman, asociada al generador ASP.NET:



> Yo aspnet

Usando aplicaciones MVC plantilla con DNX y Visual Studio Code

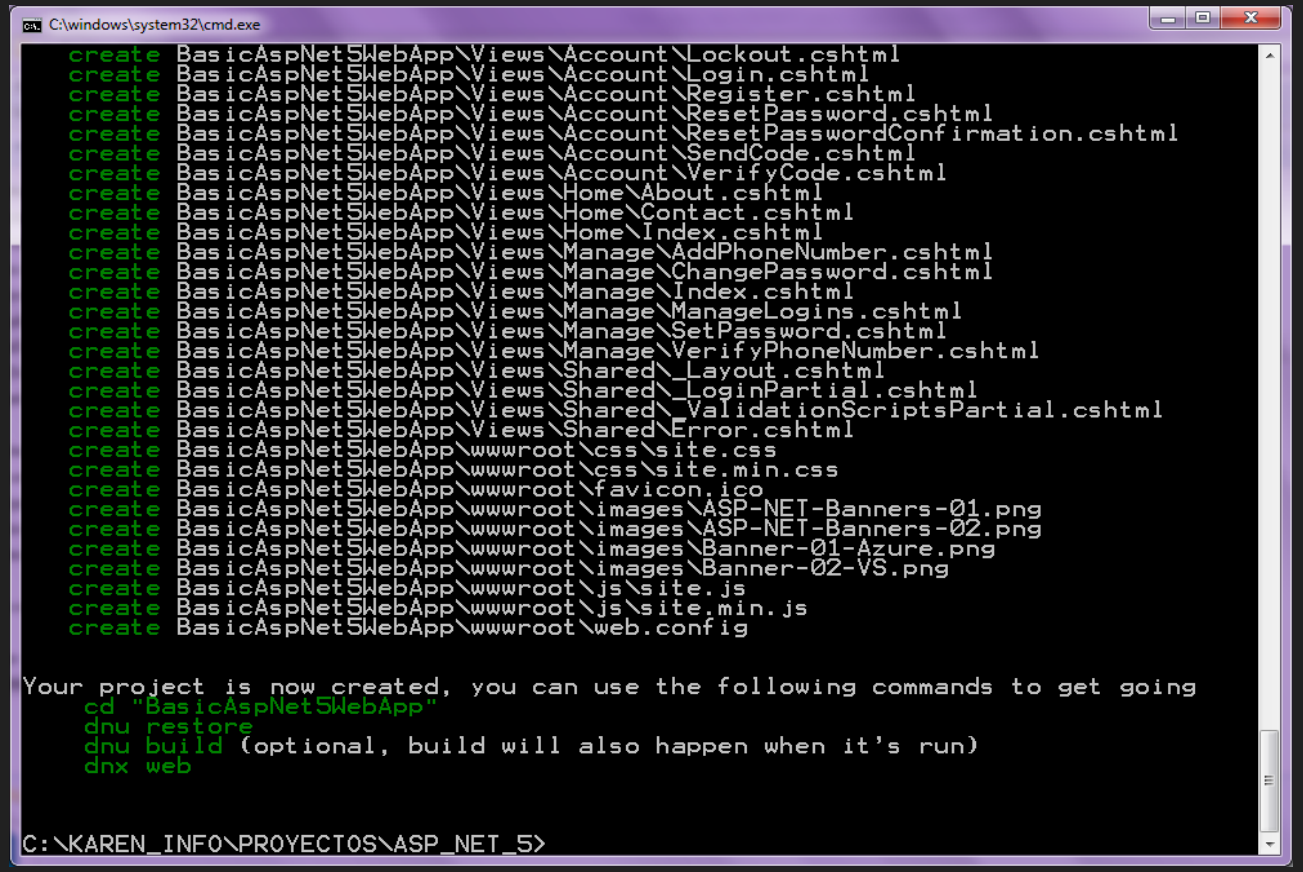
4. Seleccionamos el tipo de aplicación ASP.NET 5 del generador (En este caso, Web Application [opción Nro 3]) y el Nombre de tu aplicación web:



> Yo aspnet

Usando aplicaciones MVC plantilla con DNX y Visual Studio Code

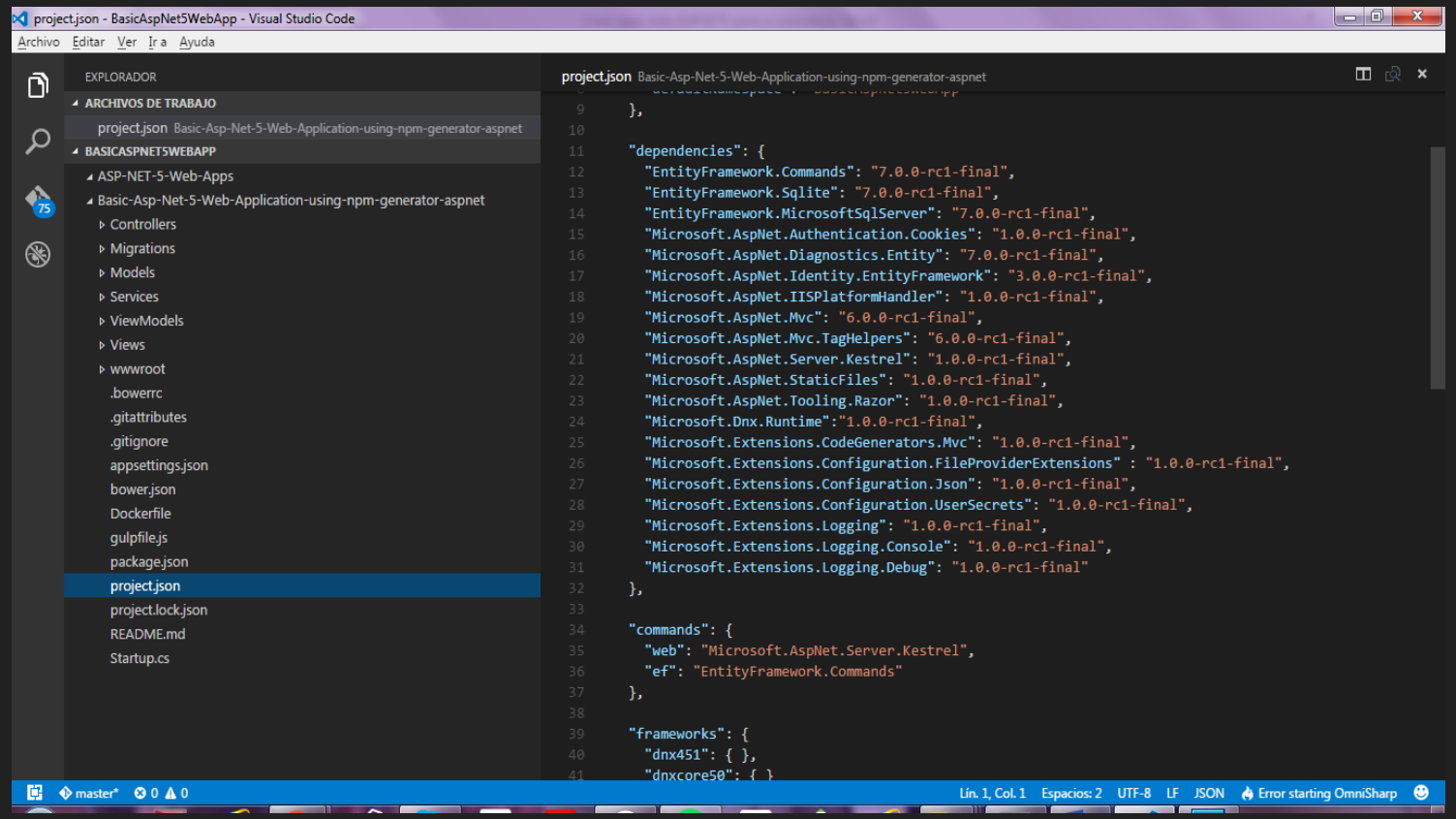
5. Y listo! La aplicación MVC 6 web plantilla fue creada satisfactoriamente:



> Yo aspnet

Usando aplicaciones MVC plantilla con DNX y Visual Studio Code

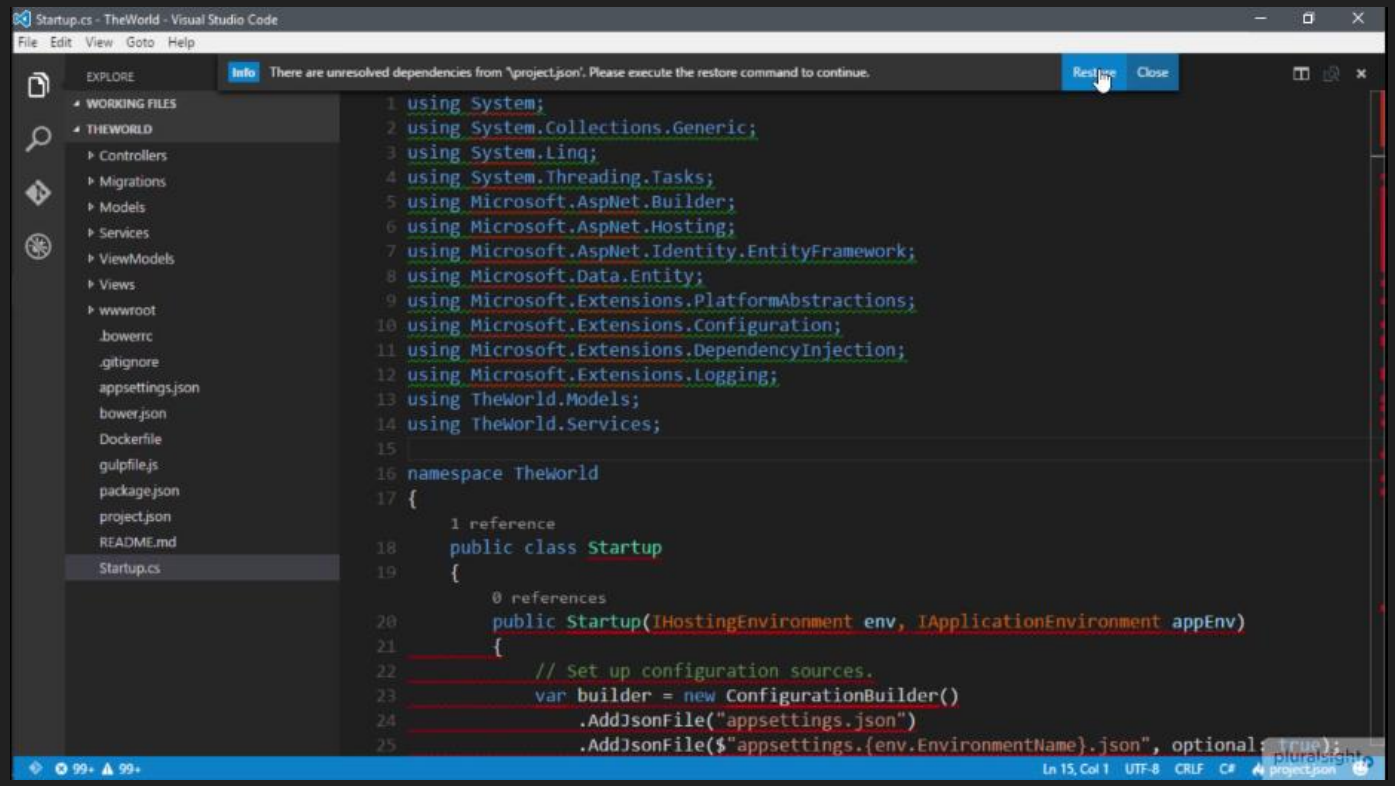
6. Abrir la aplicación generada por Yeoman con Visual Studio Code:



- 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":



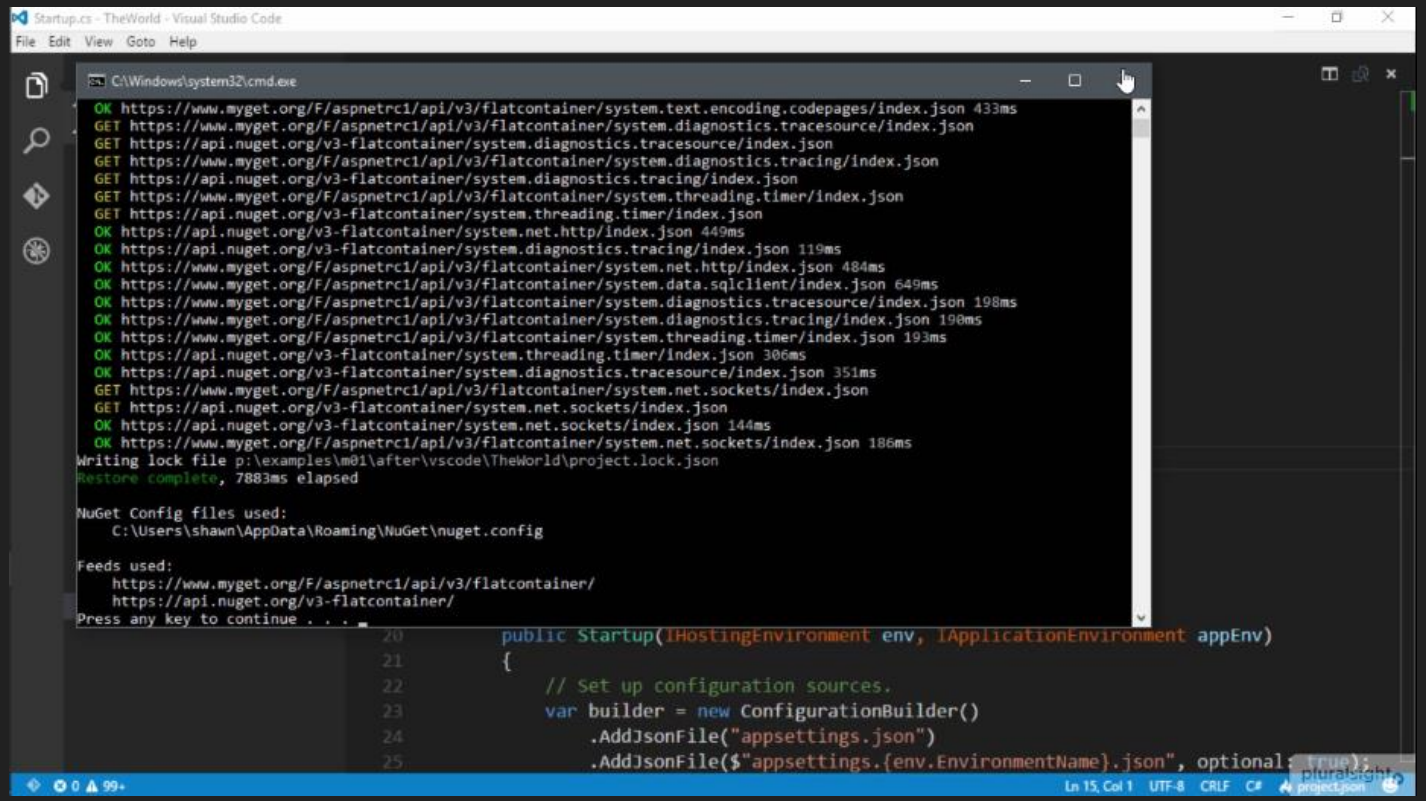
Internamente, el botón "Restore" ejecuta la herramienta runtime "dnu restore"

> 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":



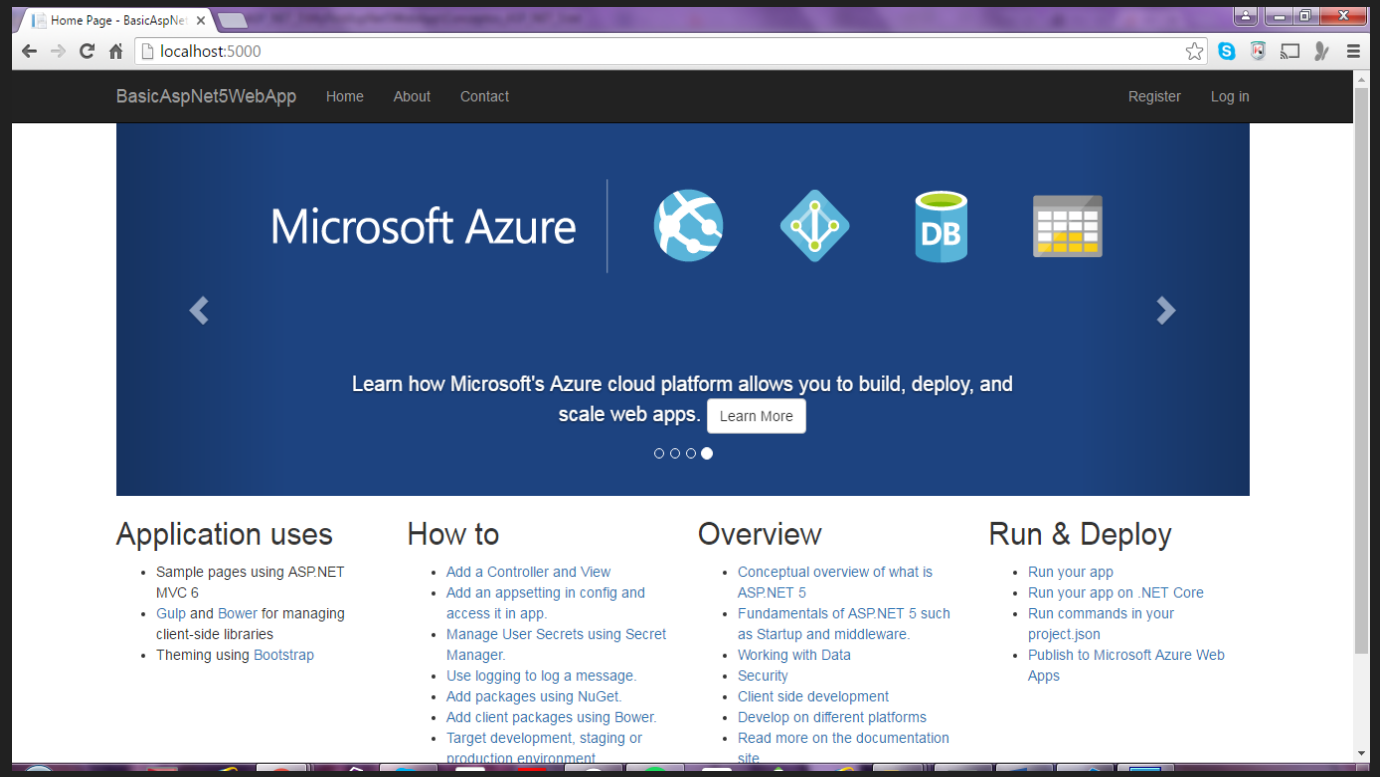
Internamente, el botón "Restore" ejecuta la herramienta runtime "dnu restore"

> 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

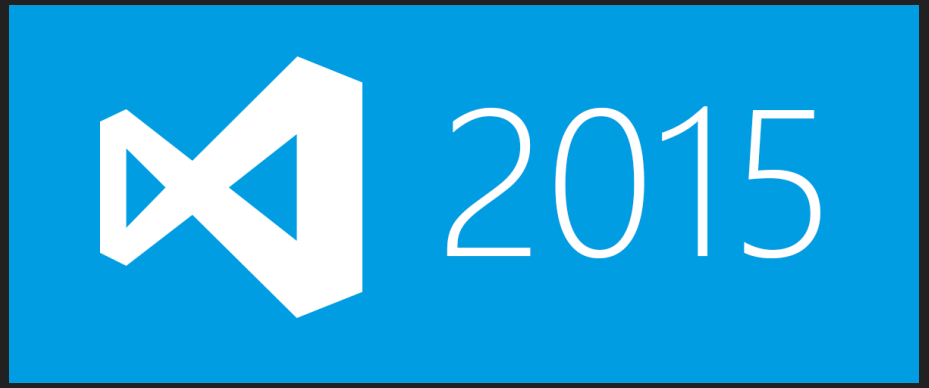
Con esta instrucción, levantamos el servidor de aplicaciones y el puerto correspondiente:

http://localhost:5000

(Este es el que trabaja Dnx web por defecto)

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

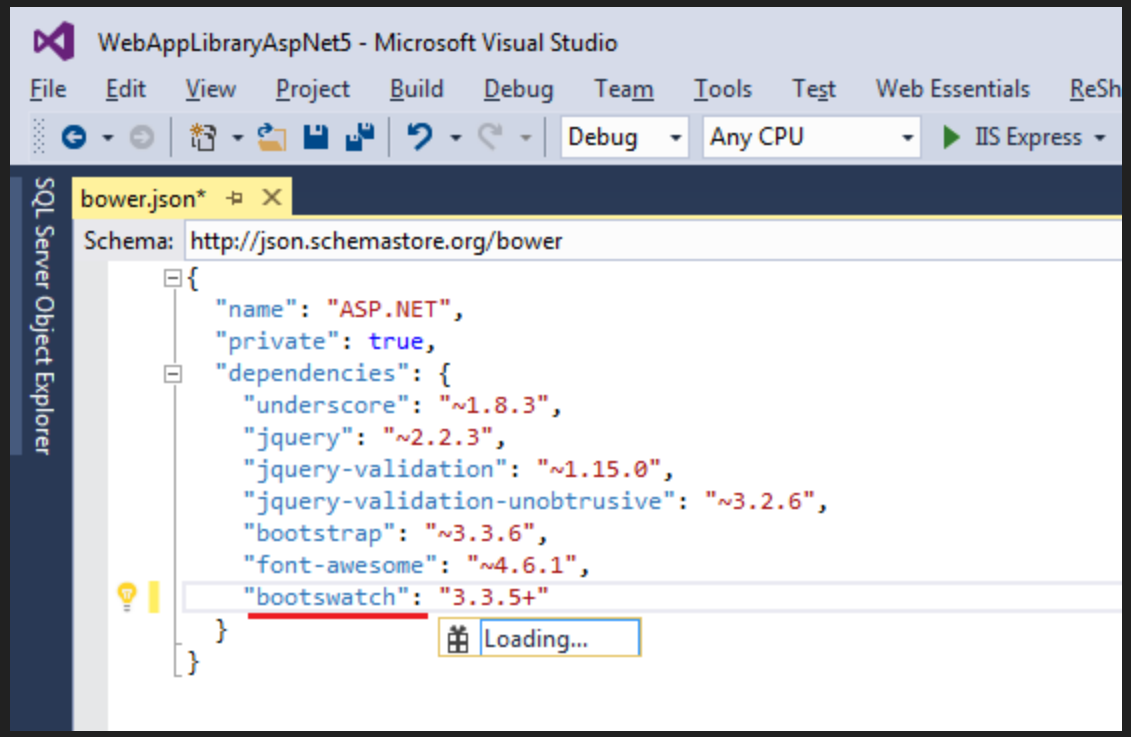
Y a partir de acá… todo es Visual Studio 2015

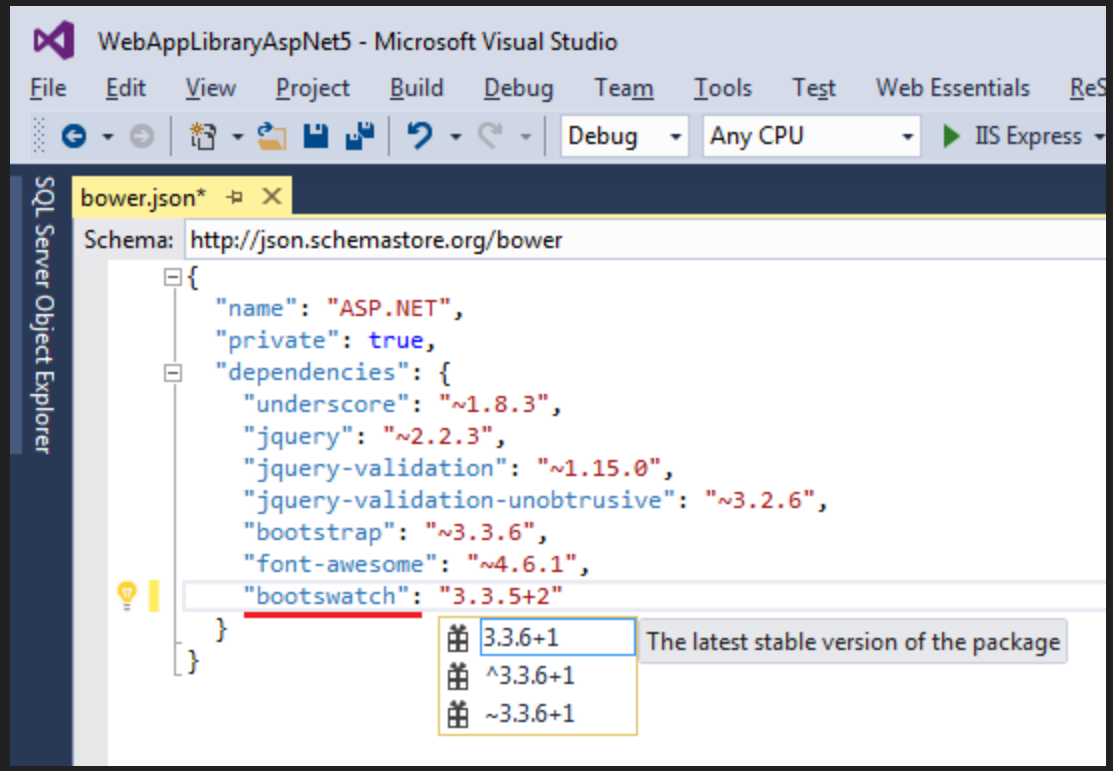


Visual Studio 2015

Microsoft

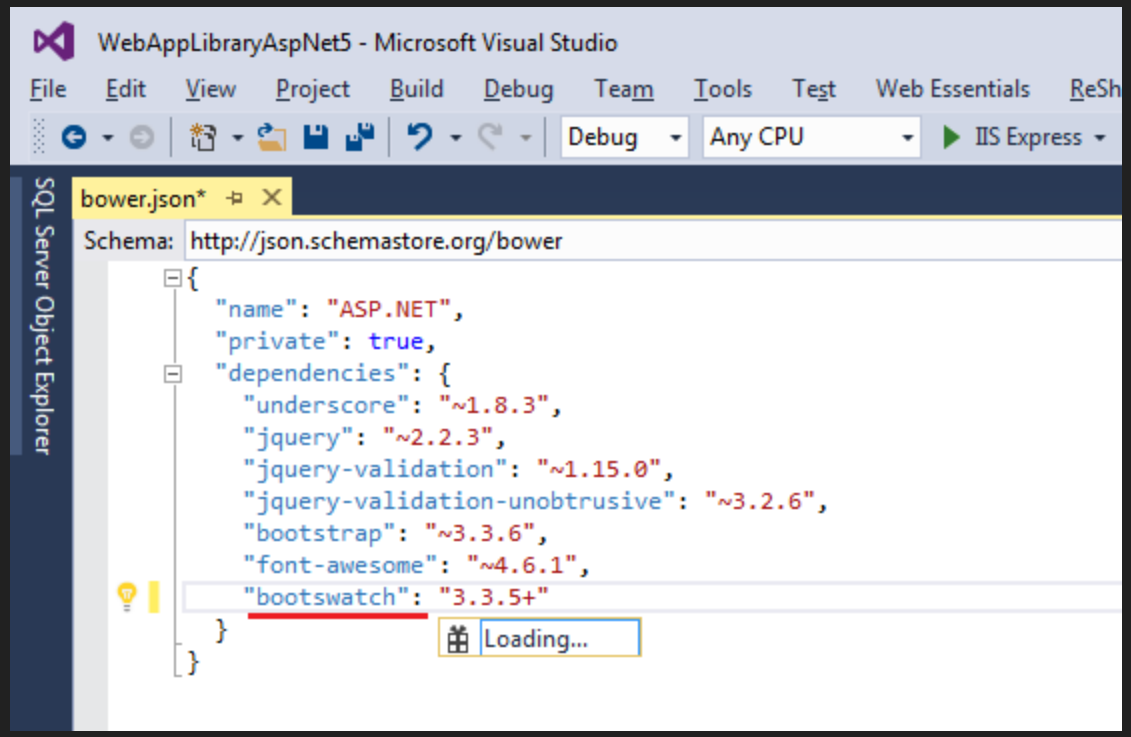
Intellisense + Bower: Adicionando dependencias por el lado del cliente

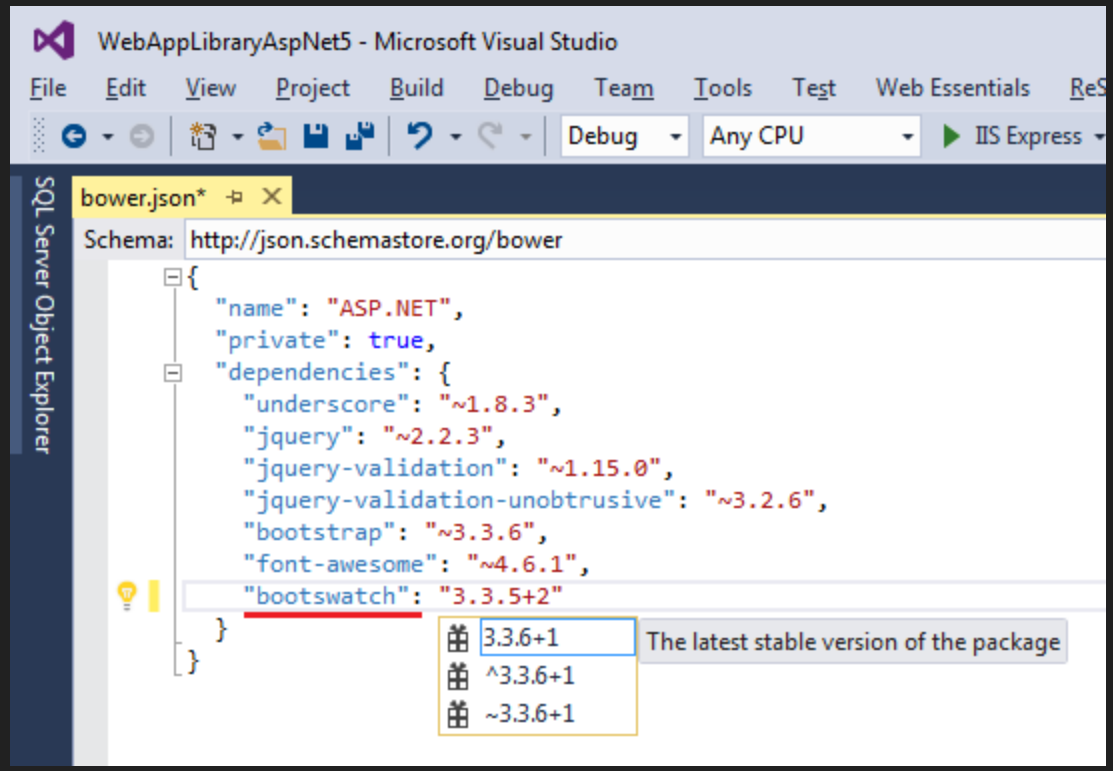




Microsoft

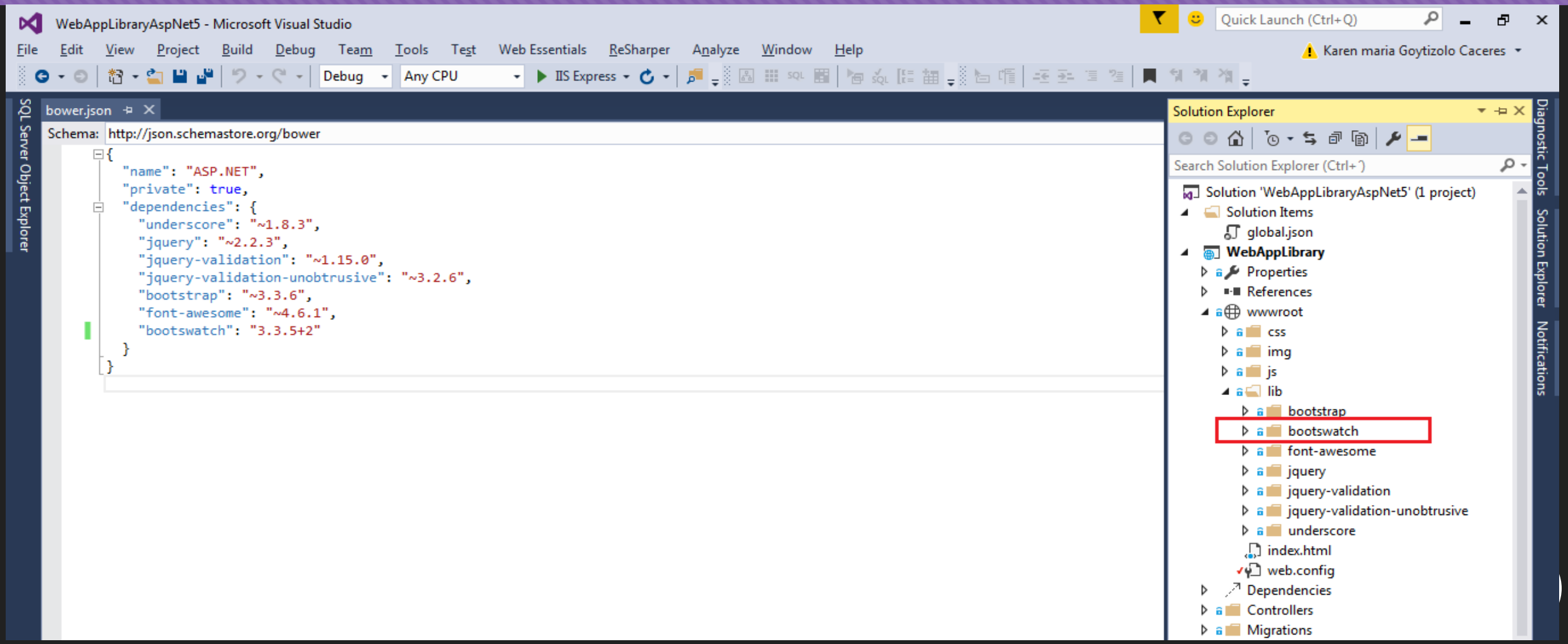
Intellisense + Bower: Adicionando dependencias por el lado del cliente



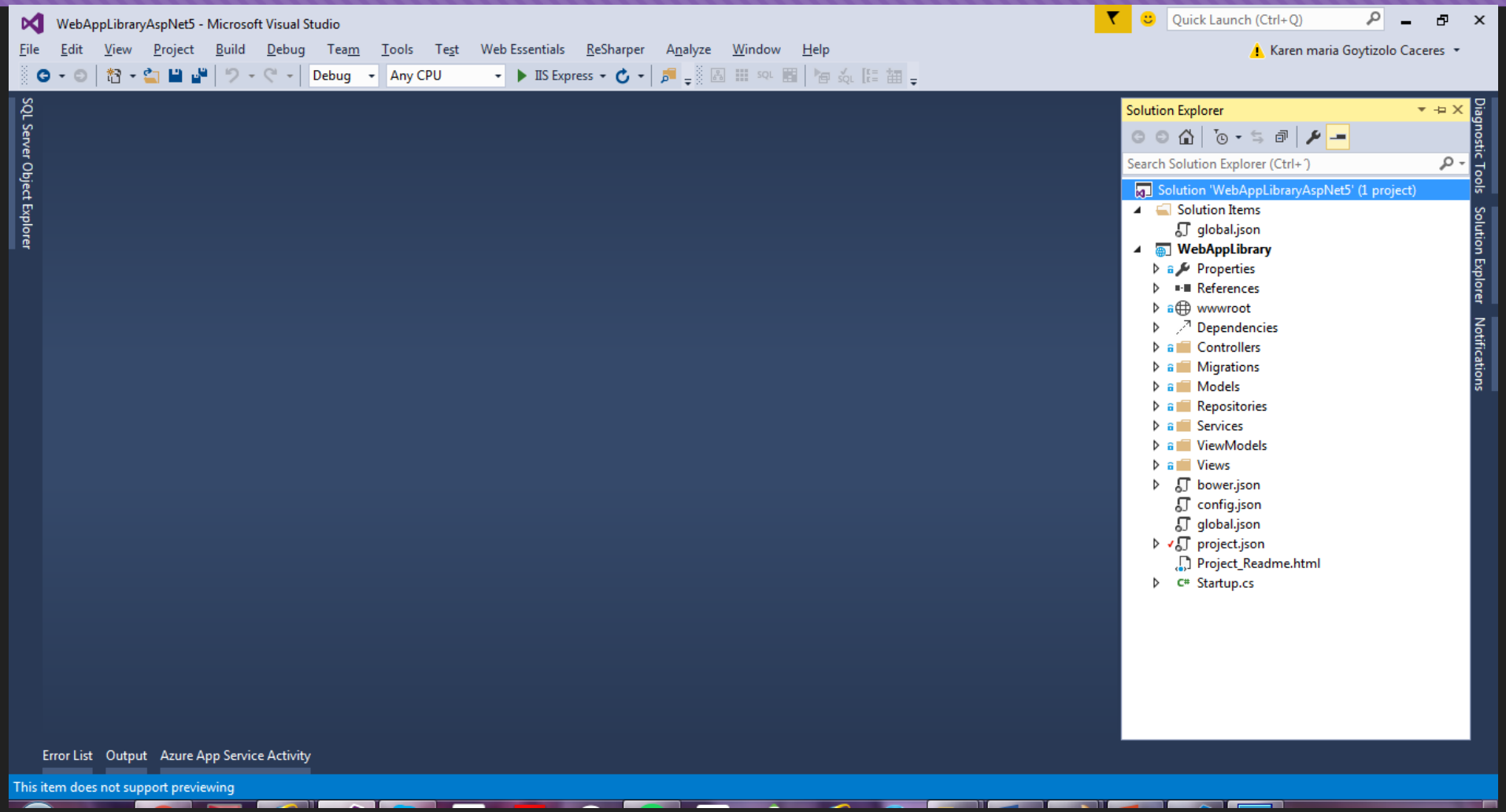


Microsoft

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
  - Wwwroot
  - 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

Microsoft

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

Microsoft

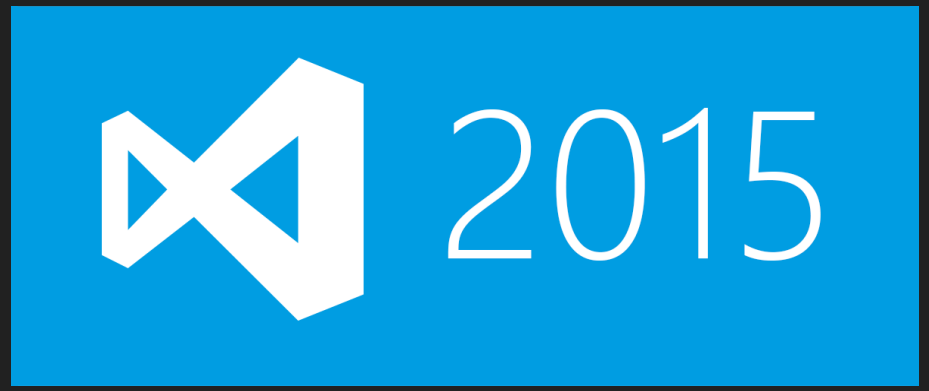
Aplicando Entity Framework 7 para armar tu base de datos

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

Microsoft

Uso de Angular JS para potenciar el lado del cliente



DEMO - Visual Studio 2015

Microsoft

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

Microsoft

- 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

Microsoft

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 (sobretodo TDD).
- Tu equipo está familiarizado con metodologías Scrum para los entregables de software.

Microsoft

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/

Microsoft

GRACIAS!!

Microsoft