



## Ingeniería de software

### Laboratorio 1

#### Pre requisitos

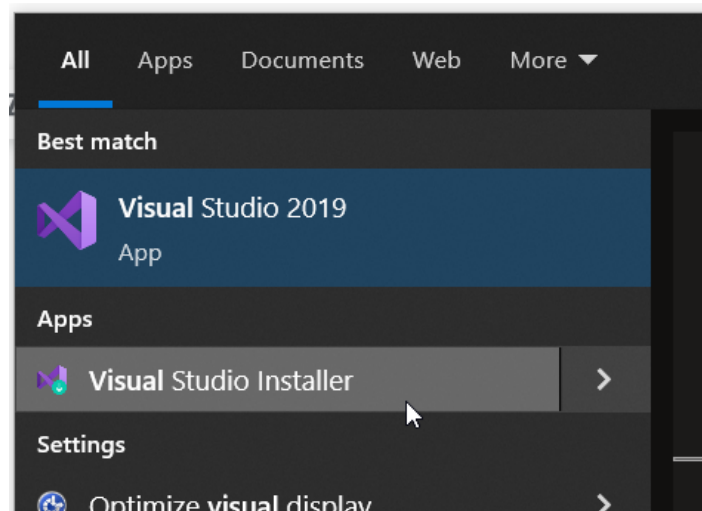
1. Visual studio 2019 instalado
2. Dependencias de ASP.NET instaladas en Visual Studio
3. Tener una cuenta en GitHub

#### Objetivo

El objetivo del laboratorio es poder crear un repositorio en GitHub y conectarlo a Visual Studio para generar un proyecto ASP.NET que pueda utilizar este repositorio como control de versiones. Como parte del laboratorio se abordarán conceptos importantes del control de versiones.

#### Instrucciones

1. Verificar que se tiene instalado Visual Studio con las dependencias de ASP.NET:
  - a. Abra Visual studio installer



- b. Instale o use la opción de modificar y asegúrese que *Desarrollo de ASP.NET and web development* y *Azure development* fue seleccionado e instalado.



Installed Available

**Visual Studio Community 2019**  
16.11.9  
 Update available  
16.11.11 [View details](#)

[Modify](#)  
[Launch](#)  
[More ▾](#)  
 Update

Visual Studio Installer

Modifying — Visual Studio Community 2019 — 16.11.9

Workloads Individual components Language packs Installation locations

Web & Cloud (4)

**ASP.NET and web development**  
Build web applications using ASP.NET Core, ASP.NET, HTML/JavaScript, and Containers including Docker support... ☒

**Azure development**  
Azure SDKs, tools, and projects for developing cloud apps and creating resources using .NET and .NET Framework... ☒

**Python development**  
Editing, debugging, interactive development and source control for Python. ☐

**Node.js development**  
Build scalable network applications using Node.js, an asynchronous event-driven JavaScript runtime. ☐

Desktop & Mobile (5)

**.NET desktop development**  
Build WPF, Windows Forms, and console applications using C#, Visual Basic, and F# with .NET and .NET Framework... ☐

**Desktop development with C++**  
Build modern C++ apps for Windows using tools of your choice, including MSVC, Clang, CMake, or MSBuild. ☐

**Universal Windows Platform development**  
Create applications for the Universal Windows Platform ☐

**Mobile development with .NET**  
Build cross-platform applications for iOS, Android or ☐

**Installation details**

- Included
  - ☒ Azure development prerequisites
  - ☒ .NET development tools
  - ☒ .NET Framework 4.7.2 development tools
  - ☒ ASP.NET and web development tools
  - ☒ Container development tools
- Optional
  - ☒ .NET Framework 4 – 4.6 development tools
  - ☒ Azure Data Lake and Stream Analytics Tools
  - ☒ Visual Studio Tools for Kubernetes
  - ☒ Azure Resource Manager tools
  - ☒ Azure Cloud Services tools
  - ☒ Service Fabric Tools
  - ☒ Advanced ASP.NET features
  - ☒ Web Deploy
  - ☒ Azure PowerShell
  - ☒ IntelliCode
  - ☐ Windows Communication Foundation
  - ☐ Azure Storage AzCopy
  - ☐ .NET Framework 4.6.1 development tools

2. Crear un cuenta en GitHub: Github es un repositorio de código el cual puede alojar código fuente de proyectos de software para ser trabajado de forma colaborativa por varias personas a la vez. Además Github permite el control de versiones. Asegúrese de tener una cuenta creada en este enlace: [Sign in to GitHub · GitHub](#)
3. Crear un nuevo repositorio: Dentro de su cuenta de Github, diríjase a la parte superior izquierda, en la sesión llamada repositorio y de click en New.

Search or jump to... [Pull requests](#) [Issues](#) [Marketplace](#) [Explore](#)

Recent Repositories [New](#)

Find a repository...

Learn Git and GitHub without any code!  
Using the Hello World guide, you'll create a repository, start a bra



4. Agregue un nombre a su repositorio y seleccione el repositorio como público para que pueda ser visto por otros en internet. Luego de esto de click al botón de create repository.

## Create a new repository


A repository contains all project files, including the revision history. Already have a project repository elsewhere?

[Import a repository.](#)

---

Owner \*

Repository name \*

 rebe-ov ▾


 / 

my\_project ✓


Great repository names are short and memorable. Need inspiration? How about [bug-free-train?](#)

Description (optional)

---

☒  **Public**

Anyone on the internet can see this repository. You choose who can commit.

☐  **Private**

You choose who can see and commit to this repository.

---

Initialize this repository with:

Skip this step if you're importing an existing repository.

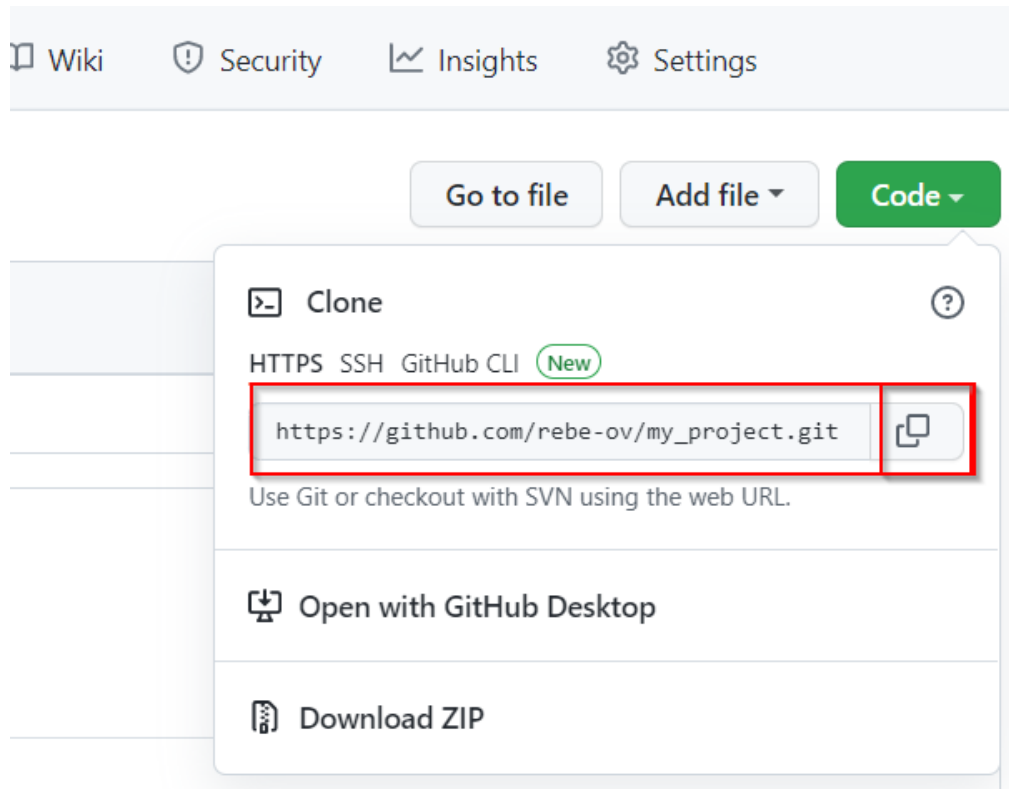
☒ **Add a README file**

This is where you can write a long description for your project. [Learn more.](#)

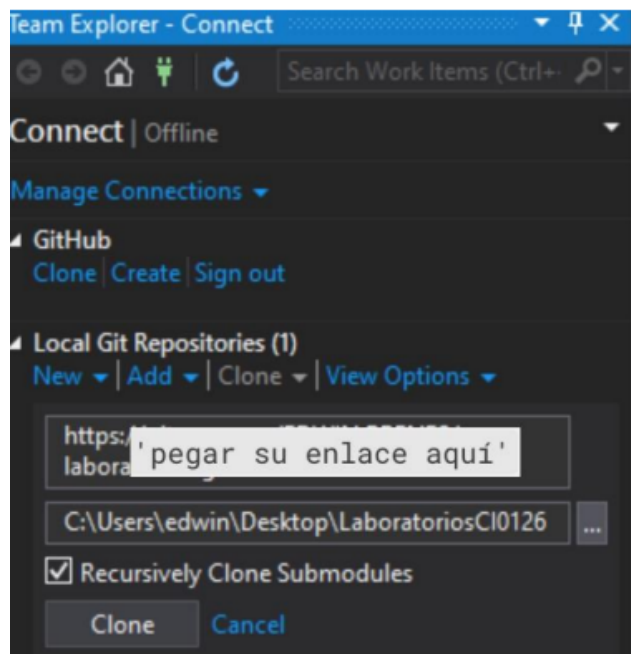
☐ **Add .gitignore**

Choose which files not to track from a list of templates. [Learn more.](#)

5. Vincule su nuevo repositorio a Visual Studio: En el botón de *code* copie link HTTPS que aparece.



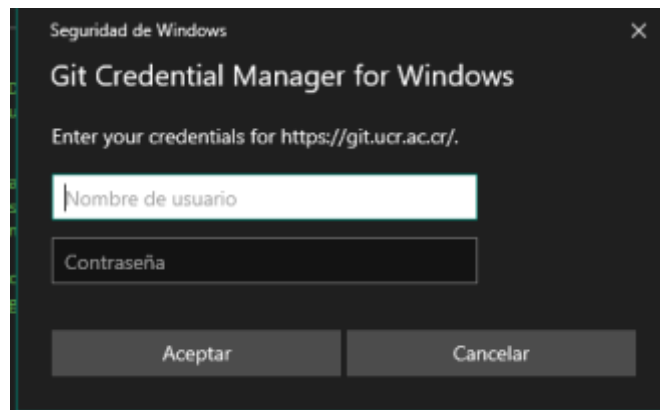
6. Abra Visual Studio, en el menú bajo *View* seleccione la opción llamada *Team Explorer*. En el panel de *Team Explorer*, seleccione la opción *Clone* bajo la sección *Local Git Repositories*. Y copie el enlace en espacio correspondiente.



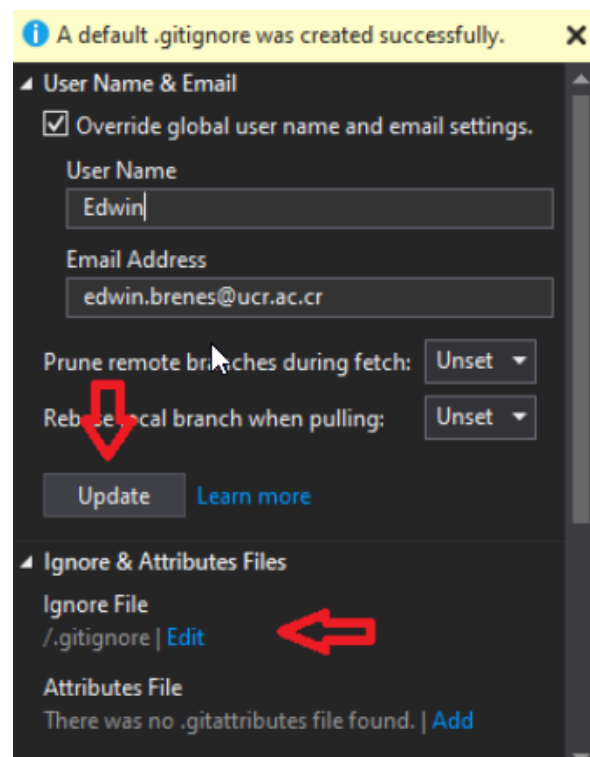
Finalmente de click en el botón de *clone*, esta acción puede tomar uno minutos.



- Posteriormente aparecerá una ventana solicitando las credenciales de su cuenta GitHub. Agregue la información y de clic en Aceptar. En este momento el repositorio será clonado.



- Es momento de configurar la conexión del repositorio a Visual Studio. Para este debe ir *home* dentro del panel de *Team explorer*. Ahí seleccione la opción de *settings* y posteriormente *repository settings*. En este espacio debe completar los datos con su información y en la parte donde de *Ignore File*, debe dar clic en *Add*, y luego dar clic en *update*.

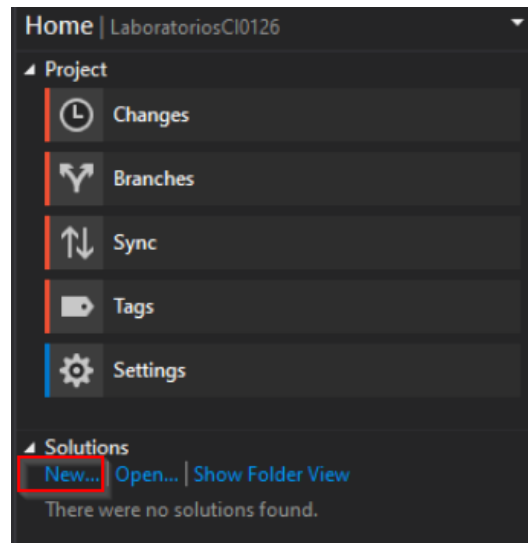


Nota: El archivo gitignore es un archivo que contiene reglas para indicar a git cuales archivos deben ser ignorados cuando se modifican o crean. De esta manera estos archivos no serán tomados en cuenta para el control de versiones y no se subirán al

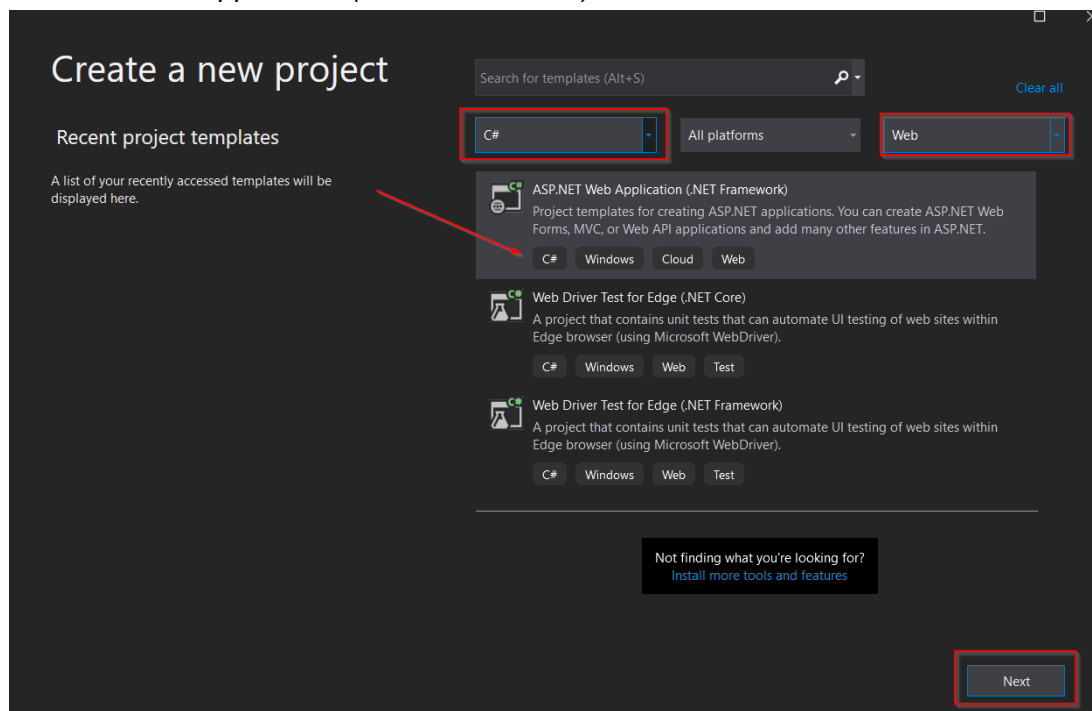


repositorio. Ejemplos de archivos: logs, dependencias externas, archivos con credenciales. Para mayor información sobre los archivos .gitignore revise el siguiente enlace [¿Qué es un archivo GITIGNORE y cómo se abre uno? | TecnoNautas](#)

9. Crear un proyecto en ASP.NET: Regrese al home del Team Explorer, en la sección de Solutions, de clic en new para crear una solución. Una solución es un contenedor para organizar uno o más proyectos de código relacionados, como un proyecto de biblioteca de clases y un proyecto de prueba correspondiente.



10. Seleccione el lenguaje C# y un proyecto tipo Web. Ahí seleccione un proyecto ASP.NET Web application (.NET Framework). De clic en next





11. A continuación agregue un nombre al proyecto y de clic en *create*

Configure your new project

ASP.NET Web Application (.NET Framework) C# Windows Cloud Web

Project name  
Laboratorio1

Location  
C:\Users\rebe\source\repos

Solution  
Create new solution

Solution name ⓘ  
Laboratorio1

☐ Place solution and project in the same directory

Framework  
.NET Framework 4.7.2

Back Create

12. A continuación le aparecerá una pantalla con el tipo de template de proyecto web que desea crear. Seleccione MVC. También puede utilizar la opción de *Also create a project for unit test*. Esta opción creará un proyecto para manejar las pruebas de su solución. Finalmente de clic en *create*

Create a new ASP.NET Web Application

**Empty**  
An empty project template for creating ASP.NET applications. This template does not have any content in it.

**Web Forms**  
A project template for creating ASP.NET Web Forms applications. ASP.NET Web Forms lets you build dynamic websites using a familiar drag-and-drop, event-driven model. A design surface and hundreds of controls and components let you rapidly build sophisticated, powerful UI-driven sites with data access.

**MVC**  
A project template for creating ASP.NET MVC applications. ASP.NET MVC allows you to build applications using the Model-View-Controller architecture. ASP.NET MVC includes many features that enable fast, test-driven development for creating applications that use the latest standards.

**Web API**  
A project template for creating RESTful HTTP services that can reach a broad range of clients including browsers and mobile devices.

**Single Page Application**  
A project template for creating rich client side JavaScript driven HTML5 applications using ASP.NET Web API. Single Page Applications provide a rich user experience which includes client-side interactions using HTML5, CSS3, and JavaScript.

**Authentication**  
No Authentication  
[Change](#)

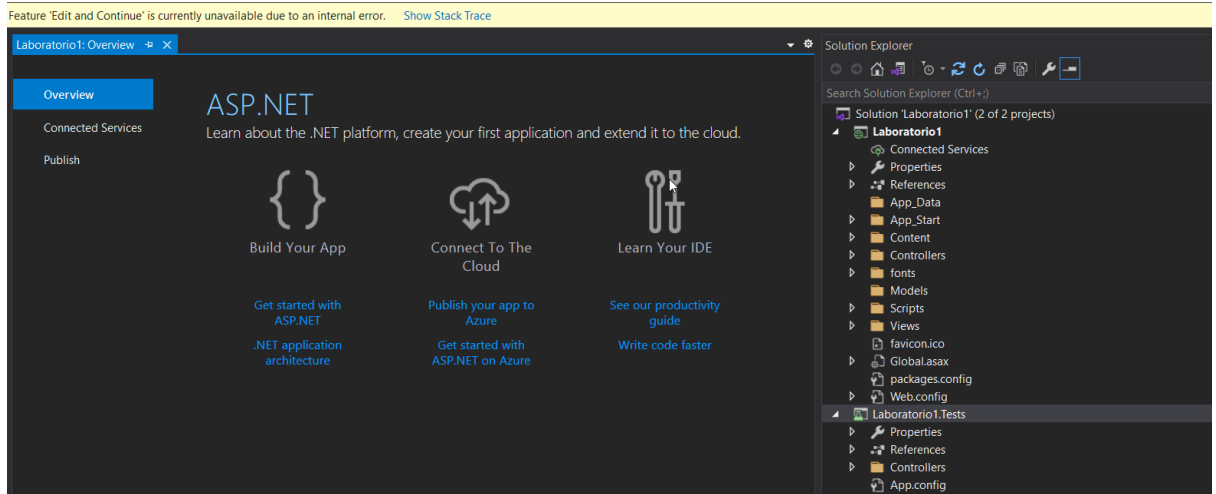
**Add folders & core references**  
☐ Web Forms  
☒ MVC  
☐ Web API

**Advanced**  
☒ Configure for HTTPS  
☐ Docker support  
(Requires [Docker Desktop](#))  
☒ Also create a project for unit tests  
Laboratorio1.Tests

Back Create



13. Seguidamente aparecerá la información sobre la solución creada



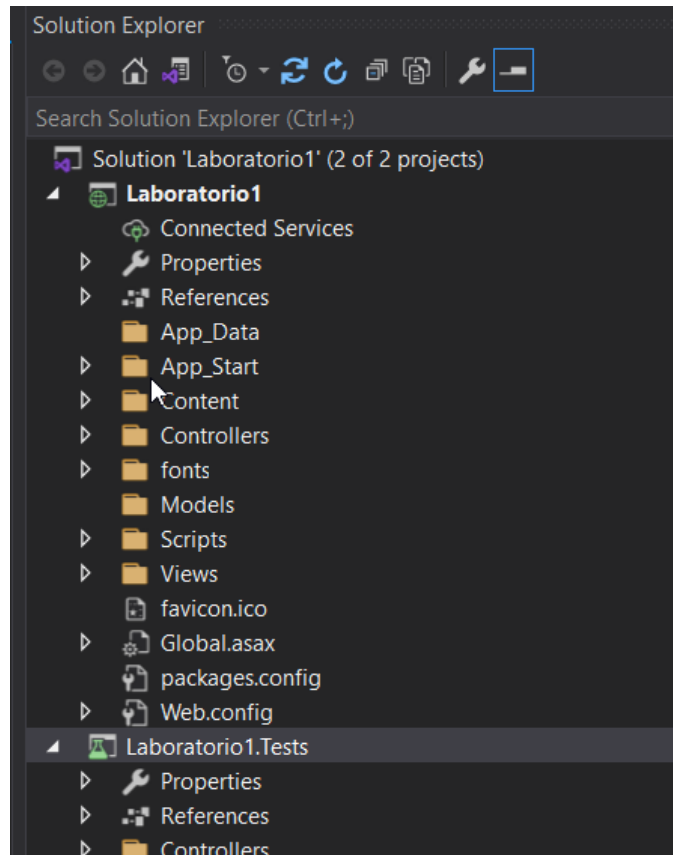
14. Tome un tiempo para revisar que es ASP.NET y las diferentes opciones que ofrese:

<https://dotnet.microsoft.com/en-us/learn/aspnet/what-is-aspnet>

15. Si no puede ver el solution explorer en el menú superior seleccione View -> Solution Explorer. En este panel podrá revisar los diferentes componentes que componen la solución, desde el proyecto principal de codificación (Laboratorio1) como el proyecto de pruebas unitarias Laboratorio1.Test. En laboratorios posteriores se explicará cómo funciona MVC, pero puede revisar cómo funciona este patron aca:

<https://dotnet.microsoft.com/en-us/apps/aspnet/mvc>

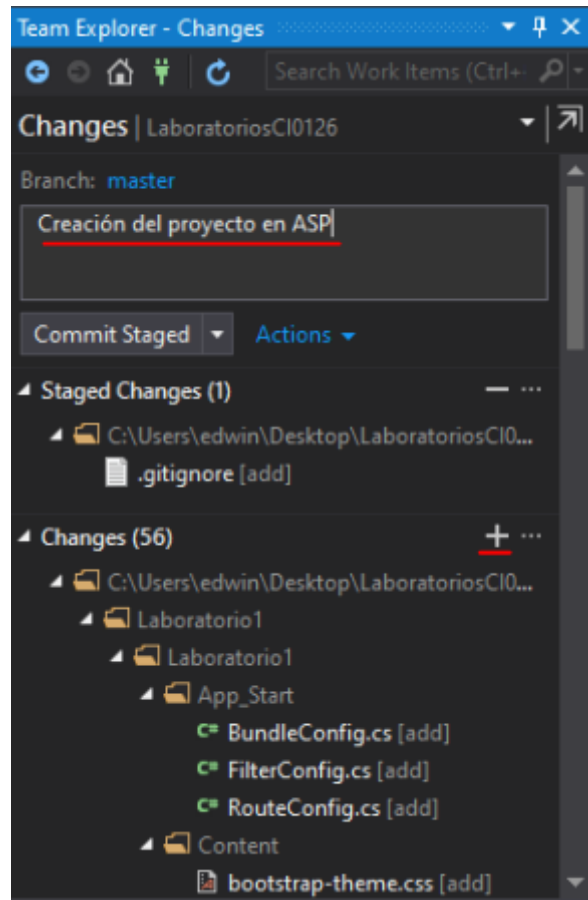




16. Actualizar el repositorio: Para poder subir la solución que se acaba de crear diríjase al *Team Explorer -> home* ahí busque las opción de *Git changes*.

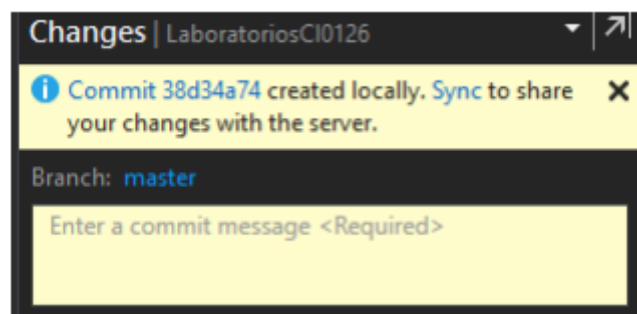
Va a notar por ejemplo que en la sección de Changes aparece un 56 (o un número similar). Esto se refiere al número de archivos modificados o agregados que no están en la última versión que tenía el repositorio (lo clonado inicialmente). Estos archivos fueron creados durante la creación del proyecto.

Agregue un comentario significativo sobre el comentario, además presione el botón de + (staged all) que funciona igual que el git add (incluye los archivos). y se presiona el botón *commit staged*.



\*\* Para quienes han usado Git en consola luego del git add, se escribe un comentario que describan los cambios realizados

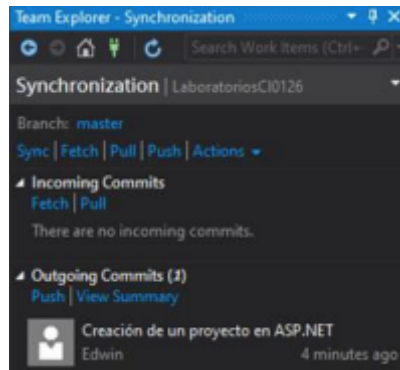
De esta manera los cambios se guardarán de manera local, sin embargo los cambios aún no se han guardado en el repositorio remoto (GitHub). Cuando se realiza un commit con éxito se recibe el siguiente mensaje:



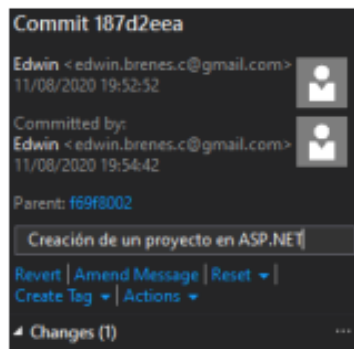
17. Ingrese a su repositorio en GitHub y verifique que efectivamente, aún no hay ningún cambio. Es momento de enviar los cambios al repositorio remoto, para ello se debe volver al *Team Explorer*, ahí se debe seleccionar la opción de sync.



En la sección de Outgoing Commits (1) se indica que hay un cambio con su respectivo *commit* pendiente de ser enviado al repositorio remoto. En esta sección aparecen los diferentes cambios pendientes de ser sincronizados con el repositorio, en este caso solo hay uno.

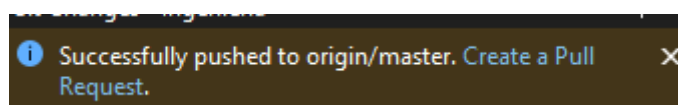


18. Dele click al *commit* respectivo y podrá ver opciones similares a la de la siguiente imagen:



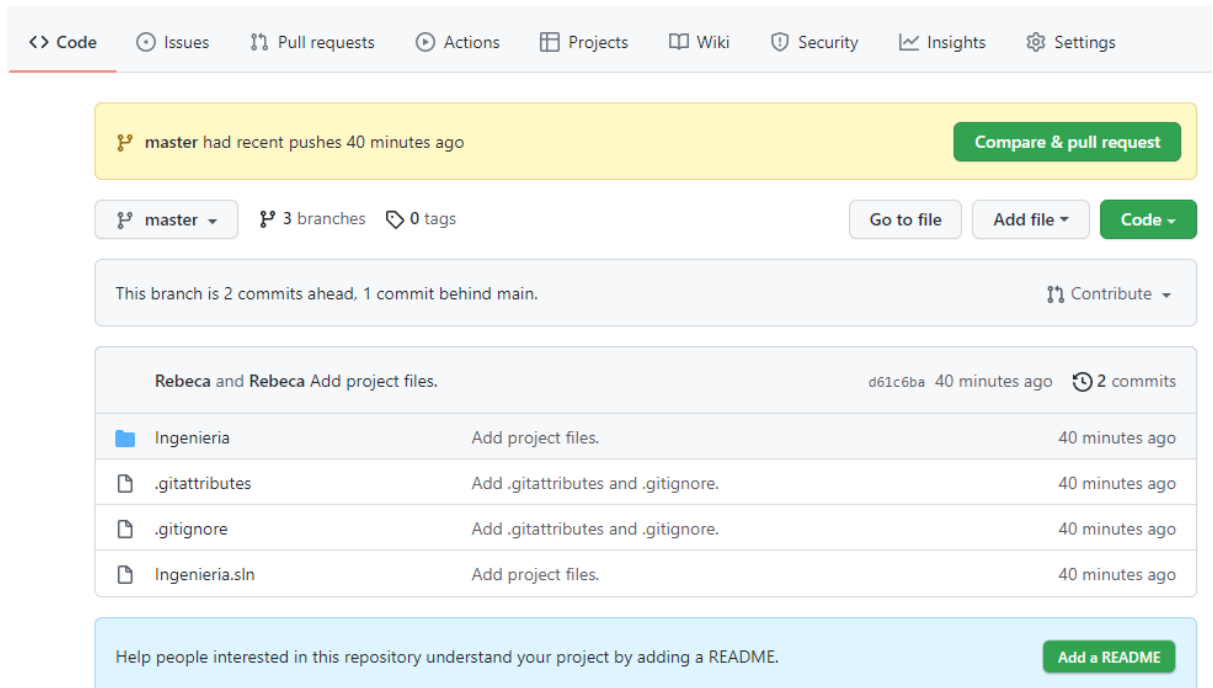
19. La imagen anterior muestra todas las acciones que usted puede realizar sobre el *commit*, desde modificarlo hasta anularlo (*revert*), también se puede modificar el comentario (*amend message*) o eliminar cambios (*reset*). También puede generar *tags* (etiquetas) Para mas información de como funcionan los *tags* diríjase: [Tags con git](#)

Es necesario destacar que los cambios que se hagan sobre el *commit* van a requerir sincronización y finalmente hacer push. Para este ejemplo no se hará ningún cambio sobre el *commit* original. Por tanto regrese al menú de Sync y luego presione la opción de Push, opción disponible en la imagen del punto 7 (outgoing commits). Una vez presionado recibirá la siguiente notificación:



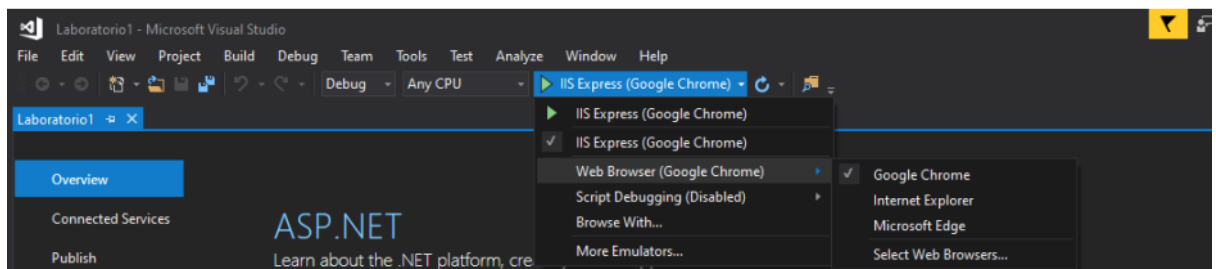


20. Vuelva a GitHub, en la sección de *code* podrá ver los cambios que se les acaba de hacer *push*.



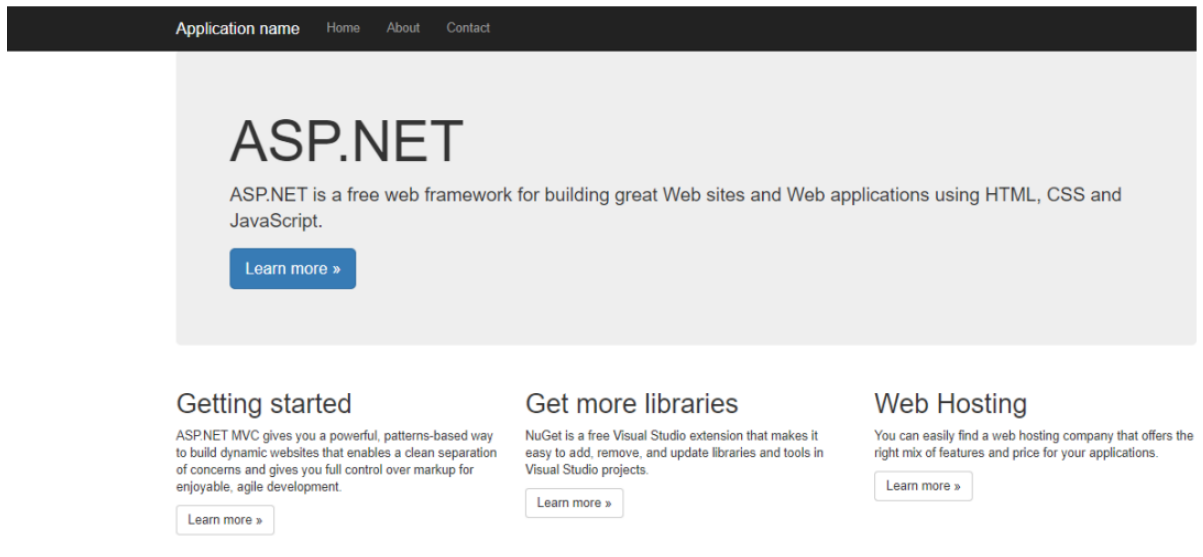
21. Compilación y ejecución del proyecto: Para poder compilar y ejecutar el proyecto diríjase a la parte superior del menú y de click en el botón *IIS Express (Google Chrome)*. Este botón compila y ejecuta el programa. Aquí usted puede seleccionar el navegador en el que desea abrir la solución web. A la izquierda se puede observar una opción *Debug*, esta opción se puede cambiar a *Release*. El modo *release* es más eficiente y permite un mejor aprovechamiento de los recursos, por lo cual si su computadora posee capacidades limitadas es recomendable que ejecute el proyecto en este modo. Sin embargo el modo Release se utiliza para ver los cambios visuales y el funcionamiento del sistema.

Por otro lado, el modo debug se recomienda cuando se desea depurar errores o rastrear el comportamiento del sistema.

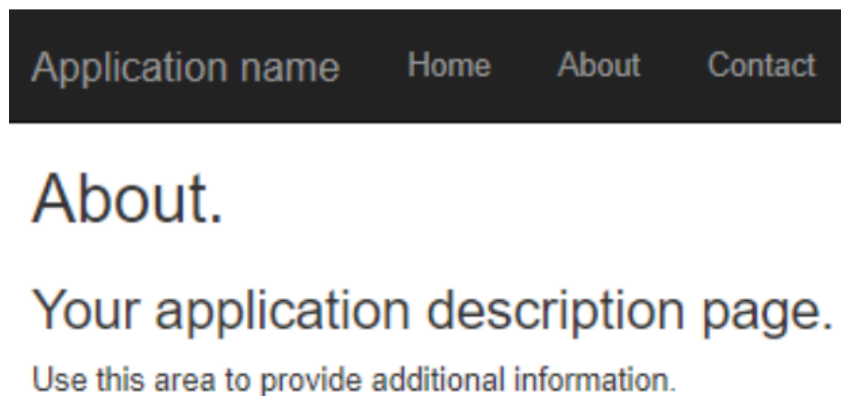




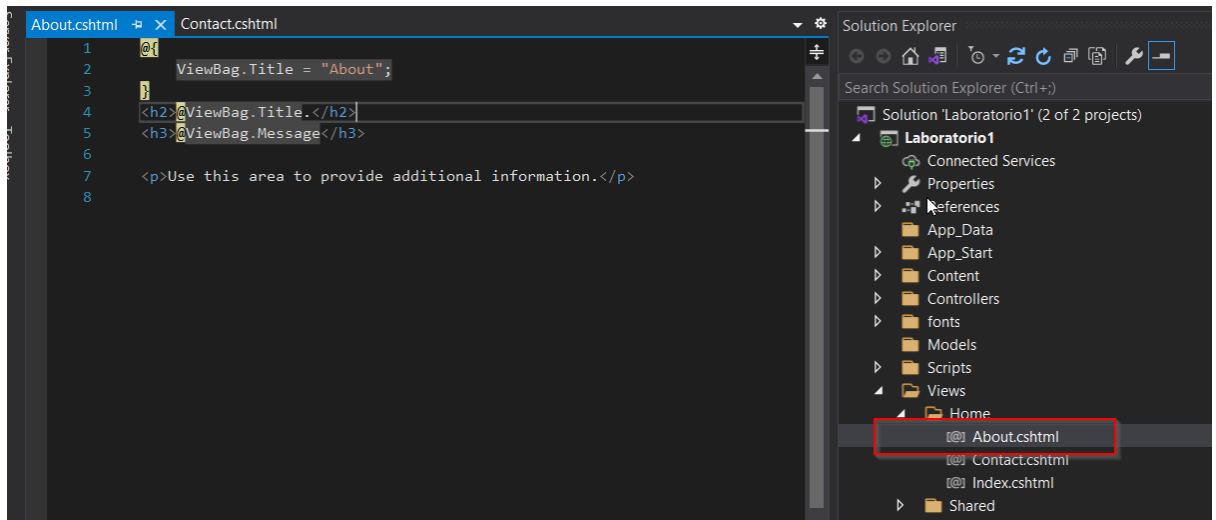
Nota: otra manera de ejecutar el programa en modo *release* es presionando las teclas **ctrl +F5** o también simplemente **F5** si quiere usar el modo *debug*. Una vez que levante la solución, el navegador seleccionado abriera y podrá ver un resultado como el siguiente:



22. Ingrese a la página de *About*, observe el contenido actual. En el siguiente paso haremos algunas modificaciones en este contenido.



23. Regrese a Visual Studio. En el panel de *Solution Explorer*. Busque la carpeta **Views** -> **Home**. Ahí encontrará 3 archivos de tipo cshtml, demos click sobre el archivo *About.cshtml* y podrá observar el siguiente código.



24. Este código hace referencia al contenido de la vista web llamada *About* (punto 22). Los archivos de extensión cshtml son ficheros denominados **Razor Pages**, los cuales son “compilados” por un motor de vistas llamado **Razor**. Las *Razor pages* tienen gran utilidad debido a que permiten escribir código HTML pero también C#, lo cual puede tener diferentes usos. Estos conceptos serán abordados en laboratorios posteriores.

A continuación se tiene una explicación rápida del código:

- Las etiquetas HTML se escriben con la misma sintaxis y el mismo propósito que los archivos HTML.
- La sintaxis `@ { }` sirve para indicar que todo el contenido dentro de los corchetes precedidos por el `@`, será código C# y en consecuencia todo el código escrito en su interior debe respetar la sintaxis del mismo (C# es muy similar a Java y C++).
- Se puede mezclar contenido HTML con C#. Para ello el código C# o sus variables tiene como prefijo el `@`
- ViewBag es un contenedor que puede ser manipulado desde los controladores de la aplicación.

Nota: En los siguientes laboratorios se verá a fondo MVC y estos conceptos.



25. Modifique el archivo About.cshtml: para cambiar el contenido de esta vista utilice el siguiente código:

```
@{
    ViewBag.Title = "Mi primera vista";
    String nombreAutor = "Su nombre";
    int nota = 100;
}

<html>
<head>
    <title> @ViewBag.Title </title>
</head>
<body>
    <div>
        <h1>Mi primera vista</h1>
        <h2>¡Hola Mundo! </h2>
        <p>
            <strong>Yo soy: </strong> @nombreAutor <br />
            Mi nota el día de hoy es: <input type="number" value="@nota" placeholder="Escriba su nota" />
        </p>
        <hr />
    </div>

    <div>
        <label>Caja para texto: </label>
        <input type="text" placeholder="Escriba cualquier cosa" />
        <hr />
    </div>

    <div>
        <a class="btn btn-info" href="/Home/Index"> Volver a la página de inicio </a> <br />
    </div>
</body>

</html>
```

26. Refresque la vista del navegador presionando **F5** en el navegador o bien realice la ejecución del sistema si la detuvo anteriormente. Verifique que tiene un resultado como el siguiente:



Application name   Home   About   Contact

# Mi primera vista

## ¡Hola Mundo!

Yo soy: Edwin Brenes

Mi nota el día de hoy es:

Caja para texto:

Volver a la página de inicio

27. Suba los cambios a GitHub haciendo un nuevo commit y un push como los descritos en los puntos del 16 al 20.

### Entregable:

Cree un documento donde muestre screenshots:

1. La creación del repositorio
2. La creación del proyecto
3. los cambios en la página About
4. El commit
5. Los dos commits en su cuenta de GitHub

En cada uno de los screenshots debe aparecer su nombre de usuario, ya sea de visual studio, de su computadora, de su cuenta Github.

Además adjunte el link de su repositorio GitHub en el documento, para que pueda ser descargado.

**Material Elaborado** por Edwin Brenes

**Atualizado** por Rebeca Obando