

## Laboratorio 03



### Instalación GitHub y Visual Studio

## Objetivos

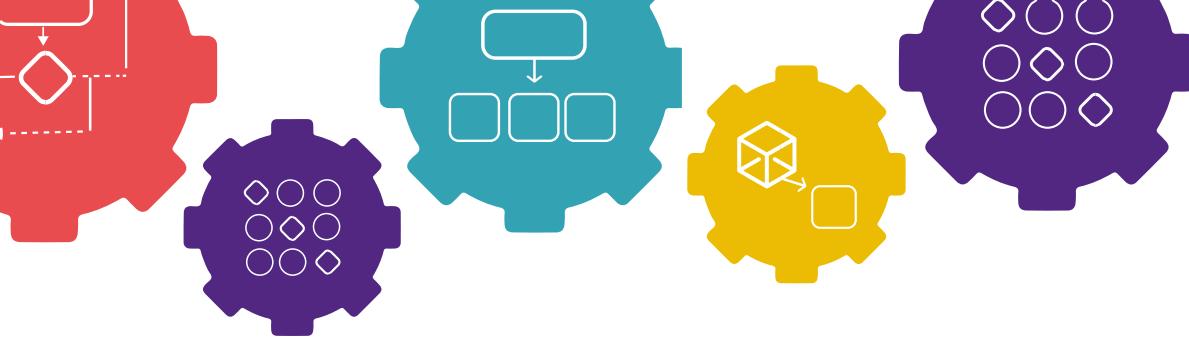
- Utilizar la plataforma de GitHub como repositorio para el portafolio para el curso.
- Instalar el entorno de desarrollo para codificar con el lenguaje de programación.



## Conceptos clave

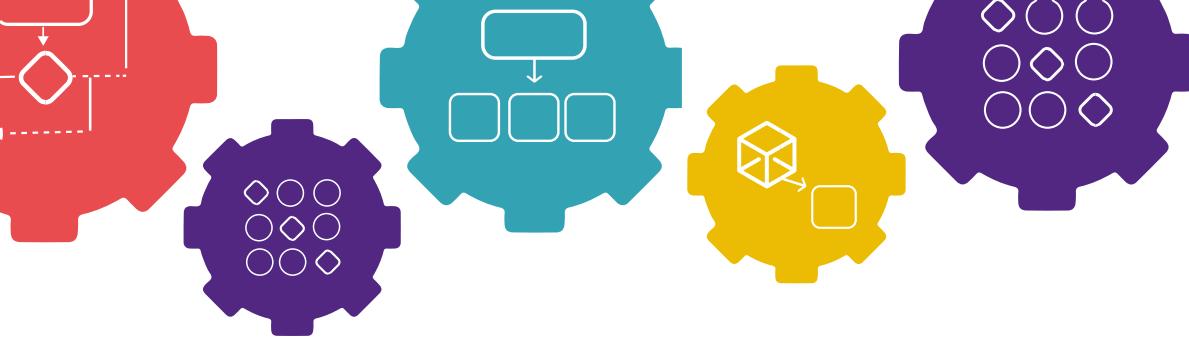
### Repositorio

- La Real Academia Española define este concepto como “Lugar donde se guarda algo”. Durante esta práctica se utilizará un repositorio de archivos digitales para que los estudiantes puedan guardar sus archivos para ir creando su portafolio.
- **GitHub**
  - Es una plataforma donde se puede almacenar, administrar y compartir código de proyectos y aplicaciones. Emplea un sistema de control de versiones que permite ir gestionando los cambios que se realizan en los archivos.
- **Comandos básicos:**
  - **Clone:** se utiliza para clonar o crear una copia de un repositorio en una computadora



- **Pull:** descarga el contenido del repositorio remoto para actualizar el repositorio local
  - **Add:** agrega un cambio al índice actual, para que sea añadido al repositorio remoto en el siguiente commit.
  - **Commit:** crea una captura del proyecto actual con todos los cambios añadidos con el comando add
  - **Push** permite subir o publicar los cambios locales al repositorio remoto
- **Programa:**
    - Un programa es un conjunto ordenado de instrucciones que le indican a la computadora qué acciones debe realizar para resolver un problema específico.
  - **Librerías:**
    - Las librerías son conjuntos de funciones y clases previamente definidas que facilitan el desarrollo de programas. En C#, se incluyen mediante la palabra clave using.
  - **Espacio de nombres (Namespace):**
    - Permite organizar el código y evitar conflictos de nombres.
  - **Clase:**
    - La clase define la estructura principal del programa y contiene métodos y variables.
  - **Método principal (Main):**
    - Es el punto de inicio de la ejecución del programa.
  - **Variables:**
    - Una variable es un espacio en memoria que almacena un valor que puede cambiar durante la ejecución.
  - **Bloques de código:**
    - Los corchetes {} delimitan bloques de instrucciones.
  - **Paréntesis y punto y coma:**
    - Los paréntesis se utilizan para pasar información a métodos y el punto y coma indica el fin de una instrucción.
  - **Ejemplo:**

```
// Librería:  
// Permite utilizar funcionalidades básicas del sistema,  
// como mostrar mensajes y leer datos del usuario.  
using System;  
// Espacio de nombres:  
// Se utiliza para organizar el código del programa.  
namespace Laboratorio3  
{
```



```
// Clase principal del programa
class Program
{
    // Método principal:
    // Punto de inicio de la ejecución del programa.
    static void Main(string[] args)
    {
        // Write:
        // Muestra un mensaje en pantalla SIN hacer salto de línea.
        Console.Write("Este mensaje se muestra ");

        // WriteLine:
        // Muestra un mensaje y luego hace un salto de línea.
        Console.WriteLine("en la misma línea.");

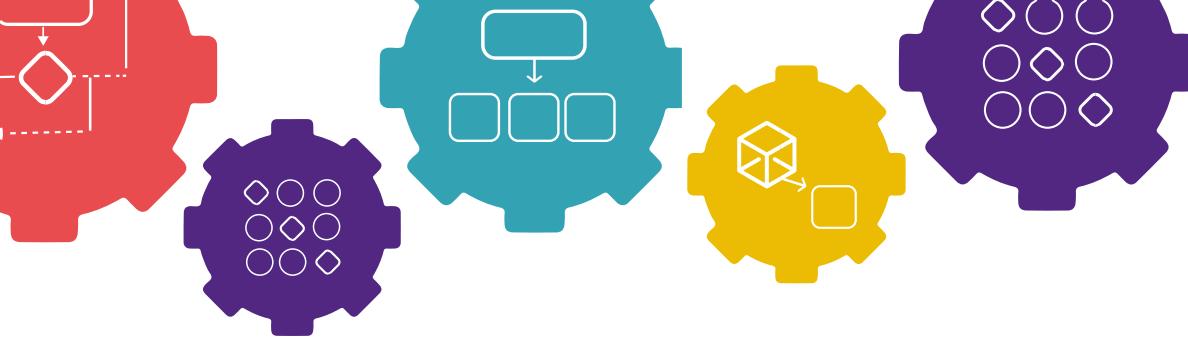
        // El cursor pasa automáticamente a la siguiente línea
        Console.WriteLine("Este mensaje aparece en una nueva línea.");

        // Variable de tipo texto
        string nombre;

        // Solicitud de datos usando Write (el cursor se queda en la misma línea)
        Console.Write("Ingrese su nombre: ");

        // ReadLine:
        // Lee una línea completa de texto ingresada por el usuario.
        nombre = Console.ReadLine();

        // Uso de la variable en una salida
        Console.WriteLine("Hola " + nombre);
```



```
// Mensaje informativo  
Console.WriteLine("Presione cualquier tecla para continuar...");  
  
// ReadKey:  
// Espera a que el usuario presione una tecla.  
// No requiere presionar Enter.  
Console.ReadKey();  
  
// El programa finaliza después de presionar una tecla
```

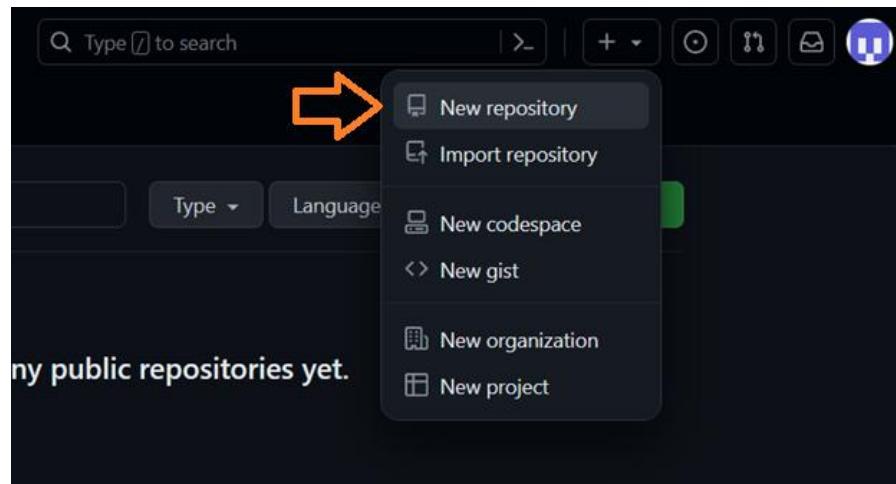
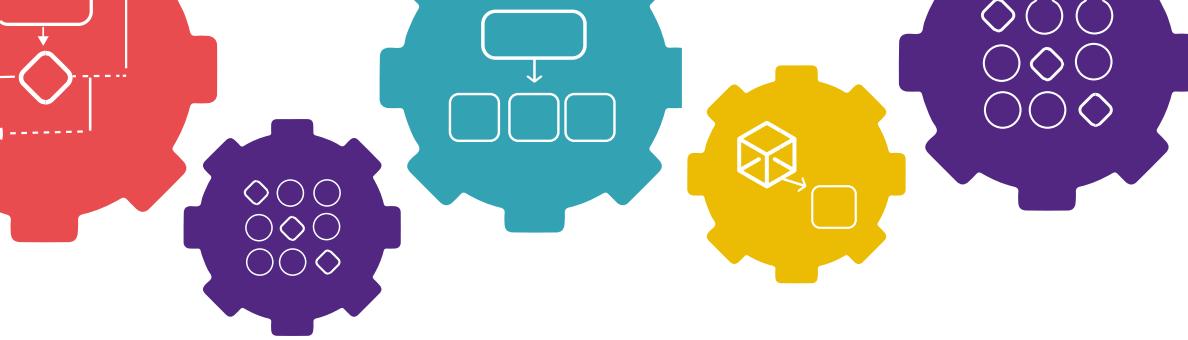
```
}
```



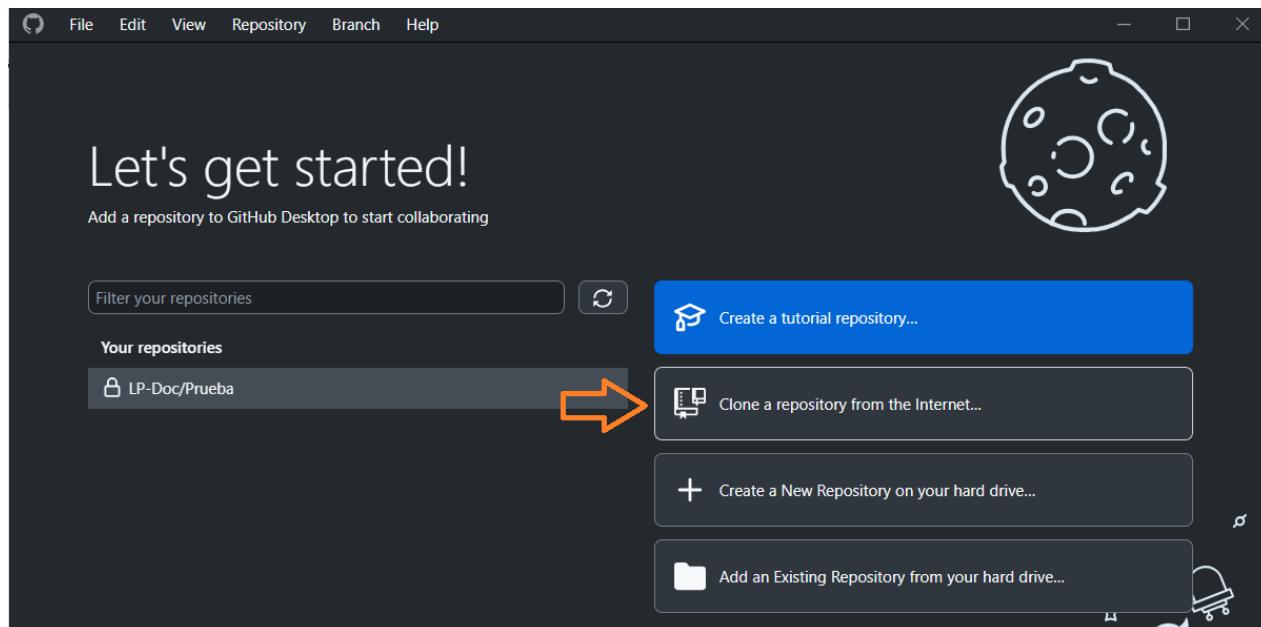
## Actividad No. 01

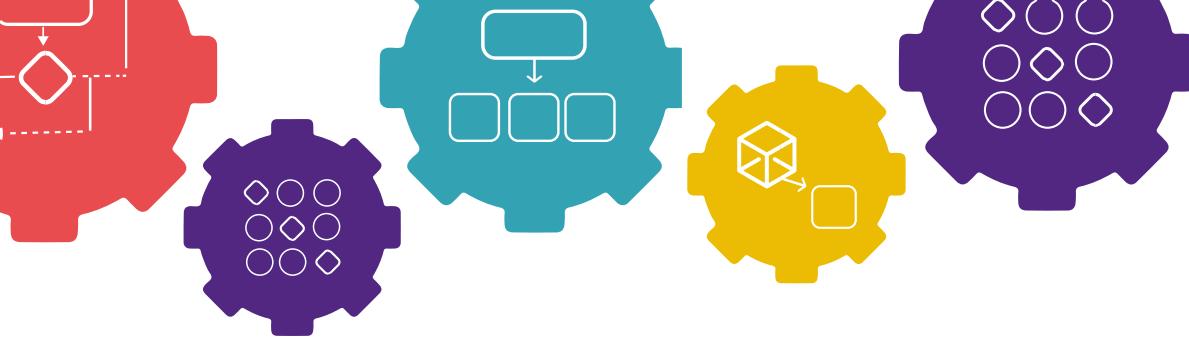
Durante esta actividad se procederá a crear un repositorio para que el estudiante pueda archivar sus documentos para el portafolio del curso.

1. Crear cuenta
  - a. Ingresar a <https://github.com/>
  - b. Ir a opción Sing up
    - i. Ingresar datos
2. Iniciar sesión
  - a. Responder preguntas de configuración de cuenta
  - b. Escoger el plan gratis o “free”
3. Crear nuevo repositorio

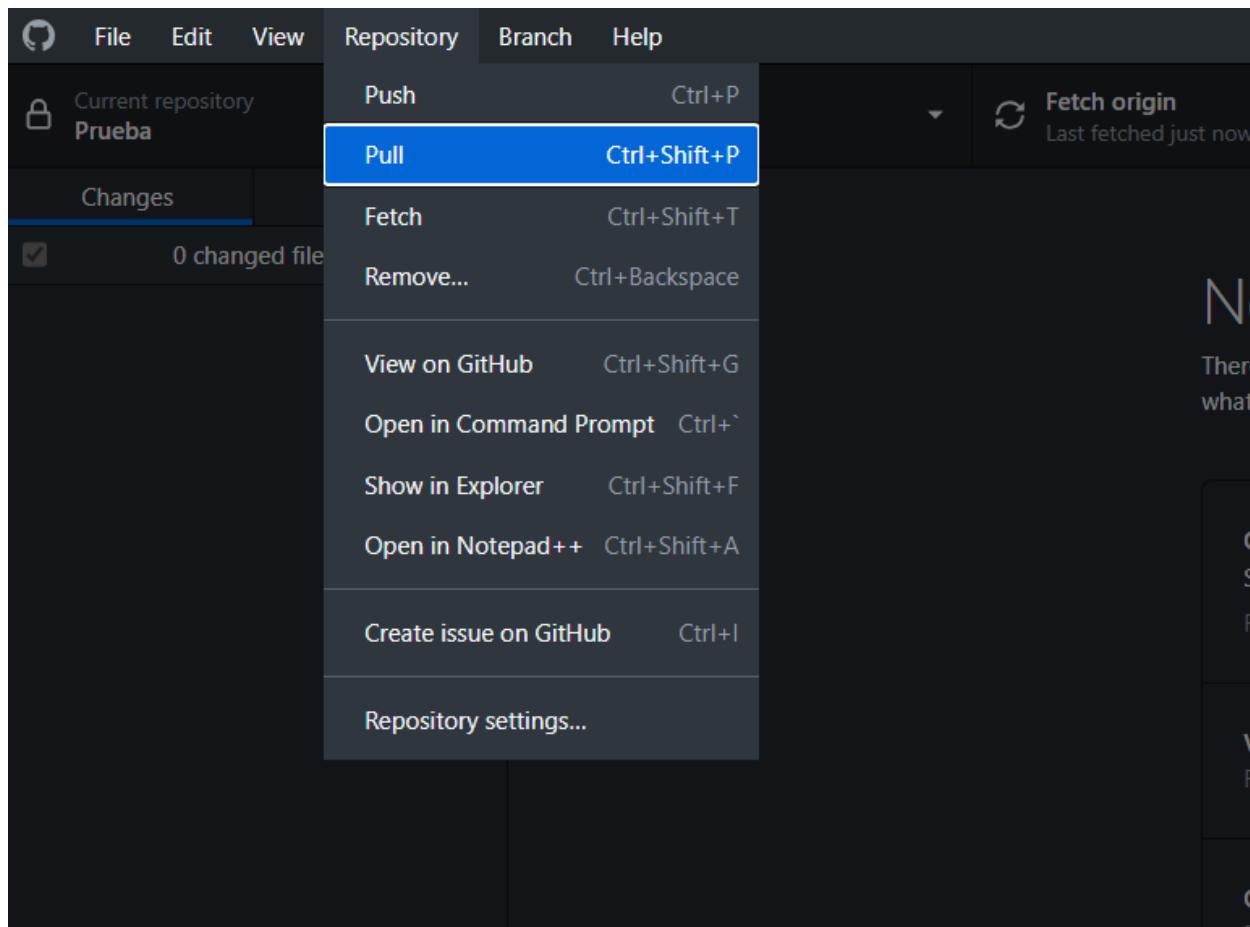


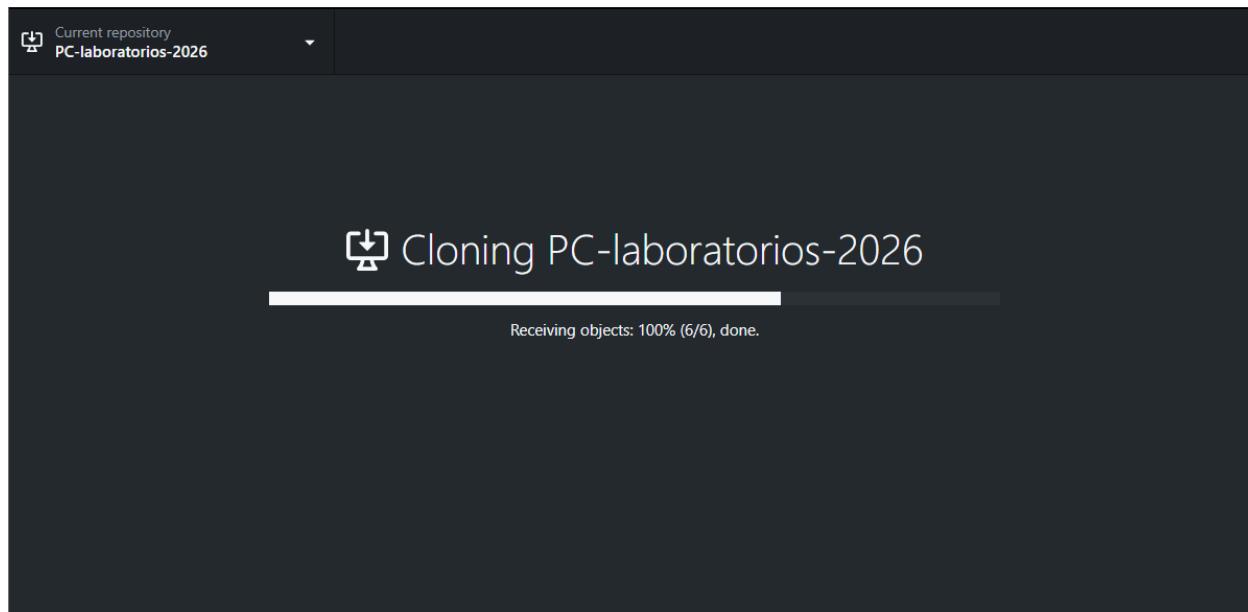
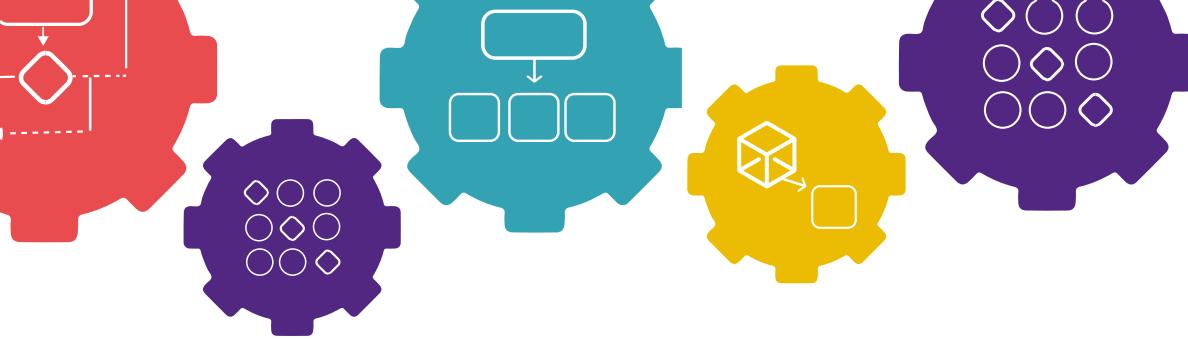
- a. Ingresar nombre del repositorio como: "Laboratorios"
- b. Marcar opción publico
- c. Marcar opción "Add README file"
4. Instalar GitHub Desktop
  - a. Ingresar a <https://desktop.github.com/>
  - b. Descargar y ejecutar programa
  - c. Iniciar sesión y configurar datos
5. Clonar repositorio



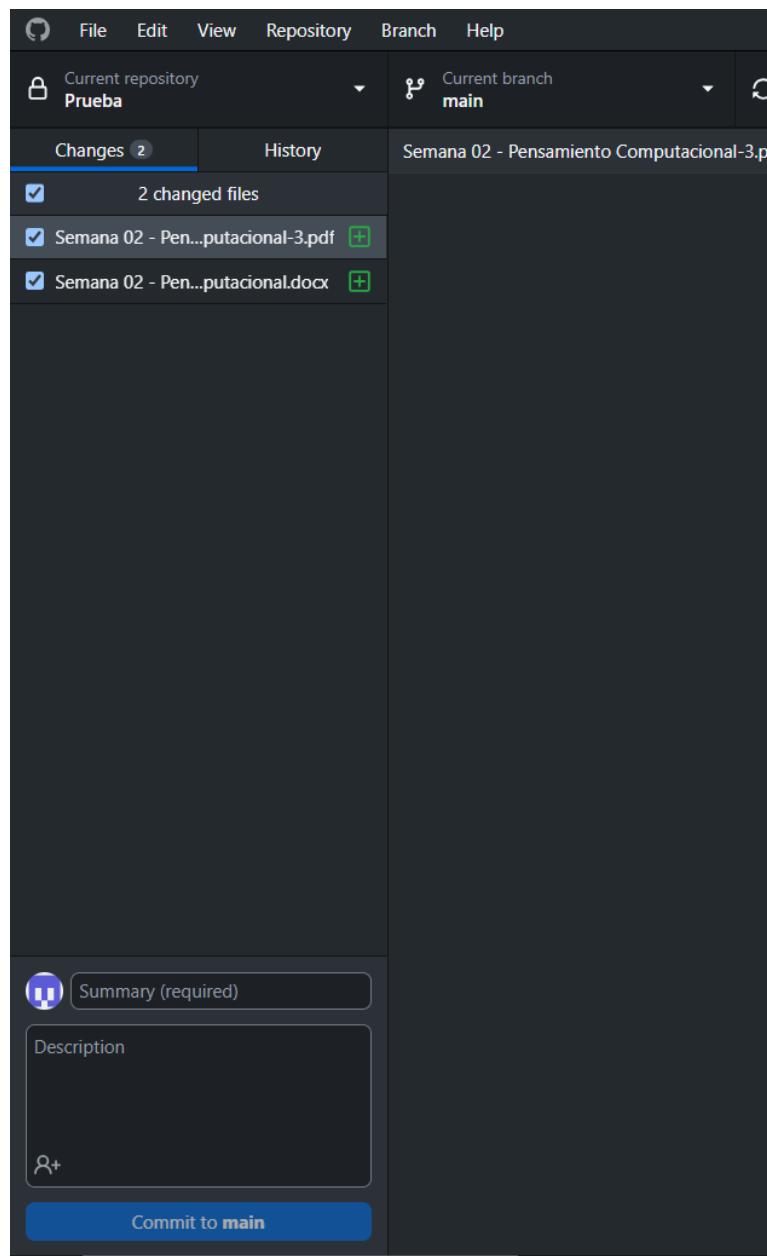
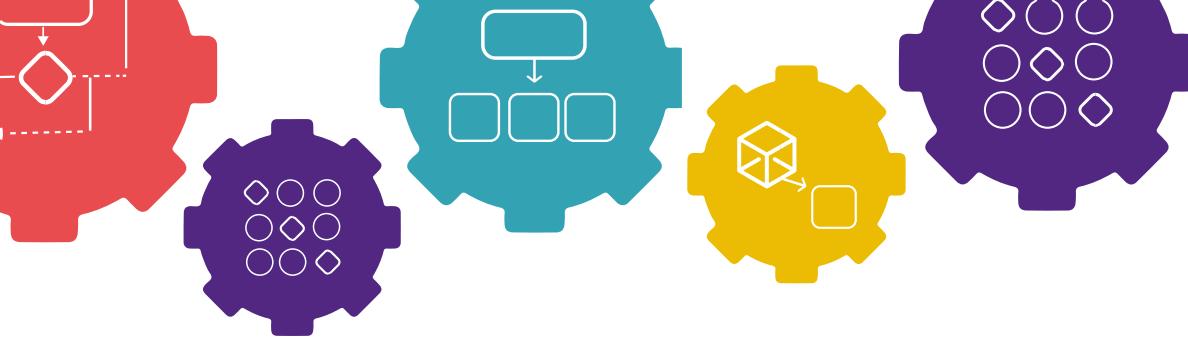


- a. Seleccionar el repositorio creado anteriormente
- b. ¿Cómo se puede verificar que se haya clonado el repositorio? Adjunte una captura de pantalla
6. Utilizar el comando pull para actualizar los archivos locales (Los que se encuentran en el equipo que se está utilizando).

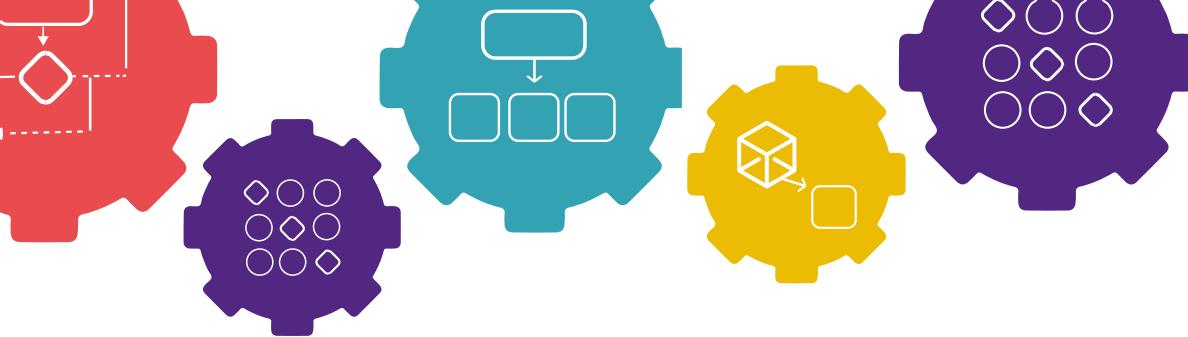




7. Para agregar un nuevo archivo al repositorio se deben de seguir lo siguientes pasos:
  - a. Ir al explorador de archivos
  - b. En este caso agregar el o los documentos PDF generados en prácticas anteriores a la carpeta local de archivos, la dirección en donde se clonó el repositorio.
  - c. Las cajas de selección al lado de los archivos representan comandos “add” para agregar los cambios al repositorio



- d. En la parte inferior izquierda se debe de agregar un título y una descripción del commit a realizar.
- e. Ir al menú principal, opción repository y seleccionar Push
8. Valide en la página web de GitHub que su repositorio haya sido actualizado.
9. Realizar cambios en un archivo:
  - a. Realizar un cambio en el archivo README y subir cambios al repositorio



- b. Ver historial de cambios del archivo. Se puede consultar en la pagina web y/o en la aplicación de escritorio.
  - i. Dentro de los cambios de cada commit aparece información como los archivos editados, el usuario, y fecha realizada. Ejemplo:

**x Free disk space.**

maryam/free-disk-space (#6277)

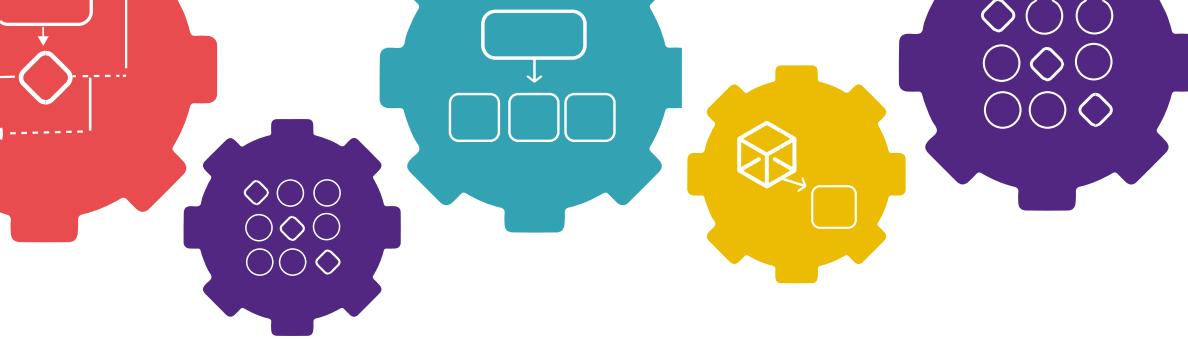
 maryamziaa committed 2 weeks ago Verified

▼ ↑ 6 .github/workflows/colab.yml

...	@@ -35,6 +35,12 @@ jobs:
35	35 - uses: actions/setup-dotnet@v4
36	36 with:
37	37 dotnet-version: '6.0.x'
38	+ - name: Free disk space
39	+ run:
40	+ sudo rm -rf /opt/ghc
41	+ sudo rm -rf /usr/local/share/boost
42	+ sudo rm -rf "\$AGENT_TOOLS DIRECTORY"
43	+ df -h
38	- name: Cache pip
39	45 uses: actions/cache@v4
40	46 with:

**10. Adjunte evidencia en un archivo PDF de la actividad realizada:**

- a. Enlace de su usuario de GitHub
- b. Captura de pantalla de:
  - i. Repositorio creado
  - ii. Archivos locales del repositorio clonado
  - iii. Historial de cambios de un archivo README.



Screenshot of a GitHub repository page for "PC-laboratorios-2026".

Repository details:

- Owner: 1251224
- Name: PC-laboratorios-2026
- Visibility: Público
- Last commit: f251bdb - Hace 12 horas
- Branch: principal
- Commits: 1
- Issues: 0
- Pull requests: 0
- Labels: 0
- Wiki: Wiki
- Security: Seguridad
- Perspectives: Perspectivas
- Settings: Ajustes

Commit history:

Author	Message	Time
1251224	Actualizar Laboratorio_5.cs	Hace 12 horas
	Laboratorio_1_listo.pdf	Agregar archivos mediante carga
	Laboratorio_2_Listo.pdf	Agregar archivos mediante carga
	Laboratorio_5.cs	Actualizar Laboratorio_5.cs
	README.md	Confirmación inicial

File list:

- LÉAME
- README.md
- Laboratorio\_1\_listo.pdf
- Laboratorio\_2\_Listo.pdf
- Laboratorio\_5.cs

Project information:

- Acerca de: No se proporciona descripción, sitio web ni temas.
- Lanzamientos: No hay lanzamientos publicados. [Crear un nuevo lanzamiento](#).
- Paquetes: No hay paquetes publicados. [Publica tu primer paquete](#).

Screenshot of a GitHub Changes view for the "PC-laboratorios-2026" repository.

Current repository: PC-laboratorios-2026

Current branch: main

Fetch origin: Last fetched just now

Changes tab (selected):

- 2 changed files
- Laboratorio\_5.cs
- README.md

History tab (disabled):

- 1 commit
- 1 file

Stashed changes:

Restore will move your stashed files to the Changes list.

File	Line	Content
Laboratorio_5.cs	1	@@ -1 +1,32 @@
Laboratorio_5.cs	2	- # PC-laboratorios-2026
Laboratorio_5.cs	3	+ Mala lic.
Laboratorio_5.cs	4	+ ..
Laboratorio_5.cs	5	+ ..
Laboratorio_5.cs	6	+ ..
Laboratorio_5.cs	7	+ ..
Laboratorio_5.cs	8	+ ..
Laboratorio_5.cs	9	+ ..
Laboratorio_5.cs	10	+ ..
Laboratorio_5.cs	11	+ ..
Laboratorio_5.cs	12	+ ..
Laboratorio_5.cs	13	+ ..
Laboratorio_5.cs	14	+ ..
Laboratorio_5.cs	15	+ ..
Laboratorio_5.cs	16	+ ..
Laboratorio_5.cs	17	+ ..
Laboratorio_5.cs	18	+ ..
Laboratorio_5.cs	19	+ ..
Laboratorio_5.cs	20	+ ..
Laboratorio_5.cs	21	+ ..
Laboratorio_5.cs	22	+ ..
Laboratorio_5.cs	23	+ ..
Laboratorio_5.cs	24	+ ..
Laboratorio_5.cs	25	+ ..
Laboratorio_5.cs	26	+ ..
Laboratorio_5.cs	27	+ ..
Laboratorio_5.cs	28	+ ..
Laboratorio_5.cs	29	+ ..
Laboratorio_5.cs	30	+ ..

Summary (required):

Description:

Commit 2 files to main



## Actividad No. 02

### Instalación de Visual Studio

A continuación, se listan los pasos para instalar Visual Studio. Visual Studio es el entorno de desarrollo de Microsoft. La edición Community es gratuita, y esta es la que utilizaremos. A continuación, encontrará los pasos para descargar e instalar:

1. Descargar de: <https://visualstudio.microsoft.com/es/downloads/>
2. Seleccionar la opción Community o Comunidad.
3. En el instalador, cuando pregunta qué deseamos instalar, marcar la casilla de Desarrollo de Escritorio de .NET

Instalación en curso — Visual Studio Community 2026 — enero 2026 Actualización de características (18.2.1)

Cargas de trabajo Componentes individuales Paquetes de idioma Ubicaciones de instalación

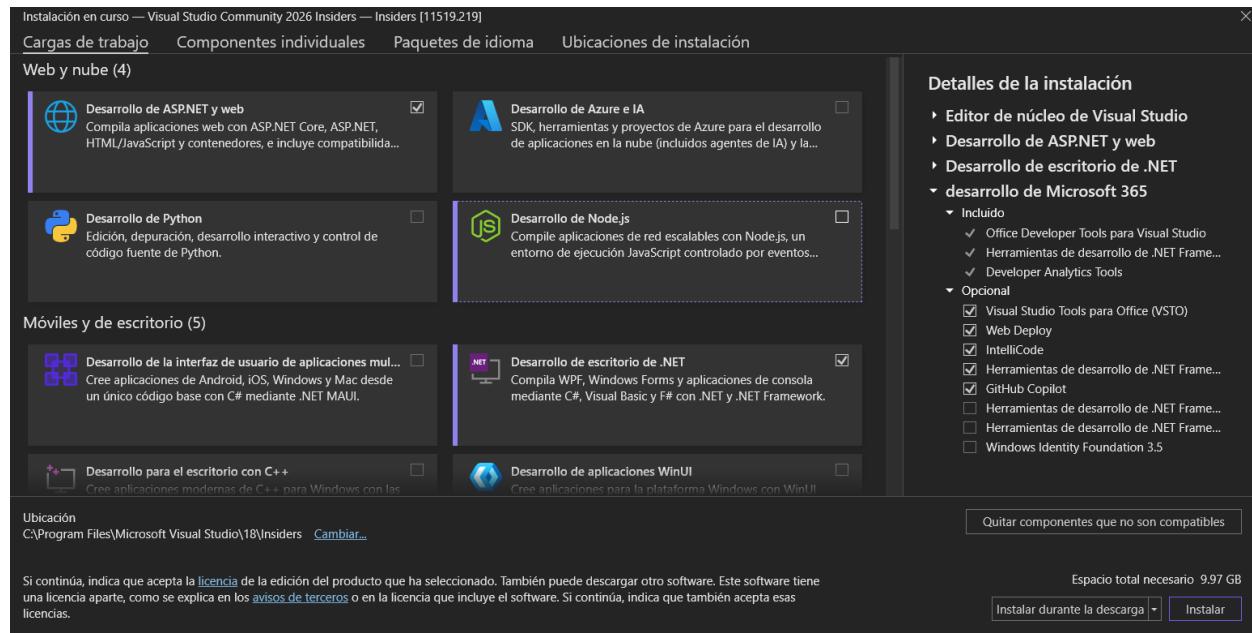
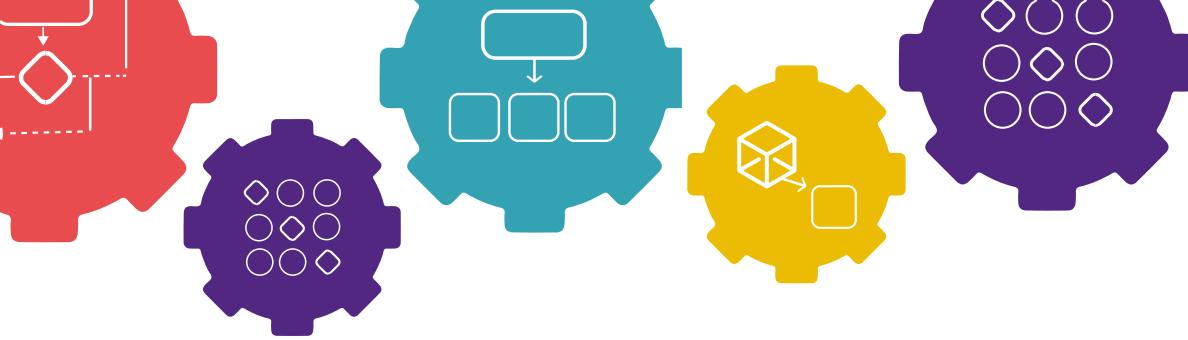
Web y nube (4)

Desarrollo de ASP.NET y web Compila aplicaciones web con ASP.NET Core, ASP.NET, HTML/JavaScript y contenedores, e incluye compatibilidad...	<input type="checkbox"/>	Desarrollo de Azure e IA SDK, herramientas y proyectos de Azure para el desarrollo de aplicaciones en la nube (incluidos agentes de IA) y la...	<input type="checkbox"/>
Desarrollo de Python Edición, depuración, desarrollo interactivo y control de código fuente de Python.	<input type="checkbox"/>	Desarrollo de Node.js Compile aplicaciones de red escalables con Node.js, un entorno de ejecución JavaScript controlado por eventos...	<input type="checkbox"/>

Móviles y de escritorio (5)

Desarrollo de la interfaz de usuario de aplicaciones mu... Cree aplicaciones de Android, iOS, Windows y Mac desde un único código base con C# mediante .NET MAUI.	<input type="checkbox"/>	Desarrollo de escritorio de .NET Compila WPF, Windows Forms y aplicaciones de consola mediante C#, Visual Basic y F# con .NET Framework...	<input type="checkbox"/>
Desarrollo para el escritorio con C++ Cree aplicaciones modernas de C++ para Windows con...	<input type="checkbox"/>	Desarrollo de aplicaciones WinUI Cree aplicaciones para la plataforma Windows con WinUI.	<input type="checkbox"/>

A red arrow points to the checkbox for "Desarrollo de escritorio de .NET".



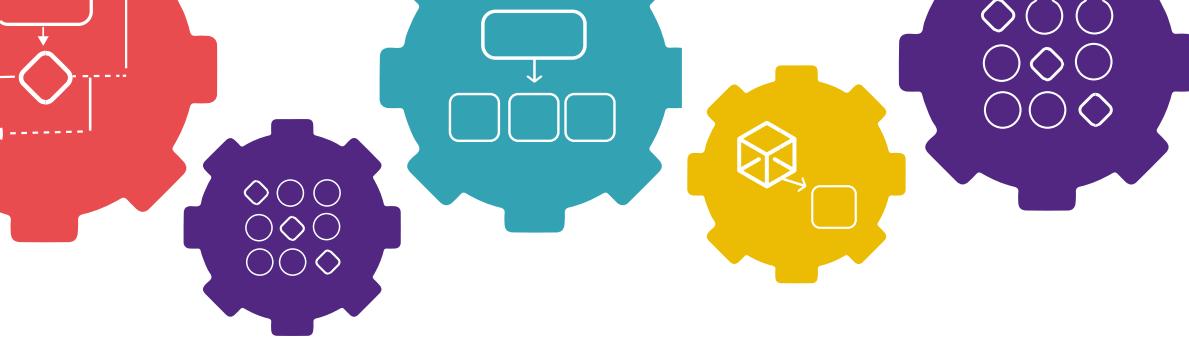
4. Después de que el programa se ha instalado, puede que solicite reiniciar la computadora.

5. Después de haber instalado el programa, abrir Visual Studio, se le solicitará que inicie sesión con una cuenta de correo electrónico de Microsoft (crearla si no se cuenta con una). Este paso es muy importante realizarlo, de lo contrario el software expirará en 30 días.

## Creación de un proyecto de consola en lenguaje C#

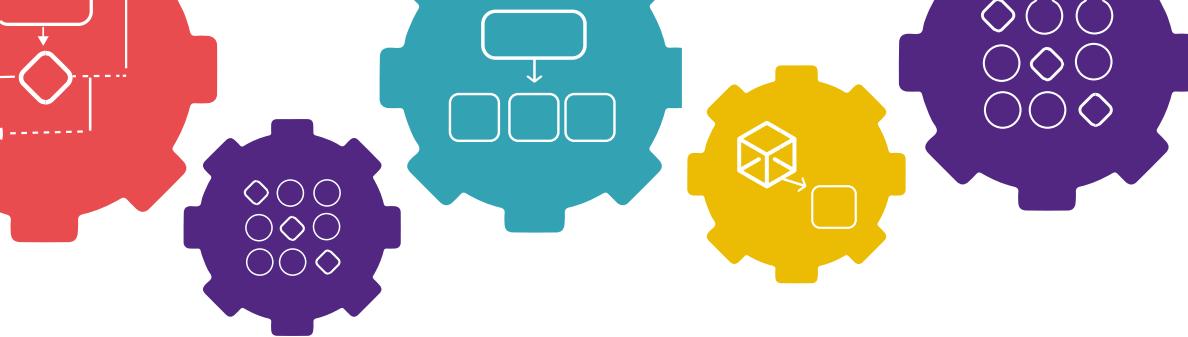
Una vez instalado completada la instalación, seguir estos pasos para crear un proyecto de consola en Visual Studio en el lenguaje de programación C#:

- 1. Abrir Visual Studio:** Al iniciar, selecciona "Crear un nuevo proyecto" en la ventana de inicio.



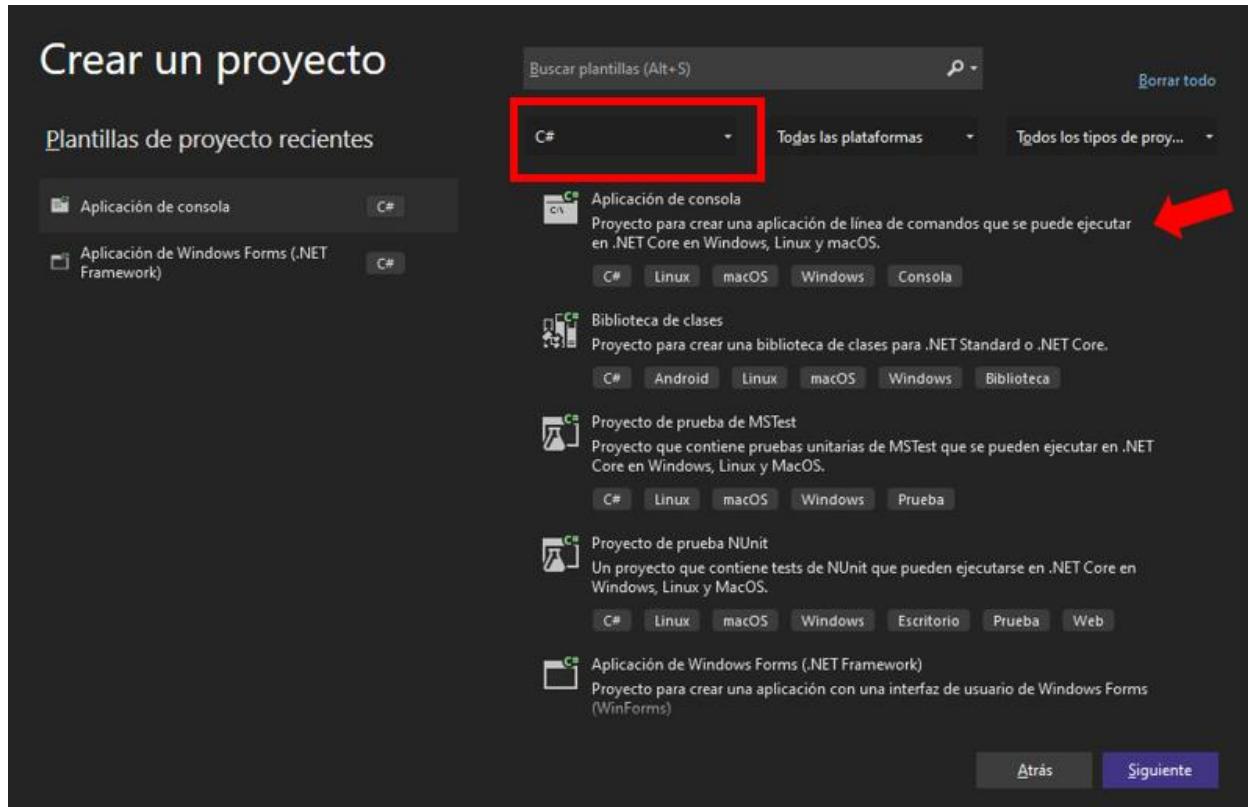
Tareas iniciales

-  **Clonar un repositorio**  
Obtiene código desde un repositorio en línea, como GitHub o Azure DevOps.
-  **Abrir un proyecto o una solución**  
Abre un archivo .sln o proyecto de Visual Studio local.
-  **Abrir una carpeta local**  
Navegar y editar el código en cualquier carpeta
-  **Crear un proyecto**  
Elija una plantilla de proyecto con la técnica scaffolding de código para comenzar.  
  
[Continuar sin código →](#)



**2. Filtrar Plantilla y seleccionar tipo de proyecto:** Seleccionar C# en el filtro de lenguaje.

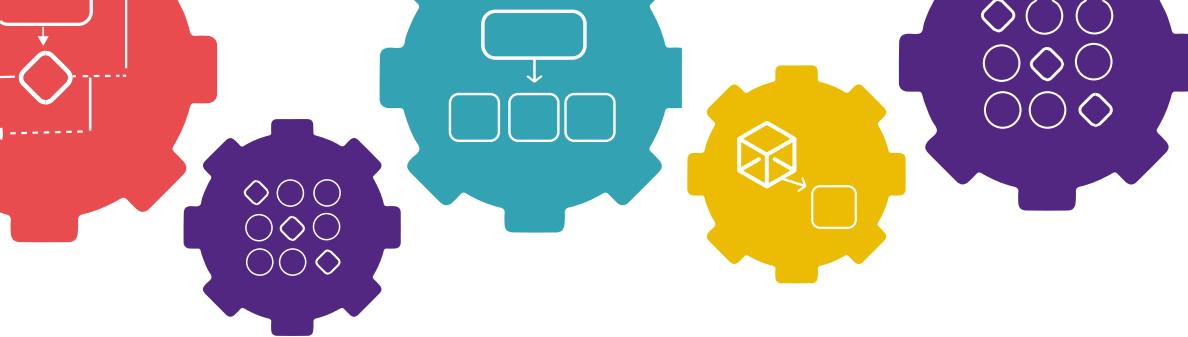
Elegir la plantilla llamada "Aplicación de consola" (que utiliza .NET) y hacer clic en Siguiente.



**3. Configurar Nombre:** Asignar un nombre al proyecto y seleccionar la ubicación donde se guardará. Para el nombre del proyecto, se colocará el número de Laboratorio, el número de Actividad, seguido de su nombre y apellido. Por ejemplo:

**Lab01-A1-Jorge Lopez**

**A lo largo del curso deberá nombrar todos sus proyectos de esta manera.**



**4. Primer programa:** Escribe el siguiente código para realizar tu primer programa en C#:

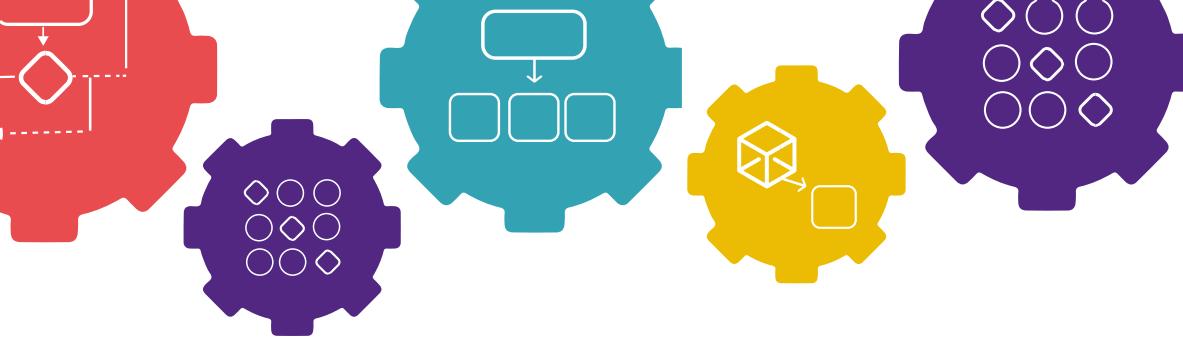
```
1 // See https://aka.ms/new-console-template for more information
2 Console.WriteLine("Hello, World!");
3 Console.WriteLine("Programa hecho por: Jorge Lopez");
4
5
```

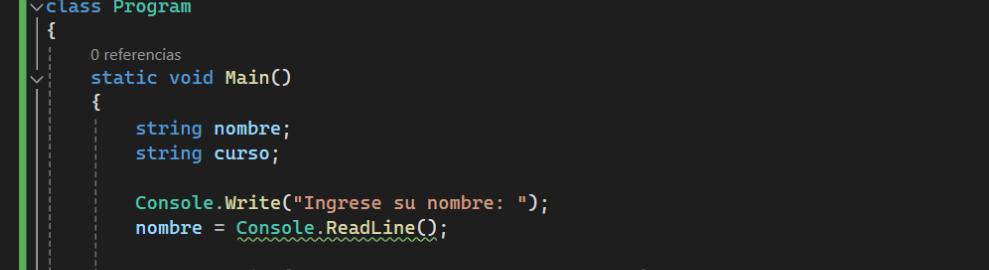
Para ejecutar el programa, presiona la tecla **F5**.

```
Hello, World!
Programa hecho por: Anderson Flores
C:\Users\flore\OneDrive\Documentos\Final_AngelFlores\Lab03-A20-AndersonFlores\Lab03-A20-AndersonFlores\bin\Debug\net
8.0\Lab03-A20-AndersonFlores.exe (proceso 14216) se cerró con el código 0.
Para cerrar automáticamente la consola cuando se detiene la depuración, habilite Herramientas ->Opciones ->Depuración ->
Cerrar la consola automáticamente al detenerse la depuración.
Presione cualquier tecla para cerrar esta ventana. . .
```

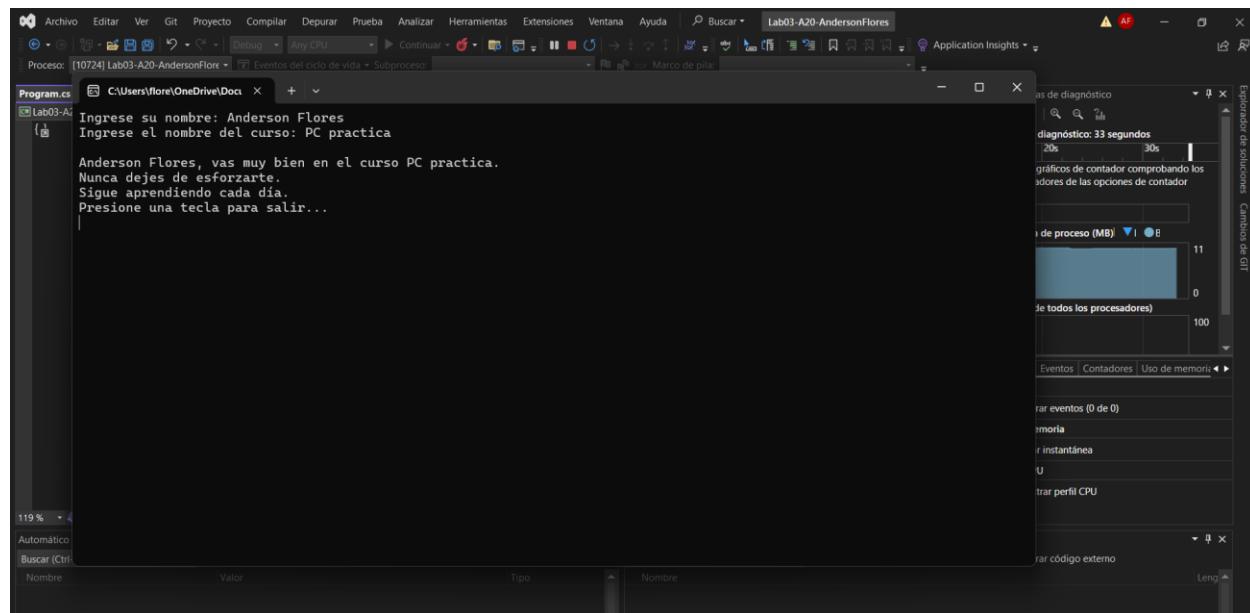
**El estudiante debe crear un programa en C# que:**

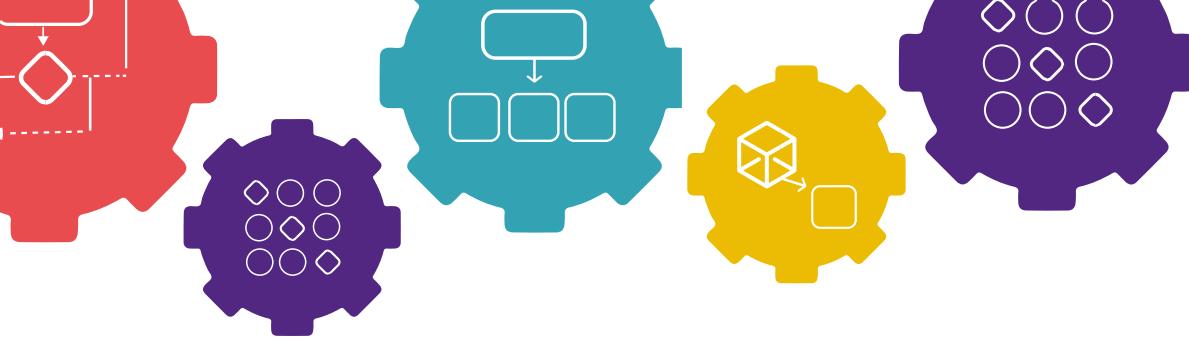
- Solicite el nombre del estudiante.
- Solicite el nombre del curso.
- Almacene los datos en variables.
- Muestre los datos anteriores y un mensaje personalizado creado por el estudiante en diferentes líneas.
- Terminar el programa presionando una tecla.





```
Program.cs* ✘ x
Lab03-A20-AndersonFlores Program Main()
1  using System;
2  class Program
3  {
4      static void Main()
5      {
6          string nombre;
7          string curso;
8
9          Console.Write("Ingrese su nombre: ");
10         nombre = Console.ReadLine();
11
12         Console.Write("Ingrese el nombre del curso: ");
13         curso = Console.ReadLine();
14
15         Console.WriteLine();
16         Console.WriteLine(nombre + ", vas muy bien en el curso " + curso + ".");
17         Console.WriteLine("Nunca dejes de esforzarte.");
18         Console.WriteLine("Sigue aprendiendo cada día.");
19         Console.WriteLine("Presione una tecla para salir...");
20
21         Console.ReadKey();
22     }
}
```





## Créditos

Este material fue elaborado tomando como referencia material docente desarrollado por el Ing. Luis Pedro Ovalle y el Ing. José Pablo Ovalle.

Material de uso académico interno. Universidad Rafael Landívar.