

Laboratorio 03



Instalación GitHub y Visual Studio

Objetivos

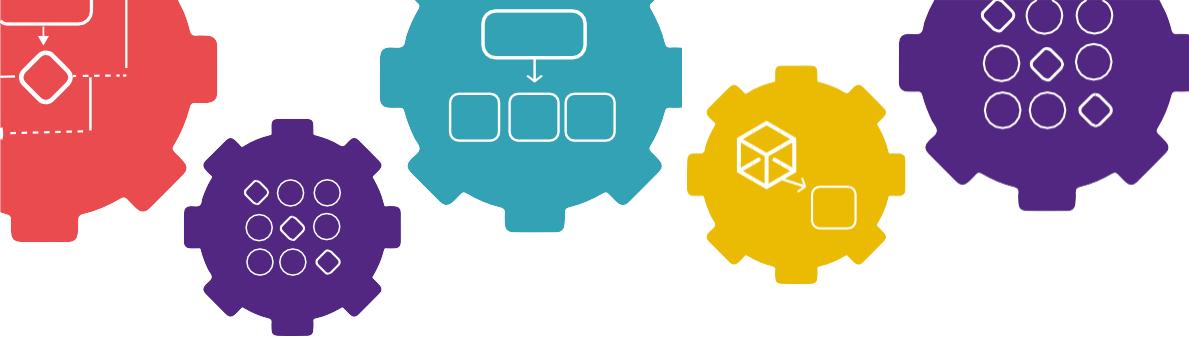
- Utilizar la plataforma de GitHub como repositorio para el portafolio para el curso.
- Instalar el entorno de desarrollo para codificar con el lenguaje de programación.



Conceptos clave

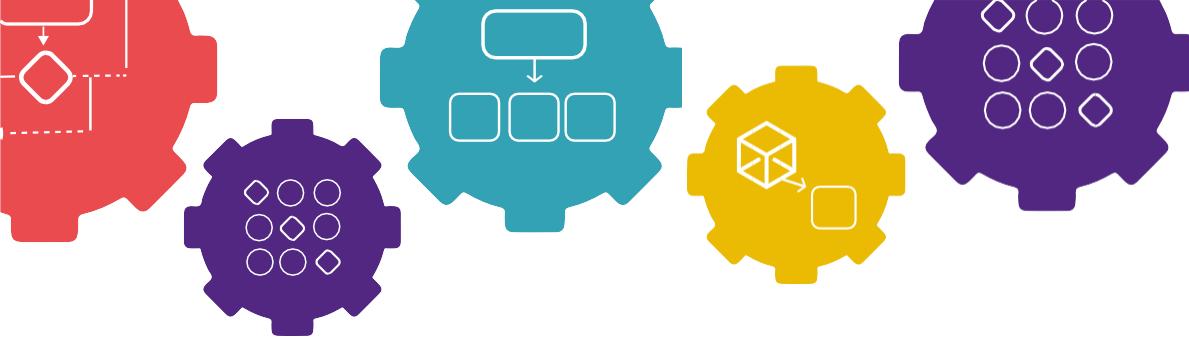
Repository

- 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:**

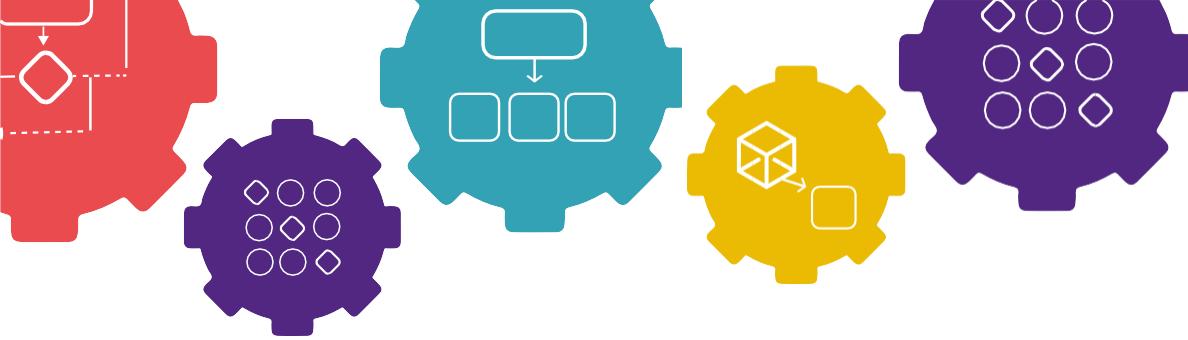
```
// Librería:  
// Permite utilizar funcionalidades básicas del sistema,  
// como mostrar mensajes y leer datos del usuario.  
using System;  
// Espacio de nombres:  
// Se utiliza para organizar el código del programa.  
namespace Laboratorio3  
{
```



```
// 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);
```



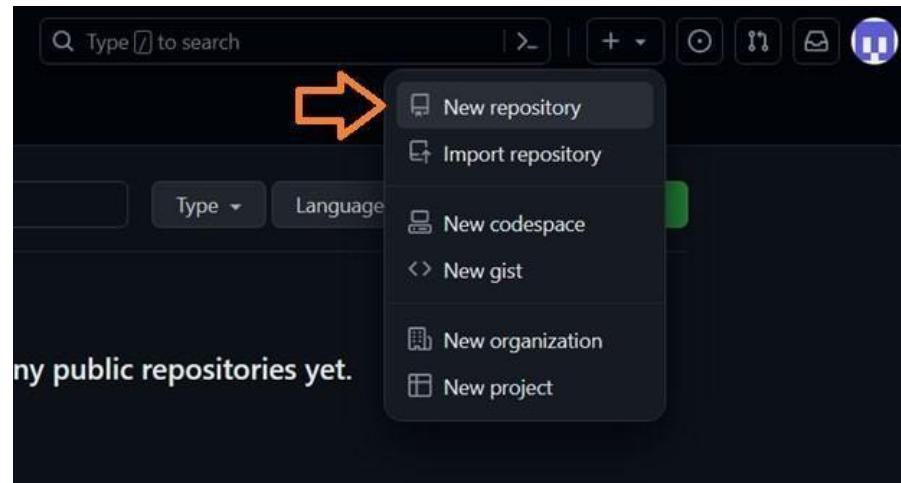
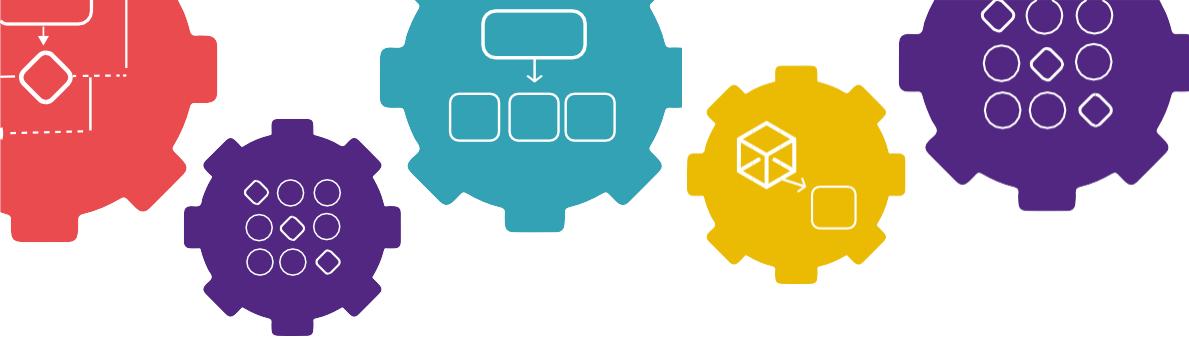
```
// Mensaje informativo  
Console.WriteLine("Presione cualquier tecla para continuar...");  
// ReadKey:  
// Espera a que el usuario presione una tecla.  
// No requiere presionar Enter.  
Console.ReadKey();  
// El programa finaliza después de presionar una tecla  
}  
}  
}
```



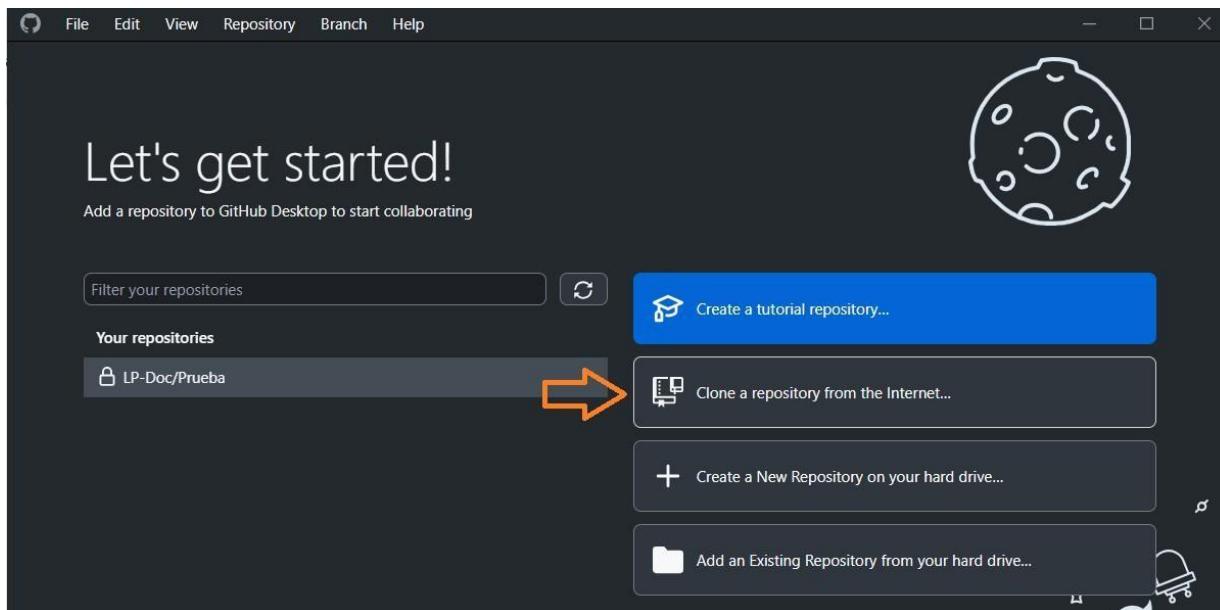
Actividad No. 01

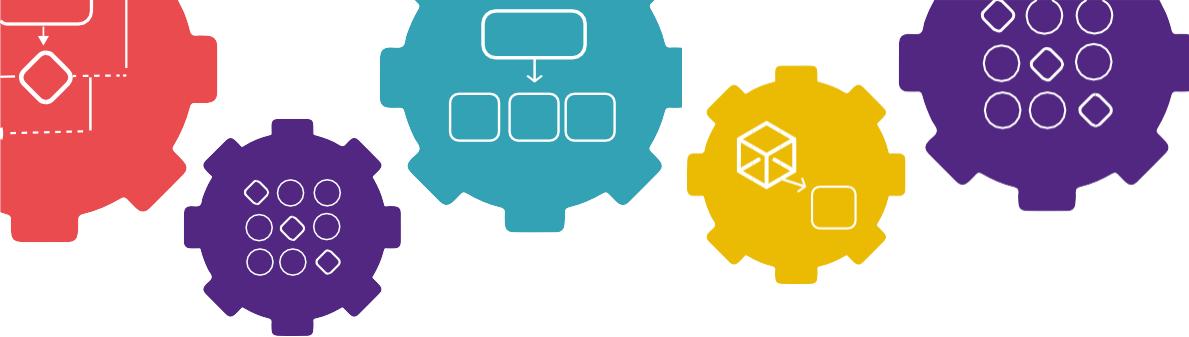
Durante esta actividad se procederá a crear un repositorio para que el estudiante pueda archivar sus documentos para el portafolio del curso.

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

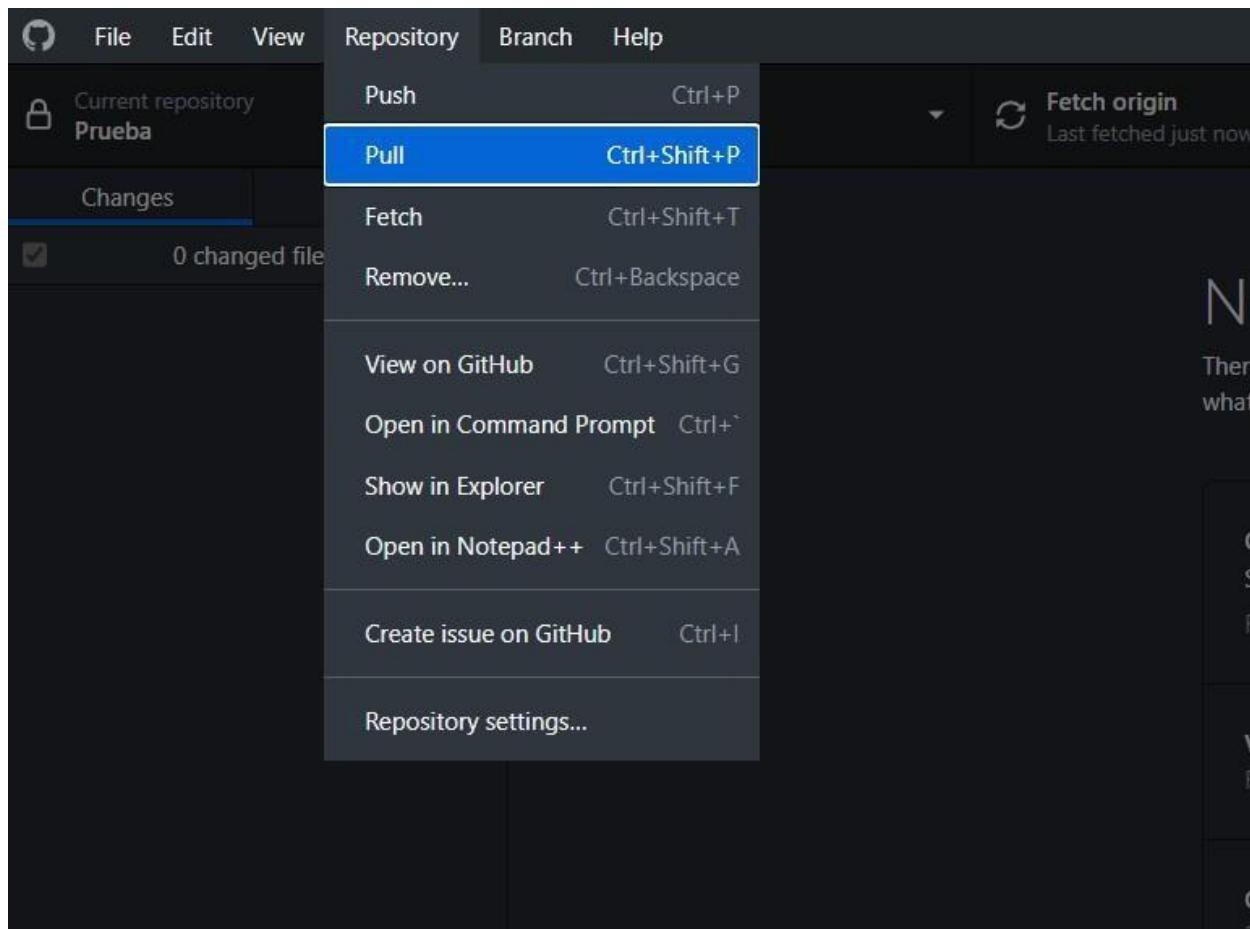


- a. Ingresar nombre del repositorio como: "Laboratorios"
 - b. Marcar opción publico
 - c. Marcar opción "Add README file"
4. Instalar GitHub Desktop
- a. Ingresar a <https://desktop.github.com/>
 - b. Descargar y ejecutar programa
 - c. Iniciar sesión y configurar datos
5. Clonar repositorio

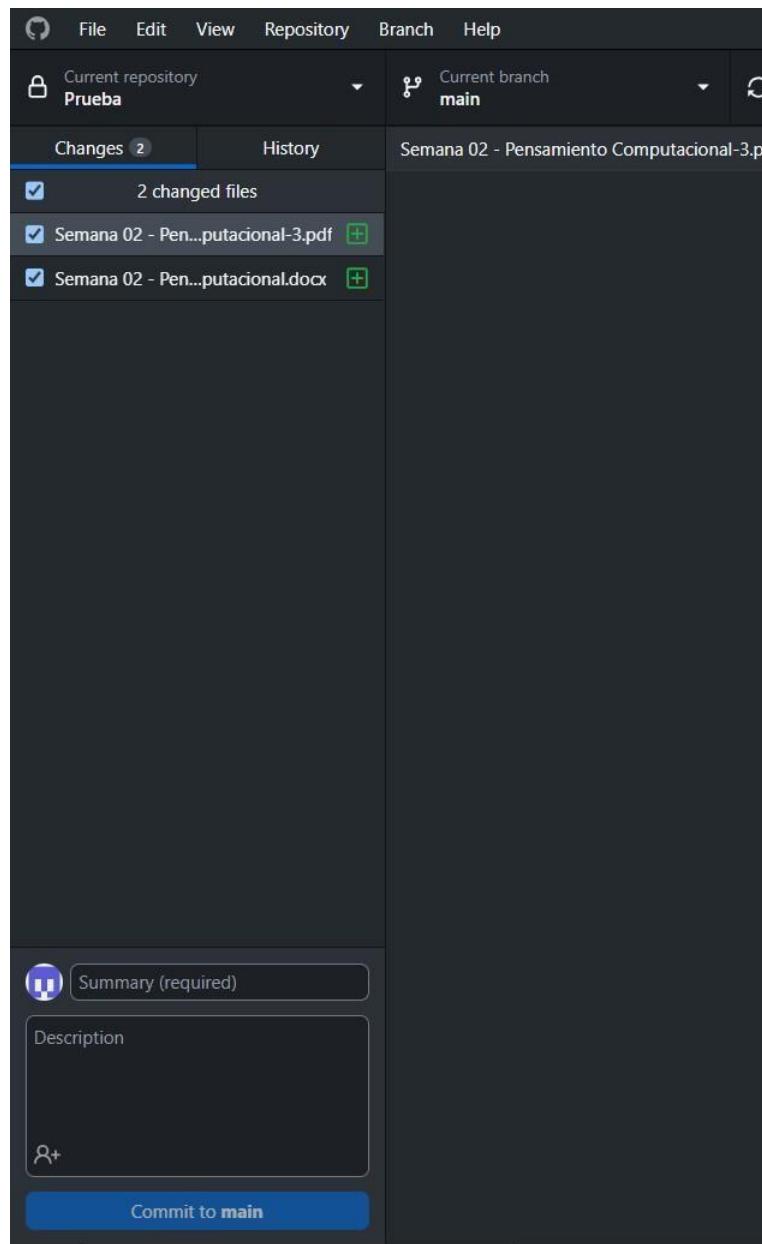
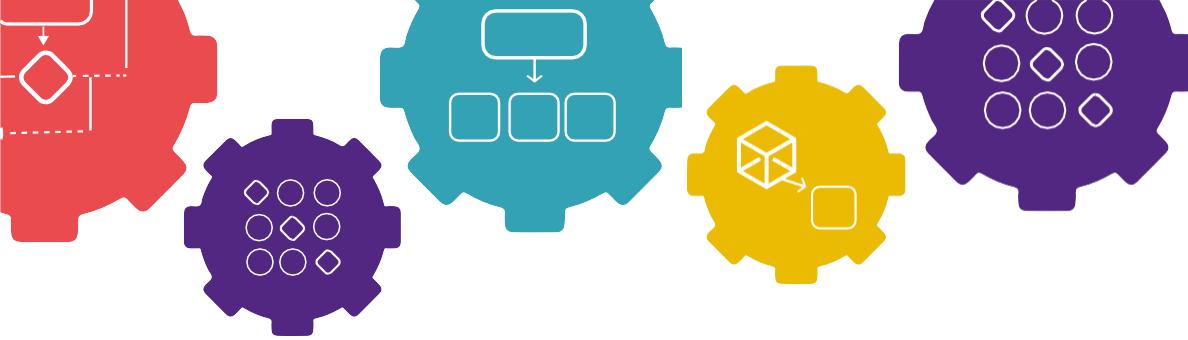




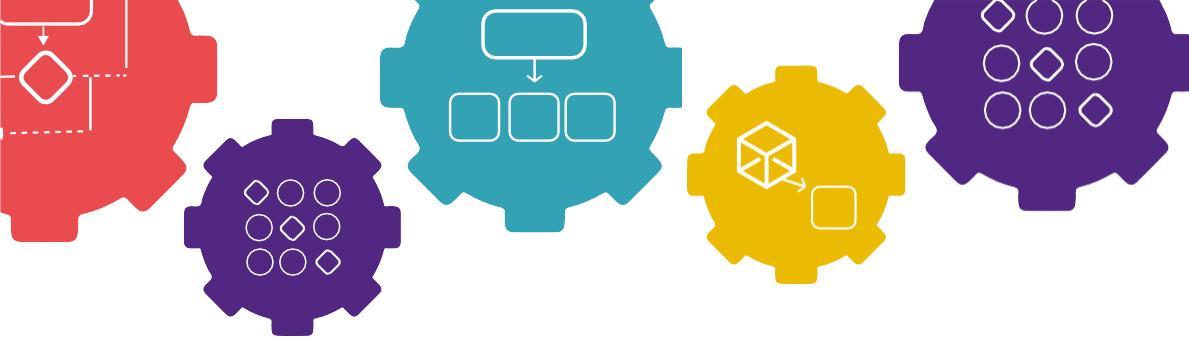
- a. Seleccionar el repositorio creado anteriormente
- b. ¿Cómo se puede verificar que se haya clonado el repositorio? Adjunte una captura de pantalla
6. Utilizar el comando pull para actualizar los archivos locales (Los que se encuentran en el equipo que se está utilizando).



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 página web y/o en la aplicación de escritorio.
- Dentro de los cambios de cada commit aparece información como los archivos editados, el usuario, y fecha realizada. Ejemplo:

Free disk space.

[maryam/free-disk-space \(#6277\)](#)

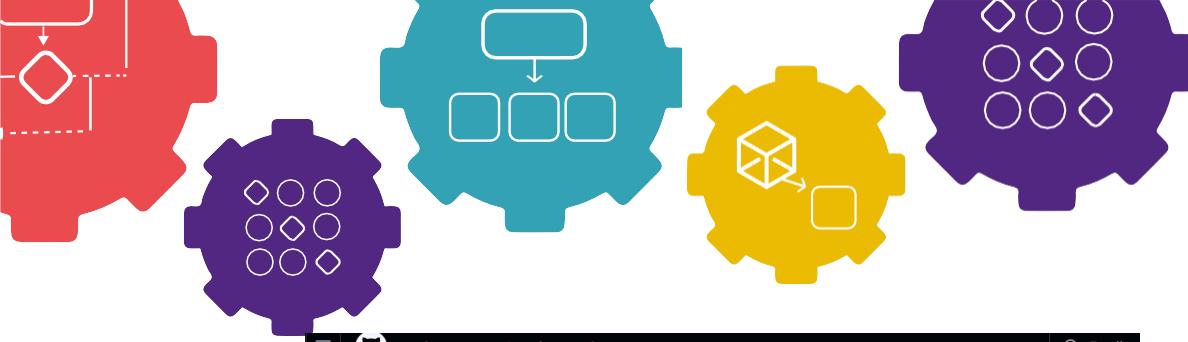
maryamziaa committed 2 weeks ago

diff 6 .github/workflows/colab.yml

Line	Change Type	Text
35	35	@@ -35,6 +35,12 @@ jobs:
36	36	- uses: actions/setup-dotnet@v4
37	37	with:
38	+ 38	dotnet-version: '6.0.x'
39	+ 39	- name: Free disk space
40	+ 40	run:
41	+ 41	sudo rm -rf /opt/ghc
42	+ 42	sudo rm -rf /usr/local/share/boost
43	+ 43	sudo rm -rf "\$AGENT_TOOLSDIRECTORY"
		df -h
38	44	- name: Cache pip
39	45	uses: actions/cache@v4
40	46	with:

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

- Enlace de su usuario de GitHub
- Captura de pantalla de:
 - Repositorio creado



Larissa-M008 / Laboratorio-PenCom

Código Asuntos Solicitudes de extracción Comportamiento Proyectos Wiki

Laboratorio-PenCom Público

principal 1 sucursal 0 etiquetas Agregar archivo

Larissa-M008 Laboratorio 4 673c8ef · hace 4 días

Laboratorio-PC-DLMB Laboratorio 4

LÉAME

Current repository: Laboratorio-PenCom Current branch: main Fetch origin: Last fetched yesterday

Changes History Filter 0 changed files

No local changes. There are no uncommitted changes in your working tree. What do you want to do next?

Open the repository in your external editor Select your editor in Options Repository menu or Ctrl + Shift + O

View the files of your repository in your external editor Repository menu or Ctrl + Shift + E

Open the repository page on GitHub Repository menu or Ctrl + Shift + G

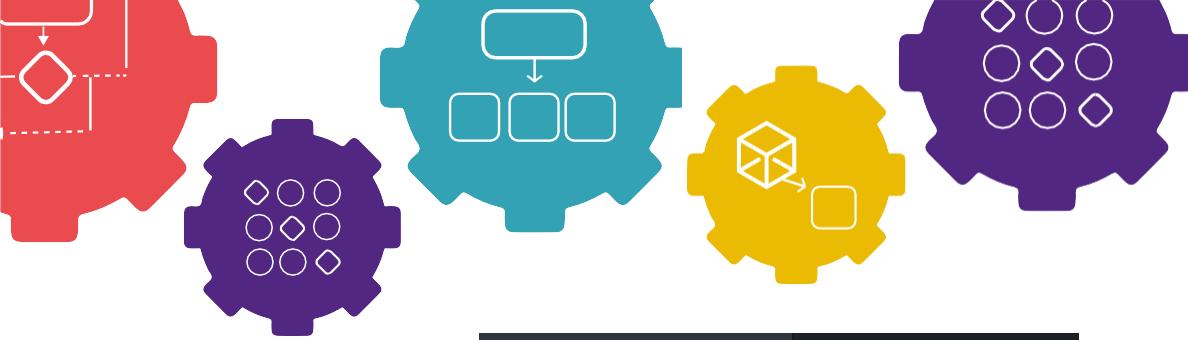
Summary (required)

Description

A+ Commit to main

This screenshot shows a GitHub desktop interface for a repository named "Laboratorio-PenCom". The repository is public and has one branch called "main". The "Changes" tab is selected, showing that there are no local changes. The interface includes a sidebar with sections for "Summary (required)" and "Description", and a bottom bar with a "Commit to main" button.

ii. Archivos locales del repositorio clonado



A screenshot of a GitHub commit history interface. The top bar shows "Current repository Laboratorio-PenCom" and "Current branch main". Below this, a "Changes 10" tab is selected, showing a list of 10 changed files. The files listed are: Lab-PC-DLMB\Lab \Lab _L..\suo, Lab-PC-DLMB\DocumentLayout.json, Lab-PC-DLMB\La..ez_1142526.slnx, Lab-PC-DLMB\Lab 5\App.config, Lab-PC-DLMB\La..._1142526.csproj, Lab-PC-DLMB\Lab...blyAttributes.cs, Lab-PC-DLMB\La...ncesInput.cache, Lab-PC-DLMB\La...Reference.cache, Lab-PC-DLMB\Lab 5\Program.cs, and Lab-PC-DLMB\La...AssemblyInfo.cs. Each file entry includes a green plus sign icon. At the bottom of the list is a "Summary (required)" section with a green plus sign icon and a "Description" input field.

iii. Historial de cambios de un archivo README.



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 que 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

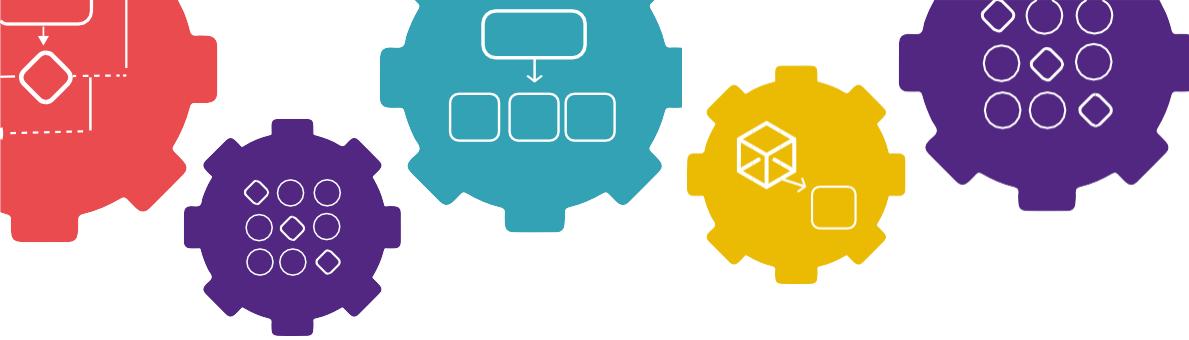
Web y nube (4)

- Desarrollo de ASP.NET y web
Compila aplicaciones web con ASP.NET Core, ASP.NET, HTML/JavaScript y contenedores, e incluye compatibilidad...
- 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...
- Desarrollo de Python
Edición, depuración, desarrollo interactivo y control de código fuente de Python.
- Desarrollo de Node.js
Compile aplicaciones de red escalables con Node.js, un entorno de ejecución JavaScript controlado por eventos...

Móviles y de escritorio (5)

- Desarrollo de la interfaz de usuario de aplicaciones multiplataforma (MAUI)
Cree aplicaciones de Android, iOS, Windows y Mac desde un único código base con C# mediante .NET MAUI.
- Desarrollo de escritorio de .NET
Compila WPF, Windows Forms y aplicaciones de consola mediante C#, Visual Basic y F# con .NET y .NET Frame...
- Desarrollo para el escritorio con C++
Cree aplicaciones modernas de C++ para Windows con...
- Desarrollo de aplicaciones WinUI
Cree aplicaciones para la plataforma Windows con WinUI.

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

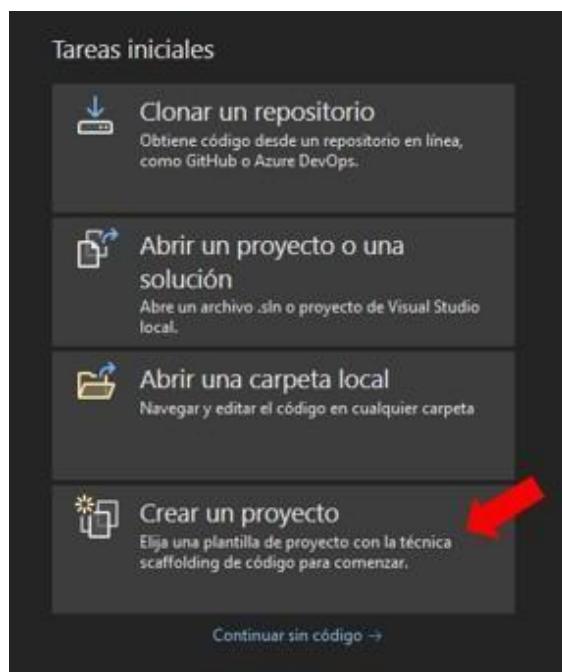


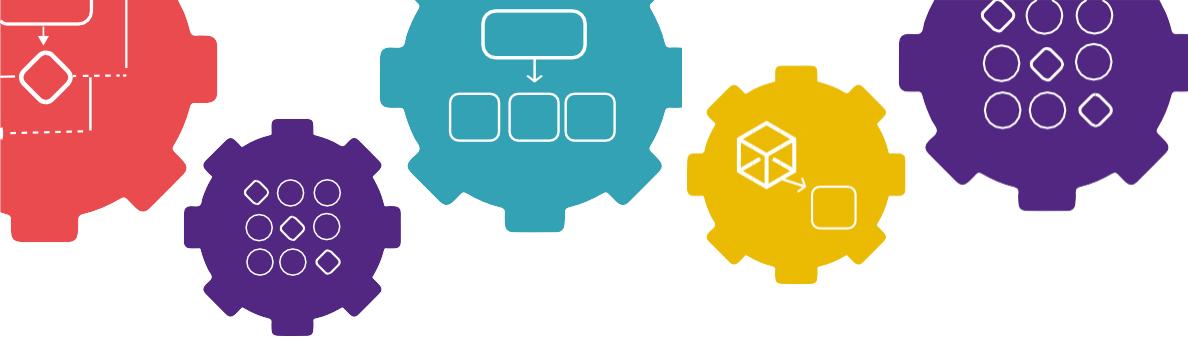
5. Después de haber instalado el programa, abrir Visual Studio, se le solicitará que inicie sesión con una cuenta de correo electrónico de Microsoft (crearla si no se cuenta con una). Este paso es muy importante realizarlo, de lo contrario el software expirará en 30 días.

Creación de un proyecto de consola en lenguaje C#

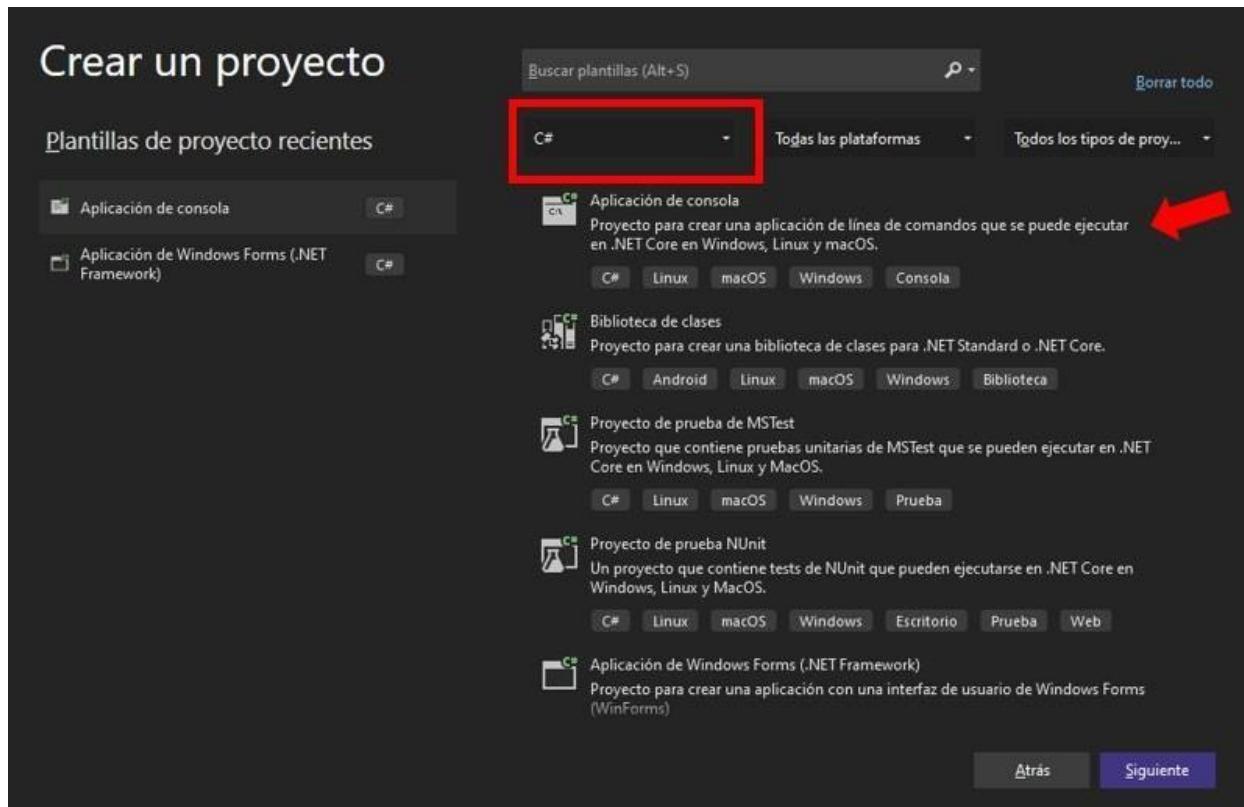
Una vez instalado completada la instalación, seguir estos pasos para crear un proyecto de consola en Visual Studio en el lenguaje de programación C#:

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





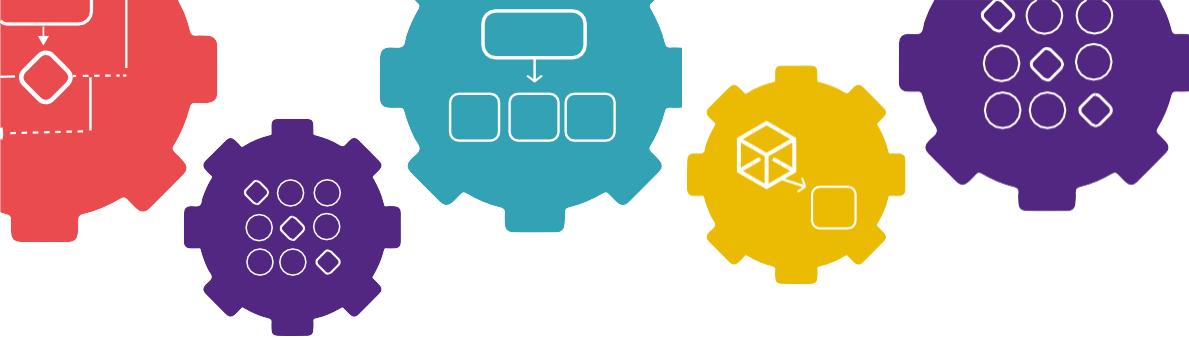
2. **Filtrar Plantilla y seleccionar tipo de proyecto:** Seleccionar C# en el filtro de lenguaje. Elegir la plantilla llamada "Aplicación de consola" (que utiliza .NET) y hacer clic en Siguiente.



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#:

```
Program.cs*  X Salida
C# ConsoleAppPrueba
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**.

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.

Ejemplo de salida esperada:

Lab de Pensamiento Computacional

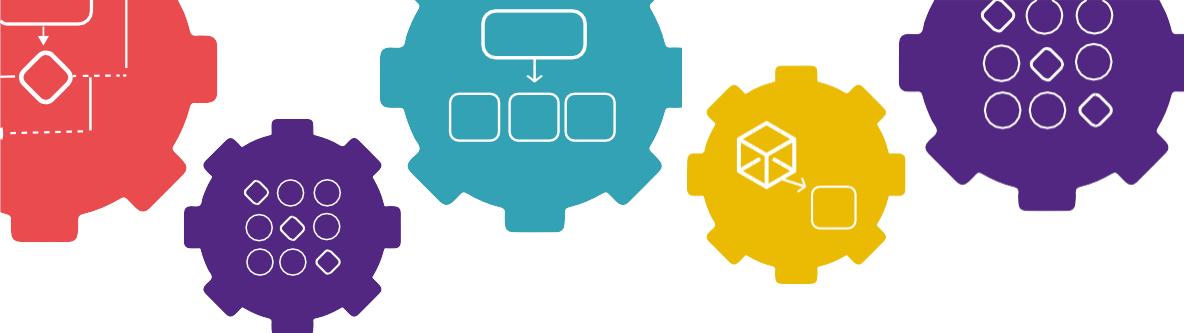
Martin, vas muy bien en el curso Introducción a Programación.

Nunca dejes de practicar.

Hoy es un buen día para no ser administrador de empresas.

Presione una tecla para salir*

EVIDENCIAS DEL TRABAJO:

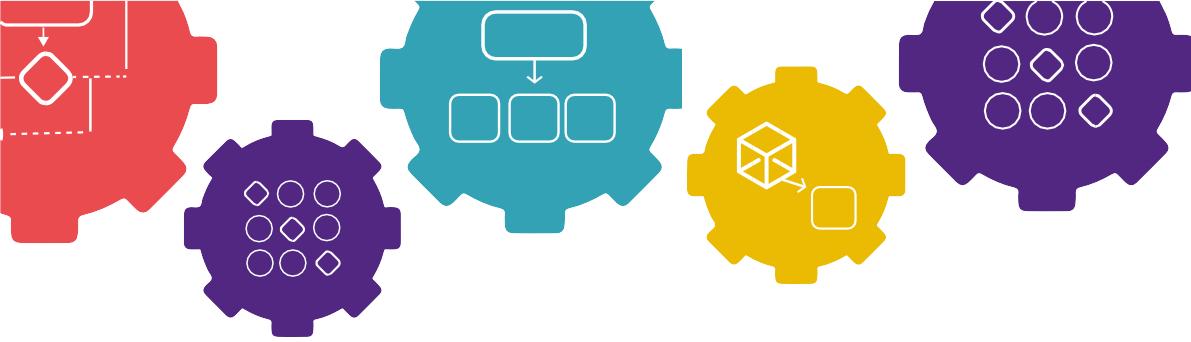


```
Lab 3, Pensamiento Computacional, Larissa Méndez
Ingrese su nombre
Lari
Ingrese el nombre de su curso
Inglés
Hola Lari
eres estudiante del curso de Inglés
Presione cualquier tecla para salir
```

The screenshot shows the Visual Studio IDE interface with the following details:

- Title Bar:** Archivo, Editar, Ver, Git, Proyecto, Compilar, Depurar, Prueba, Herramientas, Extensiones, Ventana, Ayuda, Buscar.
- Toolbar:** Includes icons for file operations, search, and navigation.
- Code Editor:** The file "Program.cs" is open, showing the following C# code:

```
Lab3_A3_Larissa_Méndez
using System;
namespace Lab3_A3_Larissa_Méndez
{
    internal class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Lab 3, Pensamiento Computacional, Larissa Méndez");
            Console.WriteLine("Ingrese su nombre");
            string nombre = Console.ReadLine();
            Console.WriteLine("Ingrese el nombre de su curso");
            string curso = Console.ReadLine();
            Console.WriteLine("Hola " + nombre);
            Console.WriteLine("eres estudiante del curso de " + curso);
            Console.WriteLine("Presione cualquier tecla para salir");
            Console.ReadKey();
        }
    }
}
```
- Status Bar:** Shows "No se encontraron problemas." (No problems found.) and "Línea: 23, Carácter: 6 | SPC | CRLF | UTF-8 with BOM".
- Output Window:** Shows "Salida" (Output) and "Mostrar salida de: Depurar" (Show output from: Debug).
- Right Sidebar:** Includes a "Chat de GitHub Copilot" section with a message from Copilot and a "Copilot" button, along with other AI-related options.



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.