

Taller de Lenguajes I - 2025

Programador Universitario / Licenciatura en Informática / Ingeniería en Informática

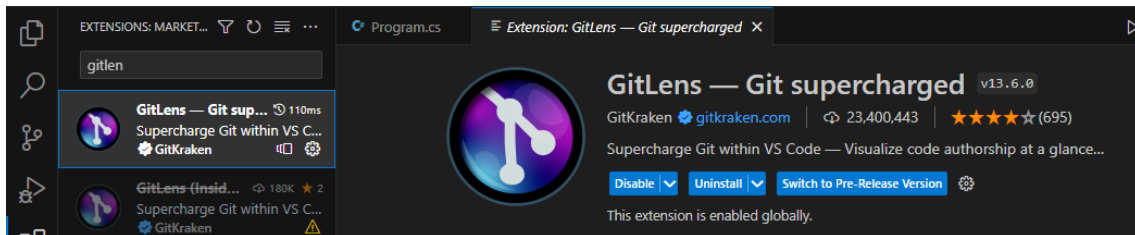
Trabajo Práctico Nro 7

Repositorio.

Crear una carpeta nueva en el disco e inicialice el siguiente repositorio en esa ubicación: <https://tinyurl.com/tl1-2025-tp7>

Mini Guia: Uso de la extensión GitLens

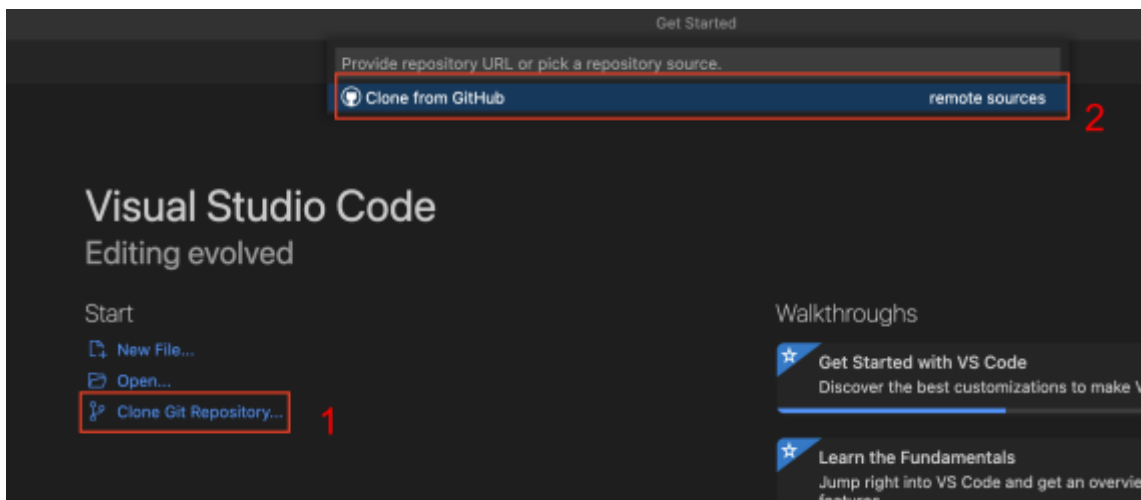
1. Instalar la extensión GitLens



2. Clonar Repositorio

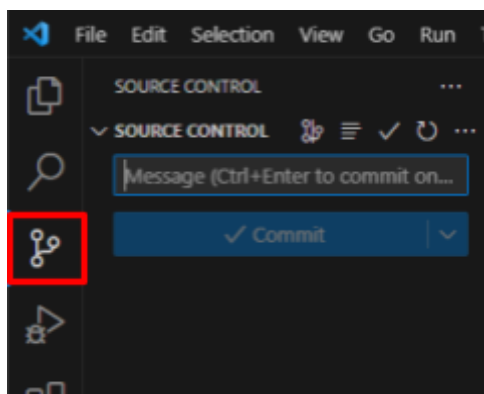
Desde una nueva ventana en el Visual Studio Code elija la opción desde

1. Start → “Clone Git Repository...”
2. se le abrirá una ventana en la parte superior donde tendrá que colocar el link del repositorio que quiera “clonar”



3. Como hacer commit

- Vaya a Source Control en el menú de la izquierda

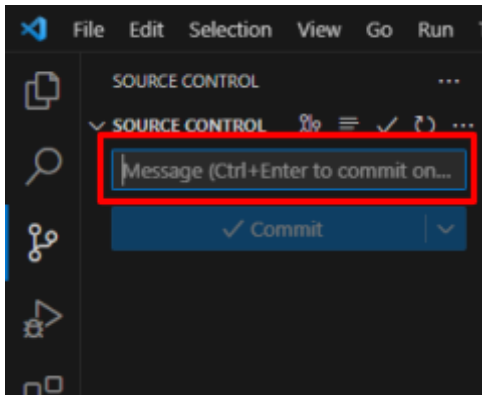


Taller de Lenguajes I - 2025

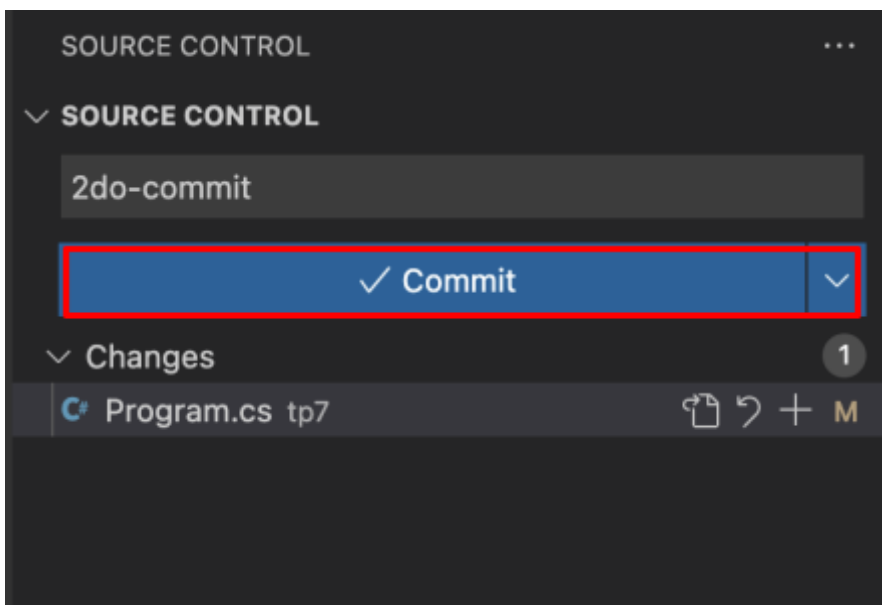
Programador Universitario / Licenciatura en Informática / Ingeniería en Informática

Trabajo Práctico Nro 7

- En la ventana que dice “Message” colocar el mensaje que quiera que tenga el commit.

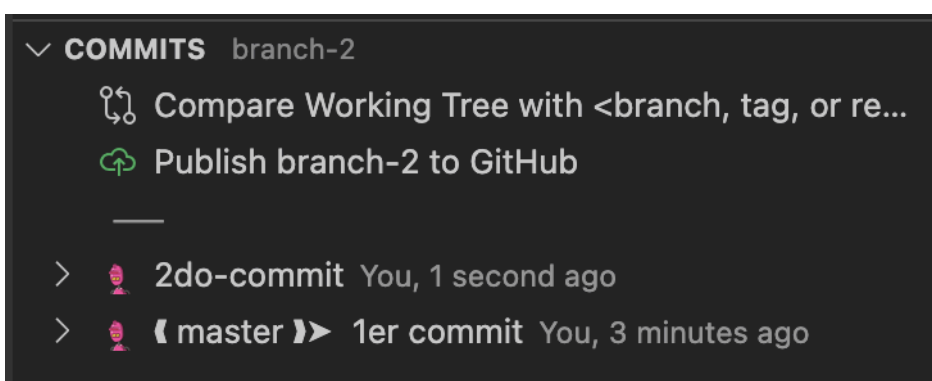


- Presionar el botón “Commit” para realizar el commit en el proyecto (el **add** se hace automáticamente subiendo todo lo que esté en “Changes”).



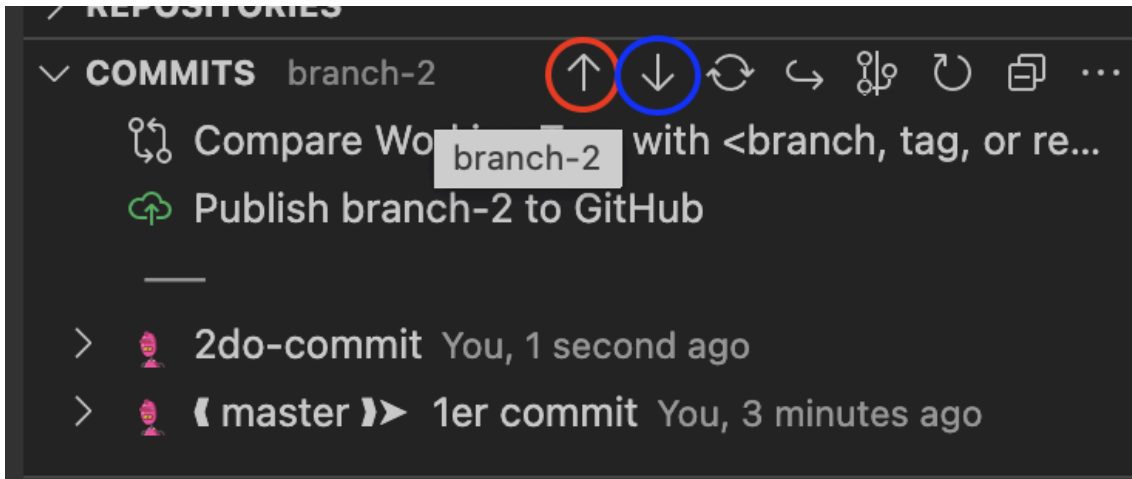
4. Historial de commits

Todos los commits realizados en el branch activo se pueden observar en la sección commits.



5. Push y Pull

Para poder realizar los push y pull en la sección commit hay 2 flechas: una hacia arriba (círculo rojo) para hacer push y una hacia abajo (círculo azul) para hacer pull.



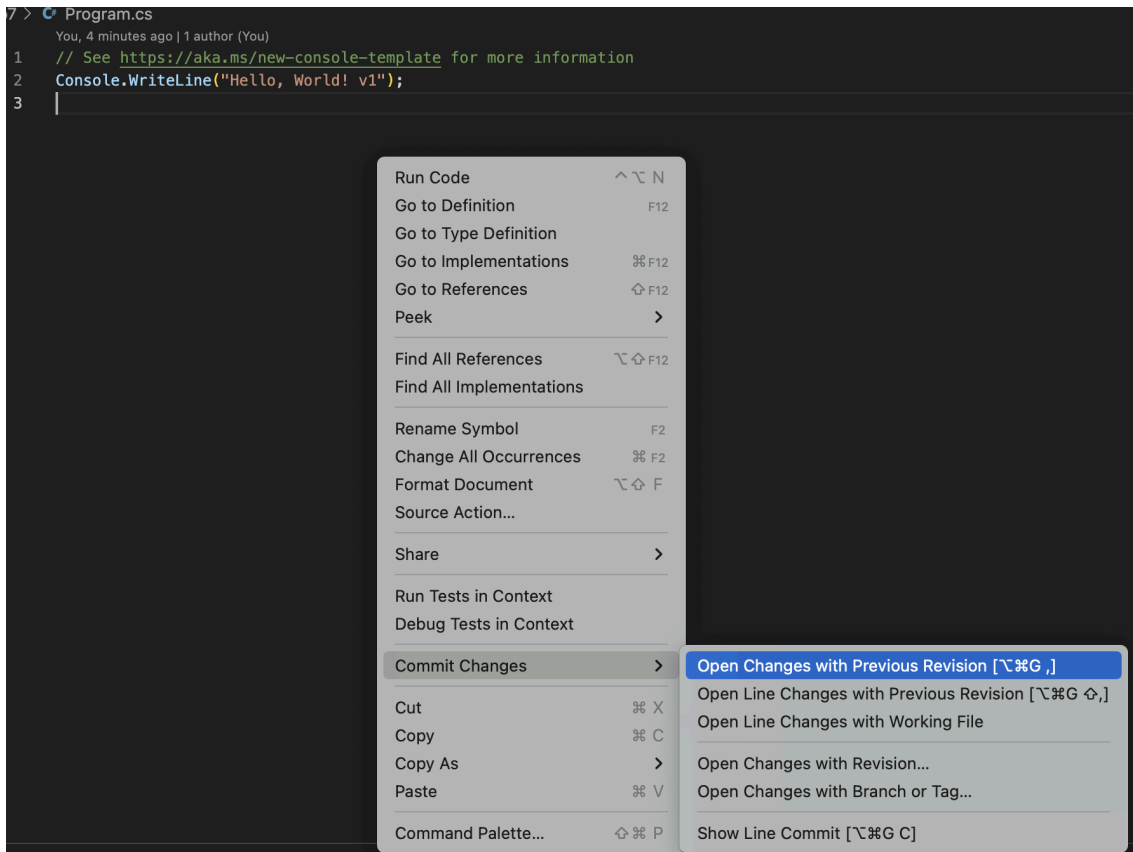
Taller de Lenguajes I - 2025

Programador Universitario / Licenciatura en Informática / Ingeniería en Informática

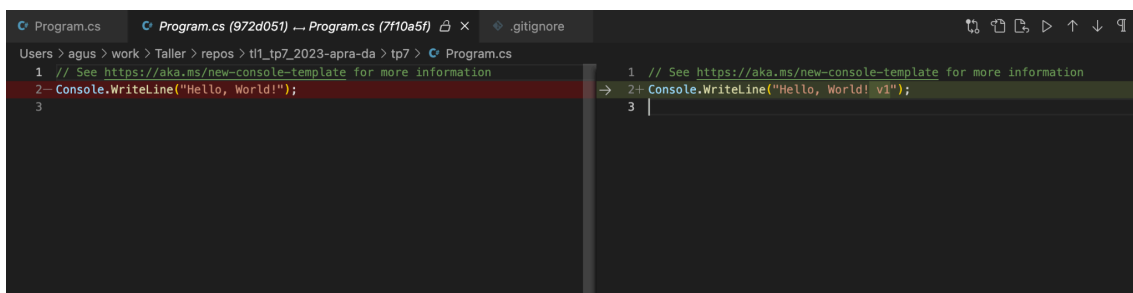
Trabajo Práctico Nro 7

6. Ver cambios en archivo - paso 1

haciendo clic derecho → commit Changes permite ver las diferencias entre la versión actual y revisiones anteriores.



Ver cambios en archivos - paso 2



Antes de comenzar a resolver los ejercicios propuestos no olvide agregar el archivo .gitignore específico para proyectos de CSharp y VisualStudioCode. Para esta tarea puede utilizar el siguiente comando:

- **dotnet new gitignore**

Ejercicio 1

Dentro de su repositorio cree una carpeta llamada "Calculadora" y dentro de ella cree un nuevo proyecto de consola de .NET usando la instrucción **"dotnet new console"** ..

Dentro de este proyecto, crear un nuevo archivo llamado Calculadora.cs y defina un espacio de nombres al comienzo del archivo (**ej. namespace EspacioCalculadora;** en la línea 1 del archivo) . Cree la clase **Calculadora** que permita encadenar operaciones sobre un mismo resultado guardado en un campo llamado dato, utilizando los siguientes métodos.

- void Sumar(double termino)
- void Restar(double termino)
- void Multiplicar(double termino)
- void Dividir(double termino)
- void Limpiar()

Cree también una propiedad llamada `Resultado` para obtener el valor del atributo dato. (Es decir solo defina el **get**).

Para utilizar esta clase desde Program.cs, no olvide incorporar al mismo el espacio de nombres definido en el archivo Calculadora.cs, utilizando la palabra reservada **using**.

Realice una interfaz de usuario para operar la calculadora que permita continuar solicitando operaciones hasta que el usuario ingrese un comando de escape.

Ejercicio 2

Dentro de su repositorio cree una carpeta llamada "SisPersonal", y dentro de ella cree un nuevo proyecto de consola de .NET usando la instrucción **"dotnet new console"** realizaremos un pequeño sistema para administración del personal de una empresa. El mismo tiene que cumplir con los siguientes requisitos:

1. Crear una **clase Empleado** para almacenar la siguiente información:

- a. Nombre (string),
- b. Apellido (string),
- c. Fecha de nacimiento (datetime),
- d. Estado civil (char),
- e. Fecha de ingreso en la empresa (datetime),
- f. Sueldo Básico (double),
- g. Cargo (cargos)

Nota: Cargo es una variable de tipo "cargos" que es enumeración (enum) que ud debería definir con los siguientes elementos: Auxiliar;

Taller de Lenguajes I - 2025

Programador Universitario / Licenciatura en Informática / Ingeniería en Informática

Trabajo Práctico Nro 7

Administrativo; Ingeniero; Especialista; Investigador. (investigue el uso de **enum** en C# para realizar esto)

2. Cree los métodos necesarios para poder obtener los datos que se detallan a continuación:

- a. Calcular lo siguiente:
 - i. La **antigüedad** del empleado en la empresa.
 - ii. La **edad** del empleado,
 - iii. La cantidad de años que le falta al empleado para poder jubilarse, considerando que la edad de jubilación es de 65.
- b. Calcular el **salario**, de acuerdo a la fórmula: **Salario = Sueldo Básico + Adicional**. Para ello el Adicional contempla la antigüedad en años, el cargo y si es casado:
 - i) 1 % del sueldo básico por cada año de antigüedad hasta los 20 años, a partir de este, el porcentaje se fija en 25%.
 - ii) Si el cargo es Ingeniero o Especialista, el Adicional se incrementa en un 50%.
 - iii) Si es casado al Adicional se le aumenta \$150.000.

Por ejemplo, si la antigüedad es de 15 años y el Sueldo Básico = \$650.000, el Adicional es $65.0000 * 0.15 = 97.500$. Si además el cargo es Ingeniero o Especialista, se incrementa el Adicional en un 50%. Esto es: $97.500 * 1.50 = 146.250$. Si a su vez es casado el Adicional será: $146.250 + 150.000 = 296.250$
- c. Cargue los datos para 3 empleados en un arreglo de tipo empleados.
- d. Obtener el Monto Total de lo que se paga en concepto de Salarios.
- e. Muestre por pantalla los datos del empleado que esté más próximo a jubilarse (incluyendo los datos del apartado 2.a y el salario correspondiente).

NOTA: Los puntos **a** y **b** deben estar definidos dentro de la clase como propiedades o métodos según corresponda