# LABORATORIO NO. 1: CONSTRUYENDO EL AMBIENTE DE DESARROLLO (VS CODE + GITHUB + PYTHON)

## Objetivo

Crear un ambiente de trabajo distribuido propio para la pareja de trabajo que le permita a los estudiantes trabajar en los laboratorios y retos del curso.

Al finalizar este laboratorio el estudiante estará en capacidad de:

1. Utilizar VS Code como IDE de desarrollo para Python.
2. Entender los conceptos y aspectos generales de un ambiente de trabajo distribuido (Repositorio GIT, IDE VS Code, Python 3.7+).
3. Administrar las versiones y ramas de código en un repositorio GIT (GitHub).
4. Administrar un depósito de versiones GIT por medio de la consola y línea de comandos.
5. Comprender la organización de una aplicación siguiendo el esquema Modelo–Vista–Controlador (MVC por sus siglas en inglés)

## Fecha Límite de Entrega

Jueves 03 de febrero, 11:59 p.m.

## Trabajo Propuesto

### PASO 1: Instalar VS Code

En el caso particular de este curso se definió utilizar el editor Visual Studio Code (VS Code) como la herramienta de desarrollo.

Se han dispuesto los siguientes recursos del curso para instalar el Ambiente de Desarrollo (IDE) en sus equipos:

- Video Uniandes para instalar y configurar VS Code: https://web.microsoftstream.com/video/ad44bb24-8cc2-4876-81b2-500166128ed8
- Video Uniandes para instalar Python 3.XX en MacOS: https://web.microsoftstream.com/video/206921f5-3939-4a9d-8e19-560cdb31939e
- Video Uniandes para Instalar Python 3.XX en Windows 10: https://web.microsoftstream.com/video/34b2c85e-7498-42ea-a8e8-e4f472f26d21

## PASO 2: Instalar GIT

De igual forma, para el curso se utilizará el Sistema de Control de Versiones Git y el administrador de repositorios GitHub.

Se han dispuesto los siguientes recursos para instalar GIT en sus equipos:

- Video Uniandes para registrar GitHub e instalar GIT en Windows 10: https://web.microsoftstream.com/video/656bd7fe-9115-479d-b410-4a076e1ac974
- Video Uniandes para registrar GitHub e instalar GIT en MacOS: https://web.microsoftstream.com/video/e790876c-7d7e-4738-8370-000b4a64cd2f

## PASO 3: Utilizar GitHub

GitHub es una plataforma en la nube que permite administrar un proyecto de software, bien sea que se trabaje de forma individual o en grupo. En este curso, publicaremos los laboratorios y esqueletos de proyectos por medio de esta plataforma.

Para continuar con el laboratorio deben tener una cuenta en GitHub, en caso de no poseerla deben **registrarse con su correo de Uniandes** en la plataforma utilizando el enlace https://github.com/join. Si ya tienen una cuenta personal con otro correo, **debe crear una nueva cuenta con el correo de la Universidad o cambiar el correo de registro de que ya existe.**

GitHub permite crear una organización, el objetivo de la organización es agrupar depósitos de software relacionados entre sí. Así que, una vez creen sus usuarios en GitHub, deben crear una **organización** para trabajar con su compañero y poder agregar al profesor y a los monitores de su sección.

Para nombrar su organización siga el siguiente formato: ***EDA2022-1-SEC<<XX>>-G<<YY>>,*** donde: ***<<XX>>*** es el número de la sección y ***<<YY>>*** el numero asignado del grupo (Ej.: ***EDA2022-1-SEC02-G01*** para el primer grupo de la sección 2 del curso).

Uno de los integrantes del grupo debe crear esta organización, para ello siga se ha dispuesto el siguiente recurso:

- Video Uniandes para crear un equipo en GitHub, URL: https://web.microsoftstream.com/video/0ba7d21e-d515-4fdb-bb68-d4abbb1e3582

Por último, recuerde que el creador de la organización debe invitar a los demás miembros (compañero, profesor y monitores).

## PASO 4: Copiar un proyecto en la organización

Un proyecto de software puede iniciarse con una plantilla en blanco o con un proyecto prexistente. Se denomina **Fork** el copiar un proyecto previo para trabajarlo de manera independiente al original.

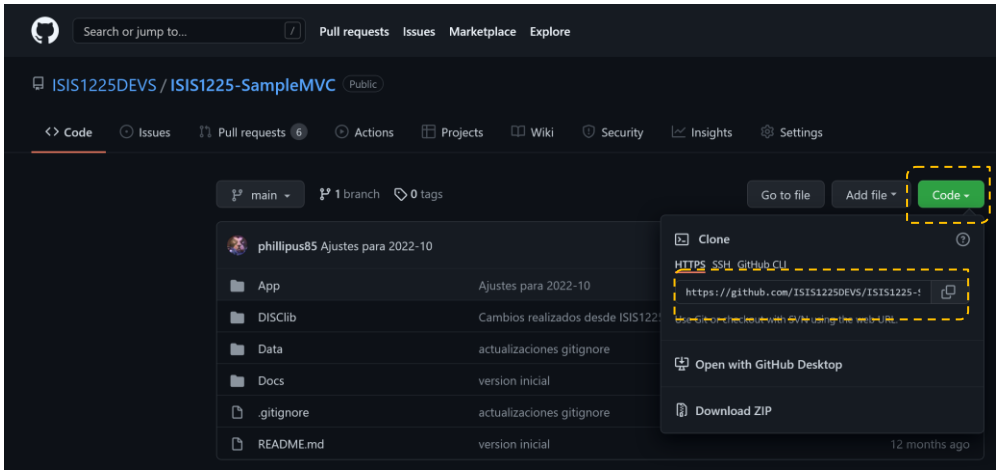
Este curso tiene una cuenta GitHub asociada cuyo enlace es https://github.com/ISIS1225DEVS/, en donde se encuentran todos los esqueletos para los retos y laboratorios de la clase.

En particular, el enlace disponible para esta práctica es: https://github.com/ISIS1225DEVS/ISIS1225-SampleMVC.git

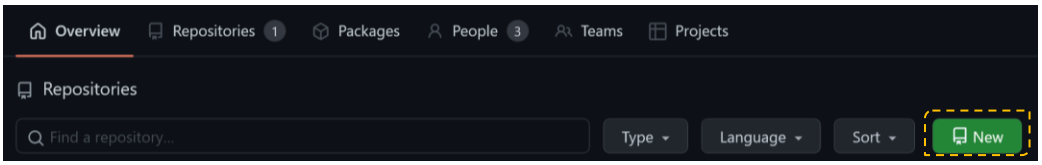
Normalmente un Fork desde un repositorio público debe ser público y el proceso para lograrlo es automatizado y soportado por GitHub, si lo desea puede ver el siguiente video para conocer el procedimiento.

- Video Uniandes para Copiar/Fork repositorios GitHub público: https://web.microsoftstream.com/video/9be3bd7c-26a8-4227-a0a9-e0d7043e0322

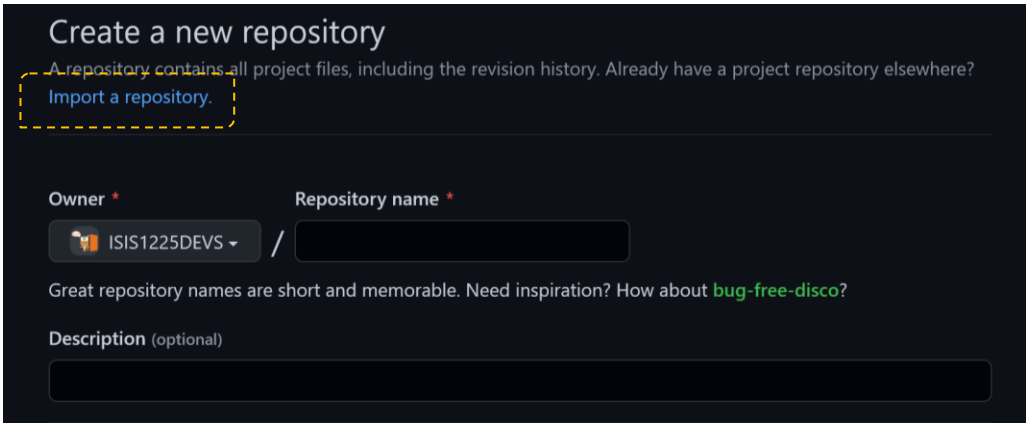
Sin embargo, para crear un **Fork privado** a partir de un repositorio **público** se necesita completar un proceso **manual**. Para ello diríjase al enlace de practica y seleccione el botón verde de **Code/Código** en la página del repositorio, copie el URL que aparece en la pestaña como se muestra a continuación.




Ahora, vuelva a su organización y en la página principal y seleccione crear nuevo repositorio con el botón verde de **New/Nuevo** como se muestra en la siguiente figura.



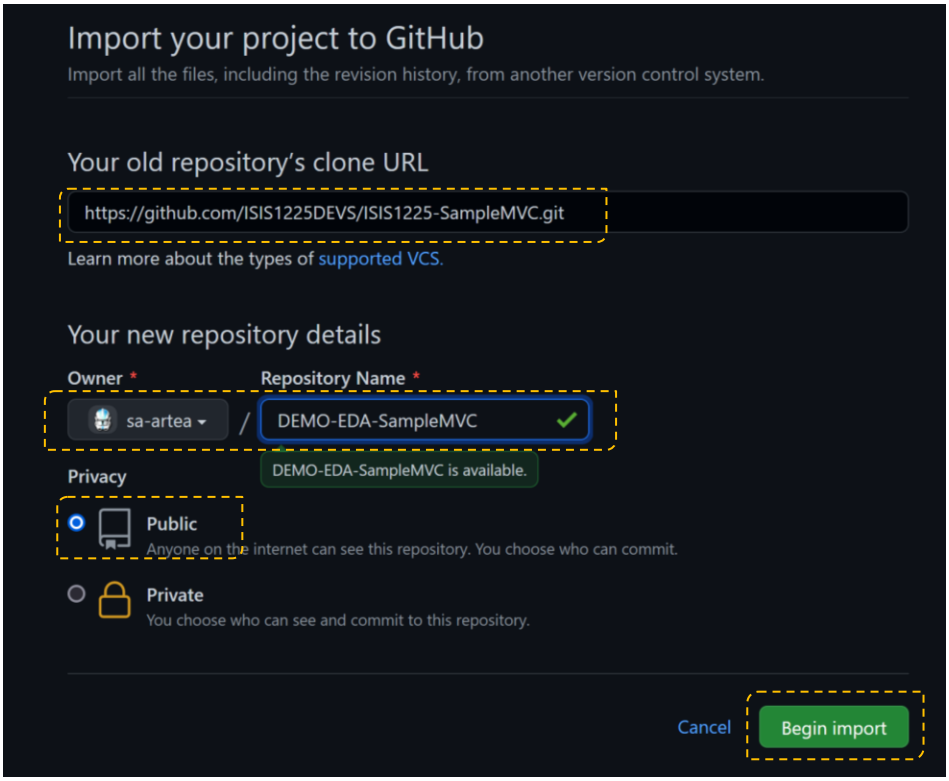

Esto lo guiará a una página donde podrá crear un nuevo repositorio independiente basado en proyecto existente seleccionando la opción **Import a repository** como se muestra en la figura.




A continuación, complete los siguientes campos de configuración indicados por GitHub:

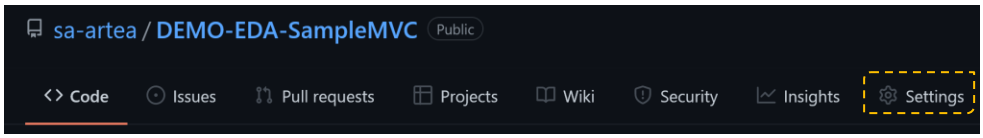
1) El URL del repositorio de origen, que en este caso el enlace de práctica.
2) El propietario del repositorio a crear, que en este caso el nombre de la organización.
3) El nombre del repositorio de destino, que en este caso será **DEMO-EDA-SampleMVC.**
4) La **Privacy/Privacidad** del repositorio de destino, que por el momento será **pública.**

Al completar los campos de configuración inicie el proceso presionando el botón de **Begin Import** como se indica en la siguiente imagen.

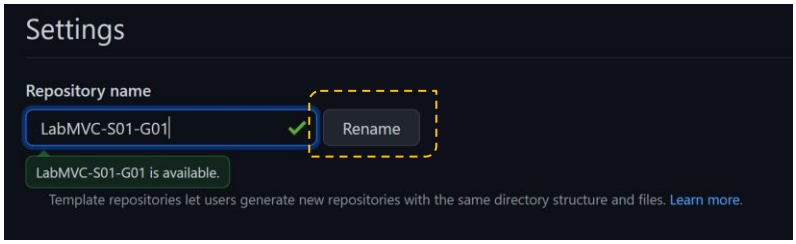



Antes de clonar el repositorio en su computador diríjase a su organización (Ej.: ***EDA2022-1-SEC02-G01*** para el primer grupo de la sección 2 del curso) y cambie el nombre del repositorio.

Renombrar el repositorio se hace ingresando al enlace de ***Ajustes/ Settings*** del repositorio como se ve a continuación:




Al ingresar, diríjase al campo donde está el nombre de su repositorio y cámbielo siguiendo el siguiente esquema: **LabMVC-S<<XX>>-G<<YY>>** donde **XX** es el numero de la semana de la práctica que se está haciendo y **YY** el número del grupo de trabajo, un ejemplo de esto es **LabMVC-S01-G01** para este **segundo laboratorio** hecho por el **grupo 1** Como se ve en la imagen.

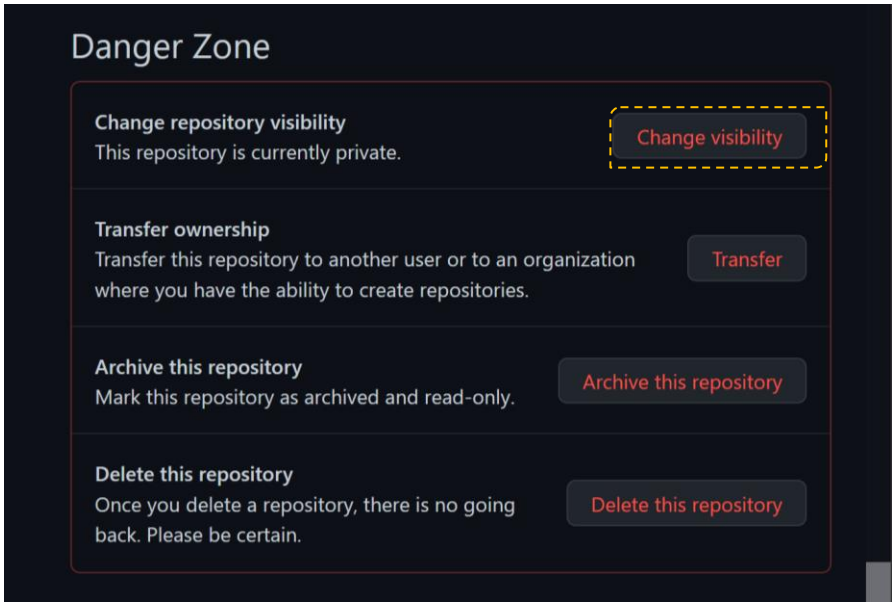


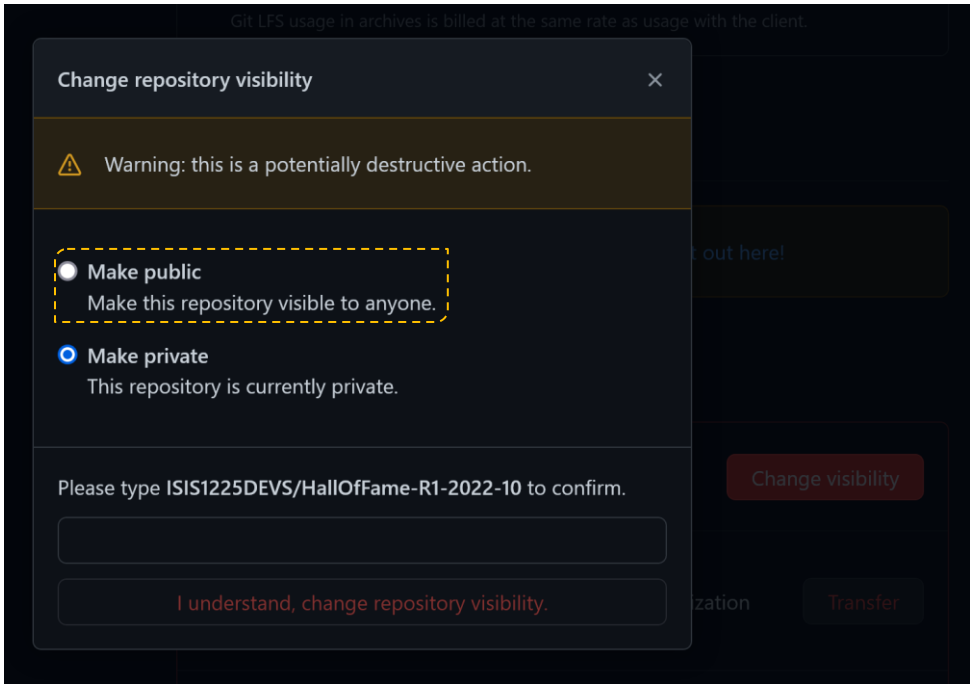

**NO necesita** agregar la sección o el semestre en este nombre porque ya está identificado en su organización.

## PASO 5: Cambiar la visibilidad del repositorio

Ahora, al final de la misma página de ***Ajustes*** (***Settings)*** del repositorio asegúrese de que la ***Visibilidad (Visibility)*** de este sea ***Privada (Private).***

Para ello, seleccione la opción para ***Cambiar visibilidad (Change visibility)*** en donde subsecuentemente deberá seleccionar la opción de **Hacer privado (Make private) el repositorio** como se puede ver en las siguientes imágenes.




**IMPORTANTE:** *Todos los repositorios de trabajo durante el semestre deben ser privados, ya que este es un trabajo propio e intransferible de las parejas.*

## PASO 6: Descargar el proyecto

GitHub guarda en la nube el código que vamos desarrollando para no perder el trabajo, permite que todos los desarrolladores tengan acceso al código y maneja una bitácora de cambios (versiones).

Sin embargo, para poder desarrollar software necesitamos tener el proyecto en nuestra máquina personal, no en la nube. Para esto debemos copiar el proyecto que se encuentra en GitHub a un espacio de trabajo adecuado en nuestro computador.

A esta operación la llamamos **clonar (clone)** un proyecto. Y el resultado es que la máquina de cada miembro del grupo posee una copia del proyecto para trabajar de forma independiente. Esta copia estará conectada al proyecto en GitHub, de forma que siempre puedan saber qué cambios se han realizado.

Para aprender a clonar un proyecto, sigan las instrucciones del siguiente video, utilizando el enlace del repositorio privado creado a partir del **ISSI1225-SampleMVC**.

- Video Uniandes Como clonar un proyecto desde Visual Studio Code: https://web.microsoftstream.com/video/8f28c48f-5acf-4ef7-961c-eebb30ea3fdf

## PASO 7: Integrar VS Code y GitHub

Ahora que tenemos instalado Git, VS Code y nuestra cuenta en GitHub, se pueden utilizar muchas formas de trabajo para integrar las tres herramientas. Una de las formas más sencillas es conectar **VS Code** con **GitHub** mediante un plugin.

Para integrar estas herramientas se ha dispuesto del siguiente recurso del curso:

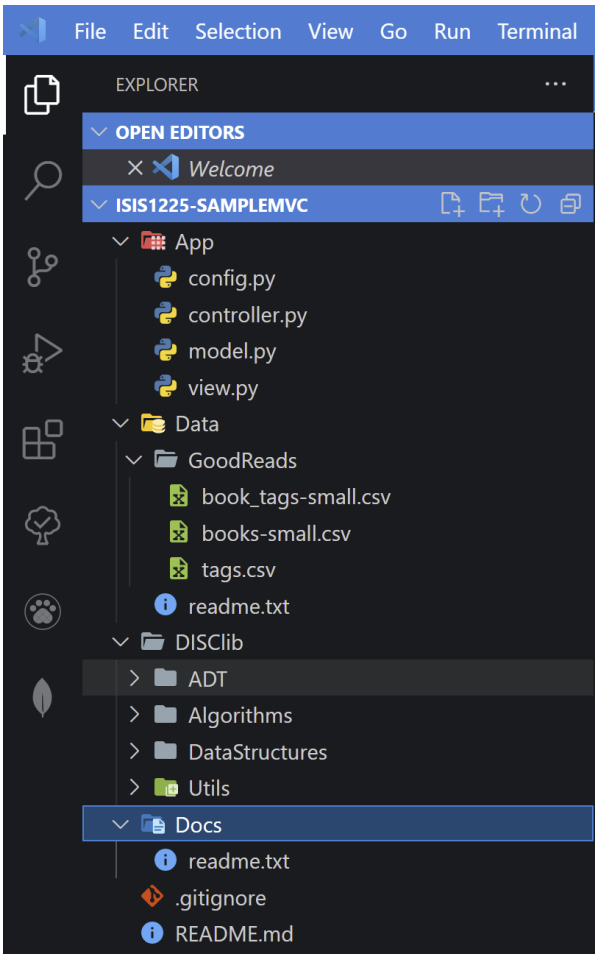
- Video Uniandes Instrucciones generales para instalar Plug-In GitHub + VS Code: https://web.microsoftstream.com/video/e8a65263-aa16-4dd3-8291-f53a53258a84

**IMPORTANTE:** Para mejorar el manejo de GIT, VS Code, GitHub y todas sus extensiones recomendamos ver los recursos complementarios externos a Uniandes recopilados en el contenido **Pills EDA** de nuestra sección unificada.

## PASO 8: Examinar el proyecto en VS Code

El proyecto **Sample-MVC** busca familiarizarlos con el editor VS Code y mostrar algunos elementos que se usarán durante el curso.

Una vez clonado el proyecto, vamos a examinarlo en VS Code para entender su estructura como se observa en la siguiente imagen




Examine el directorio llamado ***App***, en este directorio encontrará tres archivos. Uno llamado ***view.py,*** otro llamado ***controller.py*** y otro llamado ***model.py.*** Este esquema de trabajo busca organizar claramente el código en **tres** distintas partes, llamadas **Modelo, Vista y Controlador**. Este modelo es uno de los más conocidos y utilizados en desarrollo de aplicaciones de todo tipo. Muchas de las aplicaciones Web y móviles que utilizan en su celular, usan este esquema de organización.

**La vista** ***(view.py)*** contiene todo lo relacionado con la interacción con el usuario, captura y presenta la información al usuario. Se usa para facilitar actualizaciones y cambios la vista y no tiene ninguna mención a las estructuras de datos.

**El modelo** ***(model.py)*** contiene los *datos* en memoria principal. Aquí se implementan todas las operaciones, como las búsquedas y las soluciones a los requerimientos de los retos. Es importante que el modelo no conozca cómo se presentan los datos al usuario. Por lo tanto, no tiene ningún contacto con la vista *(view.py).*

Finalmente, para comunicar la vista y el modelo permitiendo su independencia se utiliza un archivo llamado **controlador** ***(controller.py)***. Al revisar el ejemplo del laboratorio, se dará cuenta que la única función del archivo *view.py* es interactuar con el usuario y comunicar todas las operaciones del usuario al controlador.

Al revisar el archivo ***controller.py*** encontrará que se encarga de leer los archivos de datos e invoca las operaciones del modelo ***(model.py)***. Además, recibe las respuestas de la operación del modelo y las envía a la vista***(view.py)***.

Finalmente, revise cuidadosamente el código del archivo ***model.py.*** este archivo contiene los modelos usados para solucionar los requerimientos. Es decir que en el ***model.py*** están todos los datos que usaremos en nuestro programa.

Adicionalmente, el proyecto tiene un directorio llamado ***Data,*** el cuál inicialmente se encuentra vacío. Descargue el archivo del caso de estudio ***(GoodReads)*** ubicado en la carpeta de **datos de entrenamiento** de la sección **unificada** del curso. Descomprima el archivo **.zip* y copie los archivos **.CSV* en el directorio ****/Data/GoodReads*** del proyecto.

El proyecto maneja tres archivos:

- ***Books.csv:*** Este archivo contiene toda la información de los libros, tales como autor, ISBN, fecha de publicación e idioma entre otros.
- ***Tags.csv:*** Este archivo contiene todas etiquetas que se pueden poner a un libro, como por ejemplo el tipo de obra, es decir si es de ficción, ciencia o suspenso.
- ***Book_tags.csv:*** Este archivo tiene el identificador de un libro y el identificador de un tag, para indicar que ese libro fue etiquetado con ese tag por un usuario.

Ejecute el proyecto desde VS Code y cargue la información de los libros y los tags.

## PASO 9: Modificar el proyecto

Dado que el proyecto lee dos de los tres archivos del ejemplo, el de libros *(Books.csv)* y el de tags *(Tags.csv)*, se requiere que ustedes modifiquen el proyecto para que en el menú de opciones aparezca una tercera opción para cargar el archivo llamado *book_tags.csv*.

Busque el comentario ***"# TODO: Modificaciones para el laboratorio 1"*** dentro del proyecto, este le indicara el lugar más propicio para modificar el proyecto.

En el próximo laboratorio trabajaremos sobre dicha opción en un reposito nuevo.

## PASO 10: Actualizar los cambios

Desde VS Code, haga un **Commit** de los cambios y publíquelos en su depósito en GitHub. Para poder realizar esta tarea siga las instrucciones que se brindan en el siguiente recurso:

- Video Uniandes - Como administrar desarrollo de código desde VS Code: https://web.microsoftstream.com/video/8dd18c13-b318-4ef2-b263-5037121ac090

## PASO 11: Compartir resultados con los evaluadores

Finalmente, para realizar la entrega de los resultados de la práctica de laboratorio se debe enviar el enlace (URL) del repositorio GitHub a los monitores y profesores de su sección. Para ello, siga las siguientes indicaciones:

1. Invitar al profesor de laboratorio y los monitores de su sección a la organización del grupo.
2. Incluir en el **README** del repositorio los datos completos de los integrantes del grupo (nombre completo, correo Uniandes y código de estudiante).
3. Generar una versión en GitHub con el comentario "Entrega Final – laboratorio 1" *(git commit -m "Entrega Final – laboratorio 1")* antes de la fecha límite de entrega.
4. Enviar el enlace (URL) del repositorio por BrightSpace antes de la fecha límite de entrega.

Recuerden que cualquier documento solicitado durante en la práctica debe incluirse dentro del repositorio GIT y que solo se calificara hasta el último **COMMIT** realizado antes de la media noche (11:59 PM) del jueves 04 de febrero de 2022.