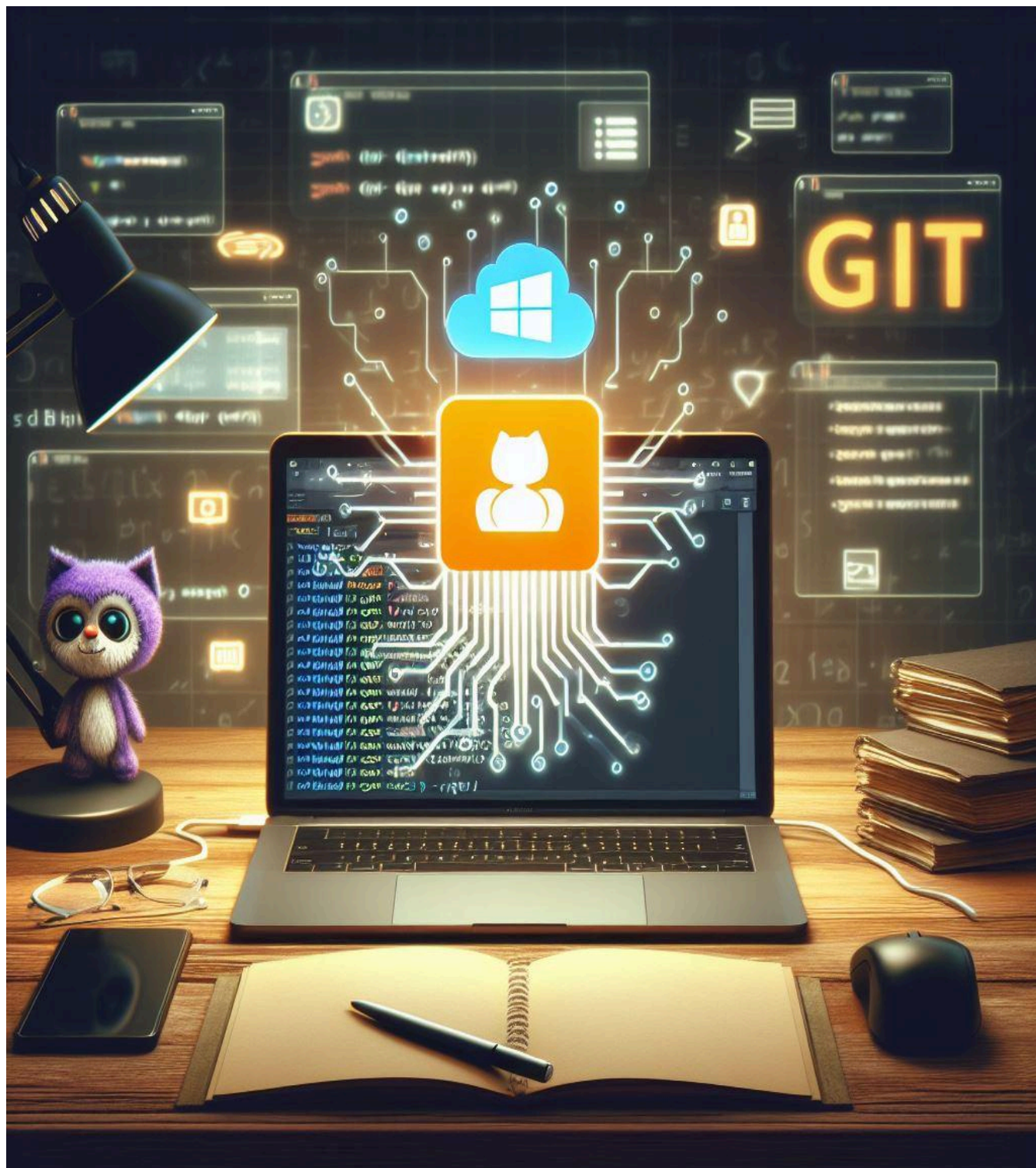


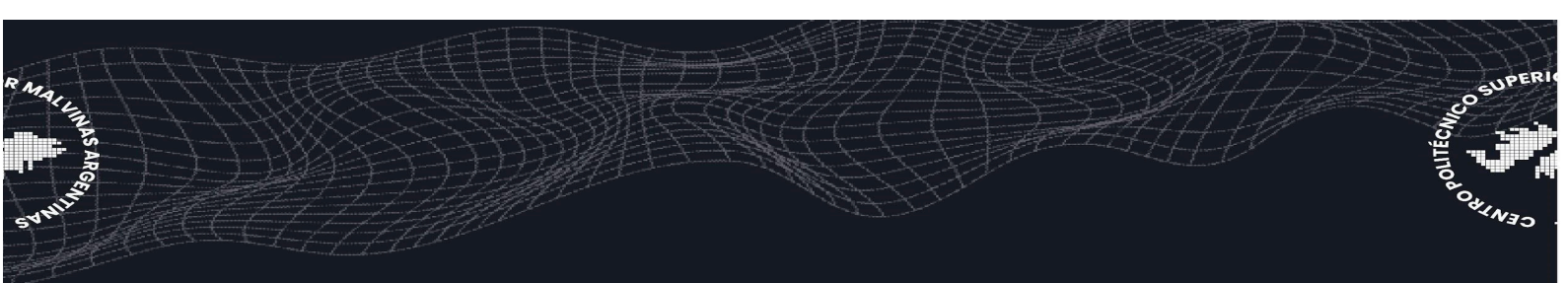


Nombre del bloque PROGRAMACIÓN II

Año de cursada 1° AÑO

Clase N° 2: Introducción a Git y GitHub con Integración en VSCode





Contenido:

En la clase de hoy trabajaremos los siguientes temas:

- Que es Git y cómo funciona
- Que es GitHub
- Instalación de Git en Windows
- Cómo crear una cuenta de GitHub
- Integración nativa de Git en Visual Studio Code.
- Conectar VsCode con Git y GitHub

Presentación:

Bienvenidos a la segunda clase de este trayecto formativo denominado Programación II 🖐️. Estoy emocionado de continuar este viaje con ustedes, explorando el fascinante mundo de la programación.

Transformación Digital y Control de Versiones

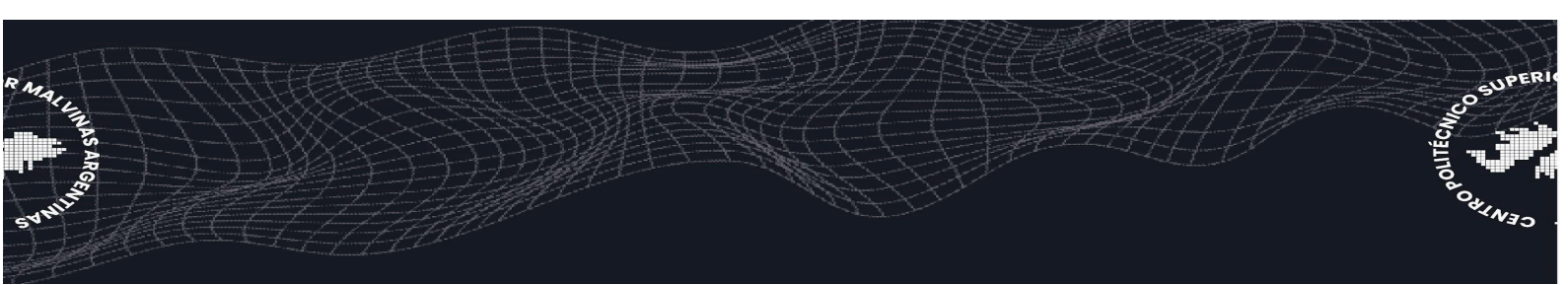
En la primera clase, presentamos cómo la tecnología y la programación están transformando nuestro mundo. Hoy, nos sumergiremos en una herramienta fundamental para cualquier desarrollador: el control de versiones.

¿Qué es el Control de Versiones y por qué es importante?

El control de versiones es un sistema que registra los cambios realizados en un conjunto de archivos a lo largo del tiempo. Esto es vital en el desarrollo de software, ya que permite a los equipos de trabajo coordinar sus esfuerzos, rastrear cambios, revertir a versiones anteriores y trabajar de manera colaborativa de manera más eficiente.

Objetivos de Aprendizaje:

- Comprender los conceptos fundamentales de Git y su importancia en el



desarrollo de software.

- Familiarizarse con la plataforma GitHub y su papel en la colaboración en proyectos.
- Dominar la integración de Git en Visual Studio Code para una experiencia de desarrollo más fluida y eficiente.

Les deseo a todos y todas que tengan una excelente cursada, estoy a disposición para facilitarles lo que necesitan y sin más preámbulo los invito a recorrer la primera clase. Exitos!! 😊

Desarrollo y Actividades:


Herramientas de trabajo:

A continuación vamos a listar las herramientas que vamos a utilizar para llevar adelante nuestra cursada:

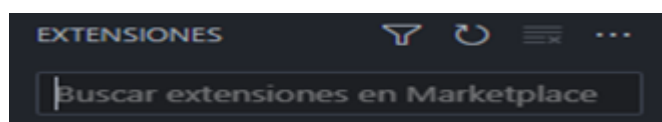
- Python 
- VsCode 
- Git  
- GitHub 

Extensiones en VsCode:

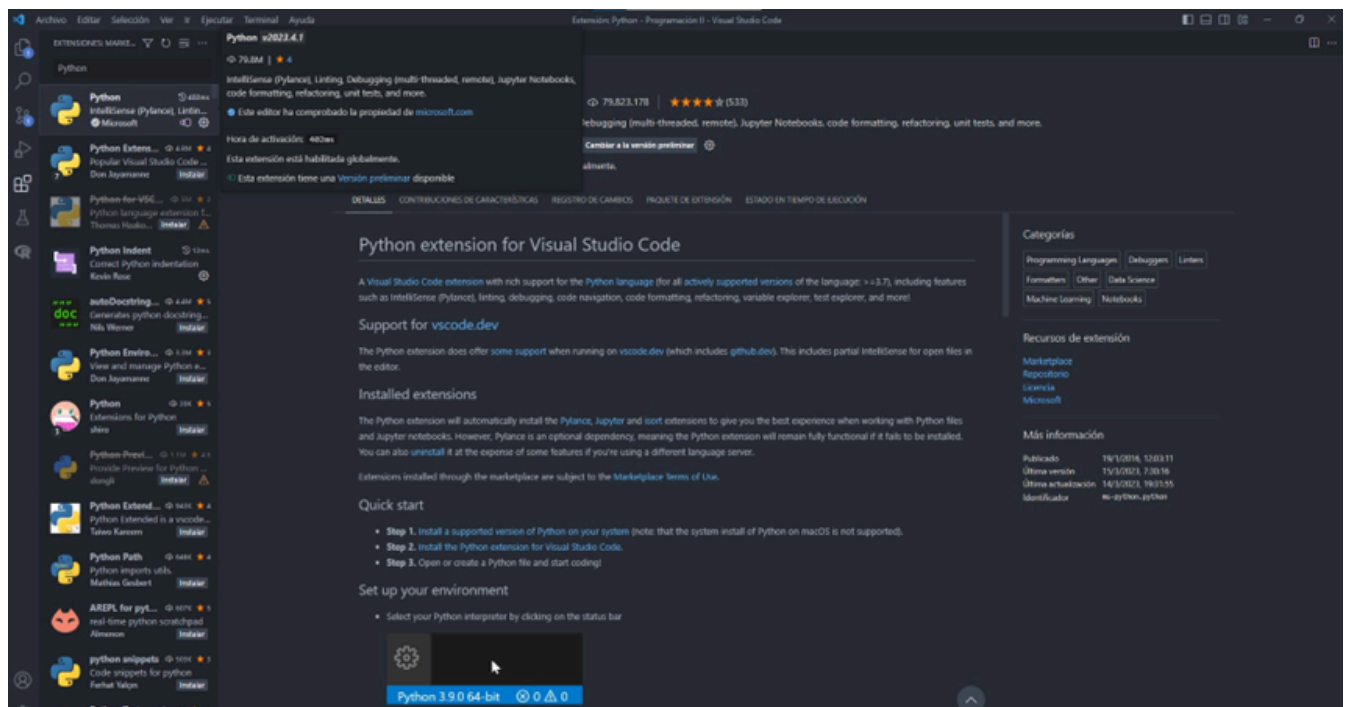
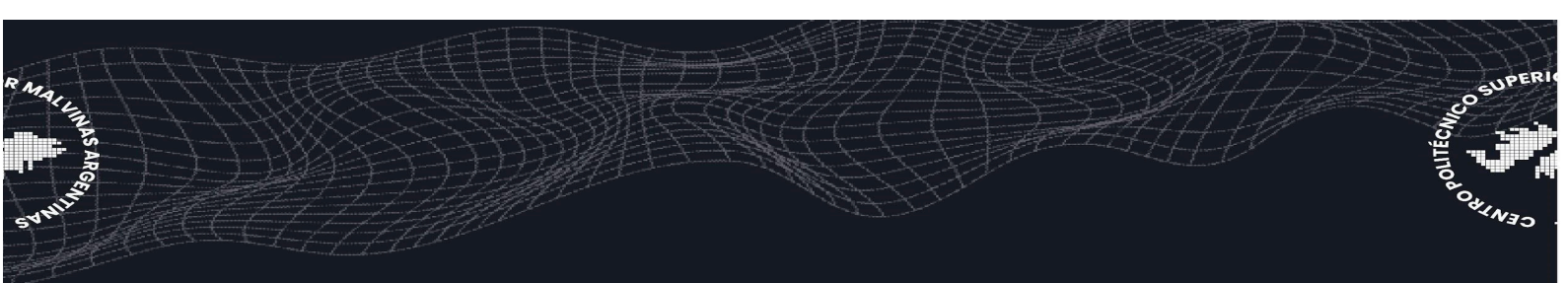
En Visual Studio Code (VSCode), las extensiones son complementos que permiten ampliar las funcionalidades y personalizar el entorno de desarrollo según las necesidades específicas del usuario. Estas extensiones pueden ser creadas por Microsoft, la comunidad de desarrolladores o empresas externas.

Para instalar las extensiones que necesitamos, hacemos click en el menú vertical de la izquierda, haciendo click en el siguiente icono  Por el momento vamos a necesitar instalar dos extensiones, al hacer click

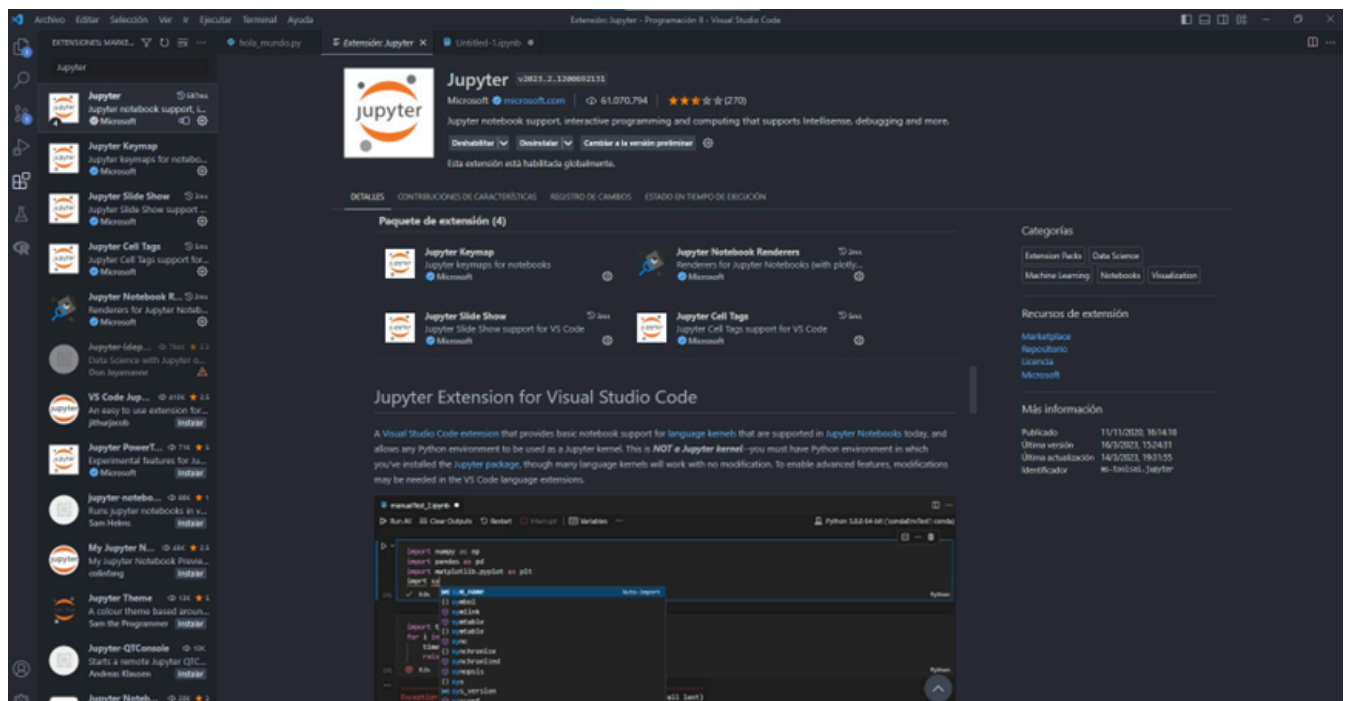
en el icono señalado se nos abre un campo de búsqueda



escribimos **Python** e instalamos la primera opción que nos aparece



Una vez instalado, vamos al campo de búsqueda y escribimos **Jupyter**



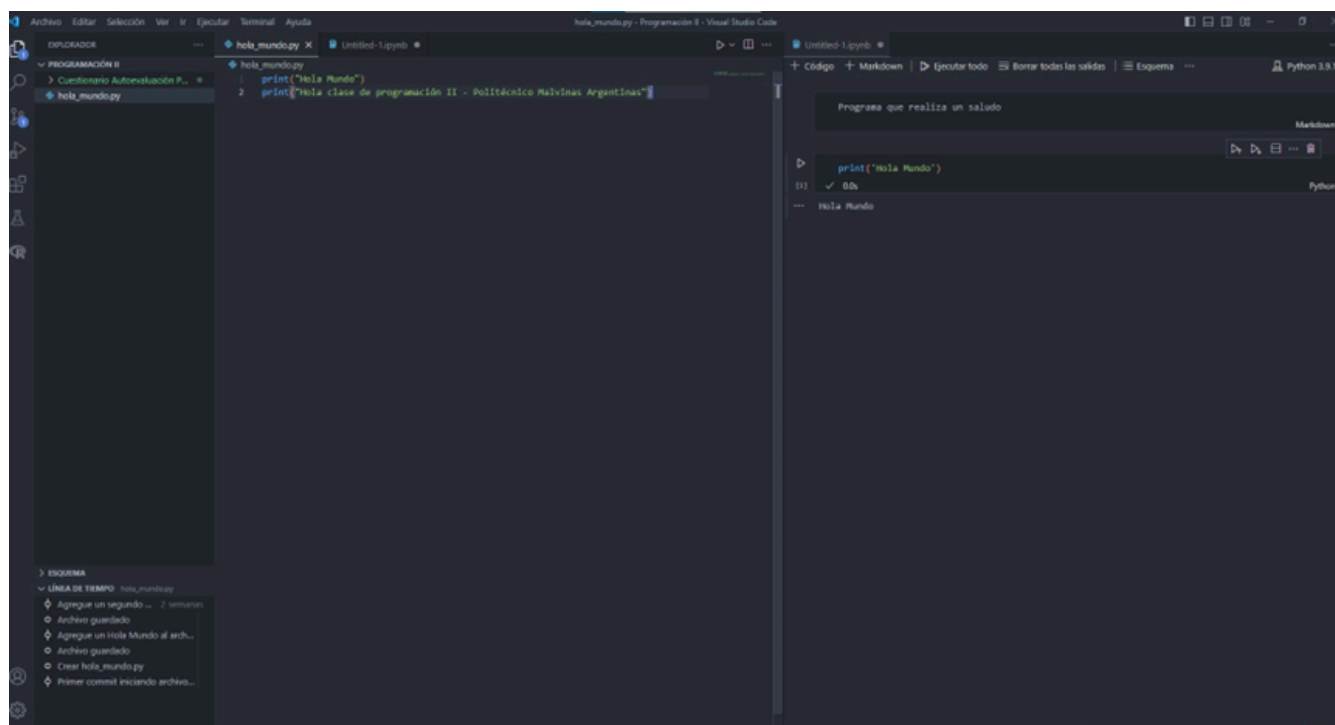
Nuevamente procedemos a instalar la primera opción que nos aparece.



Jupyter Notebook:

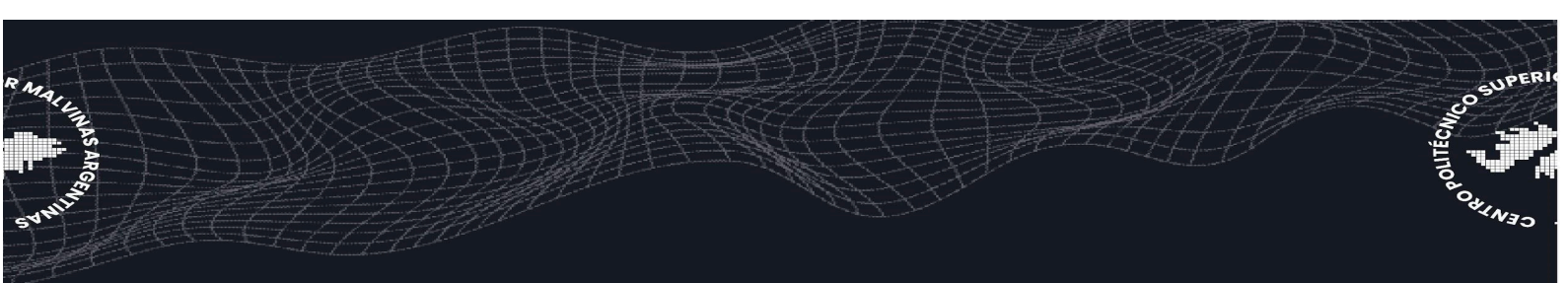
Es una aplicación web, también de código abierto que nos va a permitir crear y compartir documentos con código en vivo, ecuaciones, visualizaciones y texto explicativo. Estos documentos registran todo el proceso de desarrollo y, lo más interesante, pueden ser compartidos fácilmente con otras personas a través de correo electrónico, Dropbox, sistemas de control de versiones como git/GitHub. Los jupyter notebook se pueden trabajar dentro de VsCode a través de la extensión que hemos instalado previamente, combina celdas de código con celdas de texto donde mientras armamos nuestro código va quedando claramente documentado.

La extensión de los archivos notebook es .ipynb a diferencia de los archivos de script de python que son .py



Que es Git y cómo funciona **git**

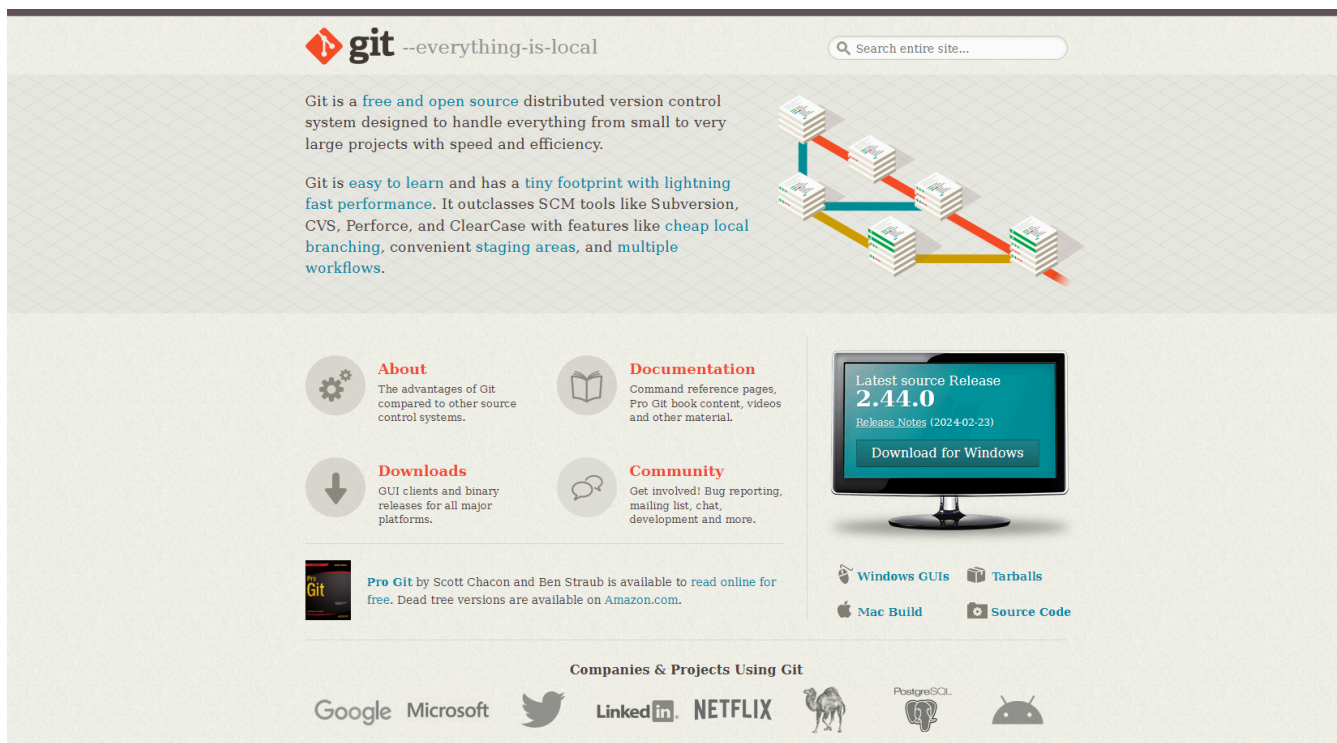
Git se ha convertido en un estándar mundial para el control de versiones en proyectos de desarrollo. Podríamos hablar mucho sobre Git. Podríamos decir que se trata de un



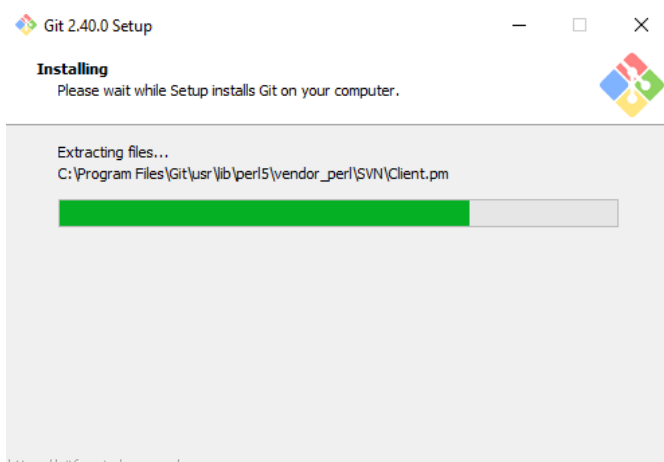
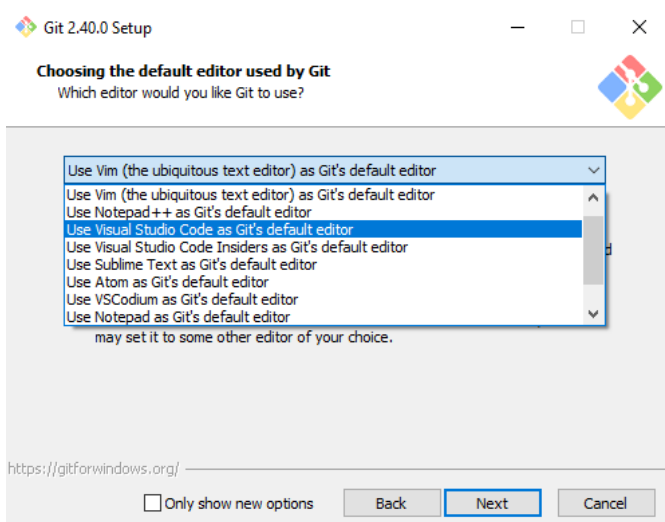
proyecto de código abierto desarrollado por Linus Torvalds, también que es una herramienta clave para el desarrollo de proyectos de software o que se trata de un sistema que presenta una arquitectura distribuida y, por supuesto, también segura. Pero, más allá de toda esta descripción compleja, el resumen es que Git se trata de un sistema de control de versiones con el que podremos disponer de todo un historial de cambios para proyectos de software con los que estemos trabajando. Supongamos que disponemos de un código fuente al que estamos realizando una serie de cambios. En el momento que queramos podemos marcar el nuevo estado en el que se encuentra dicho código fuente y anotarlo como cambio. Esto generará una nueva versión del código fuente. Aparentemente, dispondremos ahora de dos códigos fuente, el primero en su estado inicial y el segundo en su estado modificado. Sin embargo, Git no genera un nuevo archivo con una copia completa de todo el código fuente modificado, sino que simplemente almacena cuáles han sido los cambios en cuestión. De este modo, vamos generando un historial de cambios al que podemos volver en el momento que queramos. Además, Git nos permite trabajar en localizaciones a las que llamamos **"repositorios"**. Disponemos de repositorios locales en nuestros equipos de trabajo y de repositorios remotos almacenados en la nube. Ambos repositorios se conectarán y establecerán comunicación cuando nosotros lo indiquemos, transfiriendo nuestro trabajo del repositorio local al repositorio remoto y viceversa. Si trabajamos en equipo para un mismo proyecto, el resto de personas también podrán subir sus cambios al repositorio remoto. A través de Git podremos ir generando un historial de modificaciones que quedará alojado en la nube. Por supuesto, también podremos acceder a una versión concreta de nuestro código para trabajar con ella. Git se ha convertido en un estándar mundial para el control de versiones en nuestros proyectos de desarrollo. Y en esta lección nos hemos introducido aquí y hemos aprendido qué es y para qué sirve.

Instalación de git para Windows 

Ingresamos al sitio <https://git-scm.com/> y descargamos la última versión

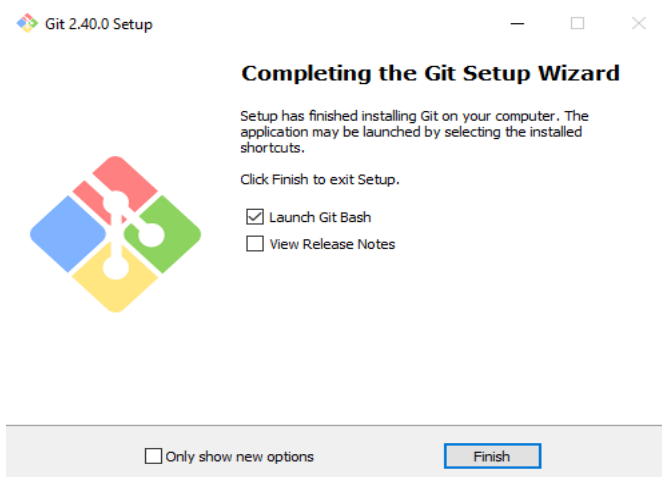


En este momento de la instalación, elegimos VsCode como el editor por defecto para trabajar con Git

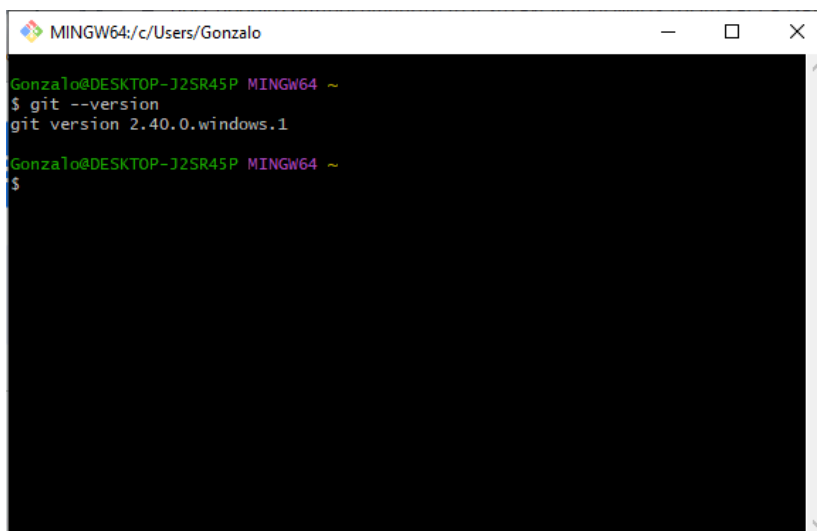


Una vez finalizada la instalación

marcamos el check para lanzar el Git Bash y presionamos Finish



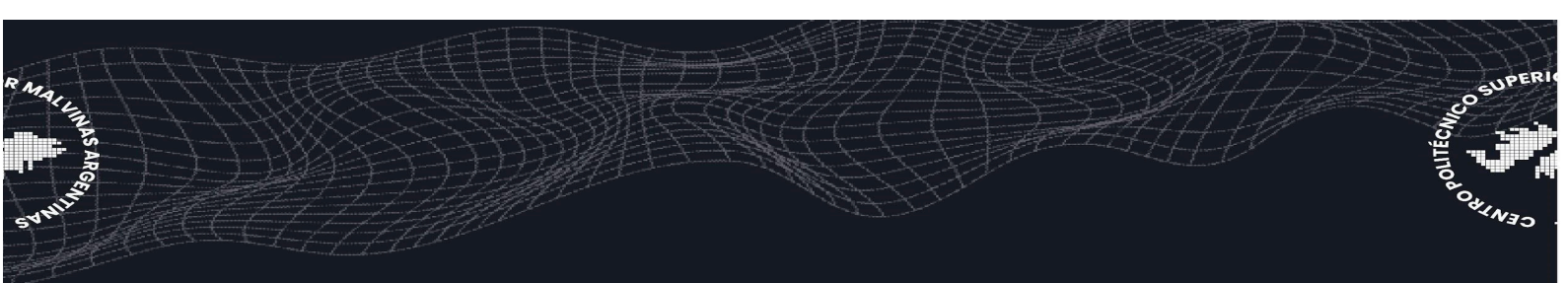
Al ejecutar el git bash se nos abre la consola de git, donde podemos ejecutar el comando ***git --version*** para chequear que esté instalado correctamente.



Instalación para Linux 🐧 o Mac 🍏

Acceder a alguno de los siguientes links:

<https://git-scm.com/book/es/v2/Inicio---Sobre-el-Control-de-Versiones-Ins>



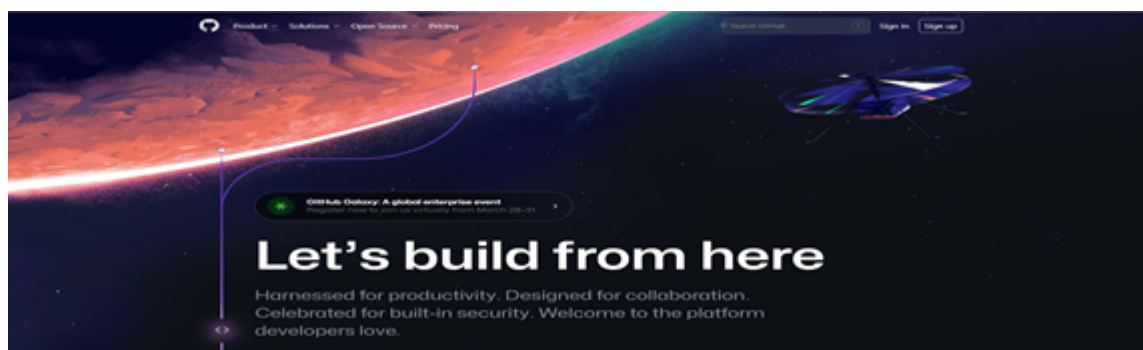
[t alaci%C3%B3n-de-Git](#)

<https://www.youtube.com/watch?v=FxN8uDk65Oo>

<https://www.youtube.com/watch?v=oV0spTF71AI>

Que es GitHub **y que ventajas ofrece:**

GitHub es un portal basado en Git que permite a los desarrolladores y desarrolladoras de software subir código de sus aplicaciones y herramientas, trabajar juntos de forma colaborativa. Ya sabemos que Git es un sistema de control de versiones que nos permite generar un historial de cambios de nuestro proyecto. Por supuesto, podemos acceder a las diferentes versiones generadas y trabajar sobre diferentes estados de nuestro código. Además, sabemos que podemos trabajar con Git a partir de repositorios locales y remotos. Partiendo de esta premisa, y siendo conscientes de que los repositorios locales van a alojarse en el equipo con el que estamos trabajando, existen numerosos servicios que nos permiten almacenar nuestro código en repositorios remotos y GitHub es sin duda el más conocido. GitHub principalmente consiste en un conjunto de repositorios remotos donde usuarios y organizaciones pueden alojar sus proyectos. En GitHub podremos crear nuestros propios repositorios remotos, alojarlos de forma pública o privada y conectarnos a ellos desde nuestros equipos de trabajo.



Cómo crear una cuenta en GitHub:



Si no tenemos una cuenta creada, el primer paso es dirigirnos a la url <https://github.com/> una vez en el sitio vamos a hacer click en “sign up” una vez hecho esto, nos pedirá que ingresemos una cuenta de correo

Una vez ingresado nuestro correo, nos pedirá que ingresemos una clave

Una vez ingresada la clave, nos pedirá que ingresemos un nombre de usuario.

Deben prestar atención para verificar si el nombre de usuario que quieran utilizar está disponible o ya está siendo utilizado.

Luego nos va a preguntar si queremos recibir en nuestro correo información sobre actualizaciones del producto o anuncios, debemos setear “y” por si o “n” por no.



Would you like to receive product updates and announcements via email?

Type "y" for yes or "n" for no

→

Continue

A continuación nos pedirá que resolvamos un acertijo 🤔 para comprobar que somos una persona real. ¡No se preocupen! , no son complicados 😊

Verificación

Resuelva este acertijo para que sepamos que es una persona real


Verificar



Una vez resuelto el acertijo, que consiste de 5 pasos, se realiza la verificación y si hicimos el acertijo correctamente, estaremos listos para crear nuestra cuenta 💪



Verify your account



Create account

hacemos click en “Create account” y a continuación nos enviarán un código a la casilla de correo que hemos indicado, revisamos nuestro correo y completamos el código.

→ Enter code

Una vez ingresado el código nos hará una serie de preguntas para configurar nuestra



cuenta

How many team members will be working with you?

This will help us guide you to the tools that are best suited for your projects.

Just me 2 - 5 5 - 10

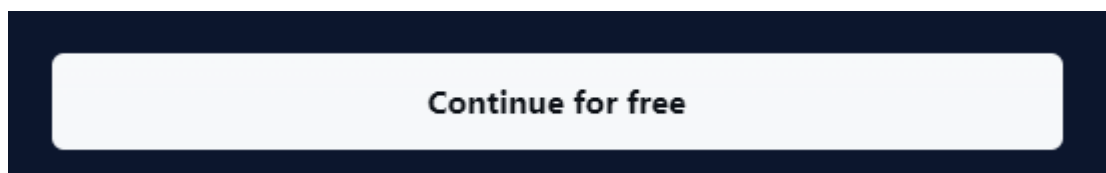
10 - 20 20 - 50 50+

Are you a student or teacher?

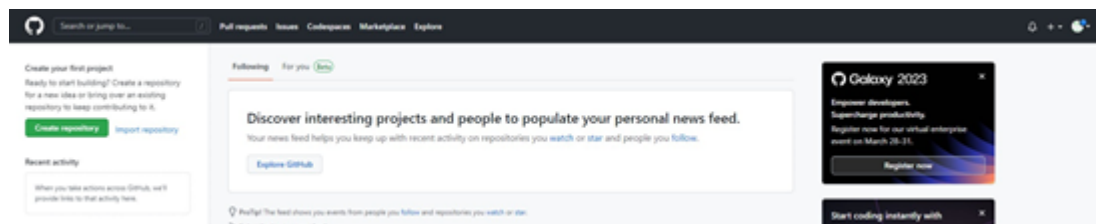
Student Teacher

Continue

seteamos en “Just me! y debajo en “Student”. A continuación nos dará la opción de si queremos acceder a la versión Pro o continuar con la libre, presionamos en la opción de continuar con la libre.



y listo! ya tenemos nuestra cuenta GitHub lista para empezar a crear nuestros repositorios de código 🤓



Configurar nombre y correo electrónico en GIT.

Abrimos una terminal o línea de comandos y ejecutamos los siguientes comandos para configurar nombre de usuario y correo electrónico. Reemplazar “Tu Nombre” y “tu@email.com” con los datos de nombre de usuario y correo con los que hemos creado nuestra cuenta de GitHub

```
git config --global user.name "Tu Nombre"
git config --global user.email "tu@email.com"
```

Una vez que hemos realizado la configuración, podemos verificar que se hayan cargado correctamente, viendo la lista de configuraciones de Git.

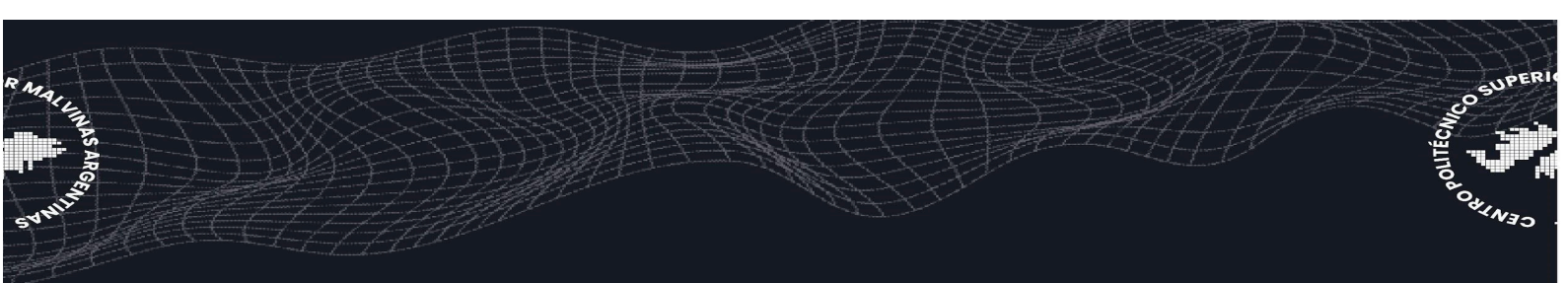
```
git config --list
```

Si necesitamos consultar por configuración en particular, podemos usar el nombre específico del parámetro de configuración.

```
git config <nombre_de_la_configuracion>
```

Ejemplo:

```
git config user.name
```



Integración nativa de Git en Visual Studio

A diferencia de versiones anteriores, Visual Studio ya dispone de integración nativa con Git, una decisión que facilita la puesta en marcha tanto de nuevos proyectos como de proyectos ya existentes. Visual Studio es un entorno de desarrollo integrado desarrollado por Microsoft con numerosas características que respaldan muchos aspectos del desarrollo de software. En versiones anteriores, la integración con Git y GitHub en Visual Studio se hacía a través de una extensión que necesitábamos instalar. A día de hoy, desde Visual Studio 2022 contamos con una mejor integración con Git y GitHub.

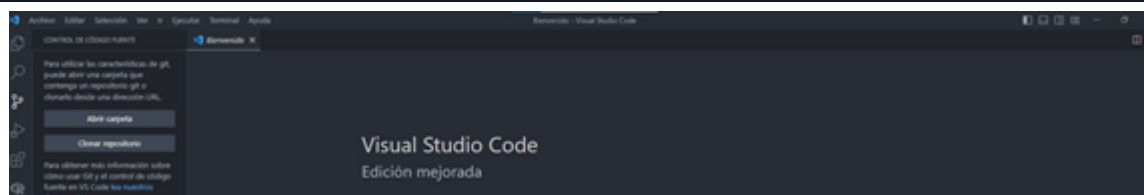
Conectar VsCode con Git y GitHub.

Ahora vamos a ver cómo integrar Git y GitHub dentro de VsCode para trabajar de una forma fácil y sencilla 😊.

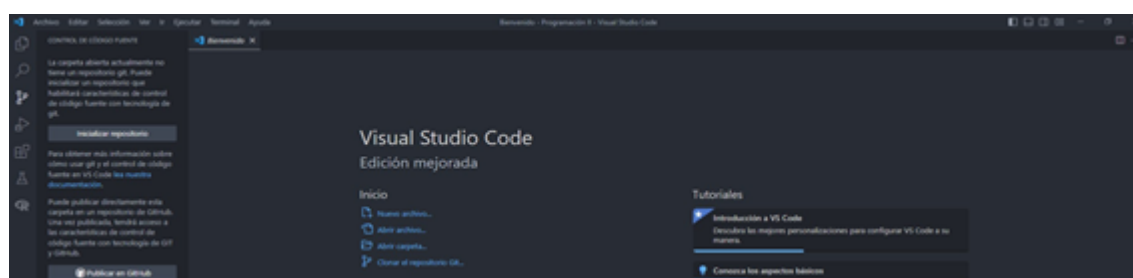
Lo primero que les recomiendo es tener una carpeta de la materia para ir almacenando los archivos de práctica y los proyectos. En mi caso yo lo tengo armado de esta manera ***C:\Politecnico Malvinas\Técnicatura en Ciencia de Datos\Programación II***. Abrimos VsCode..



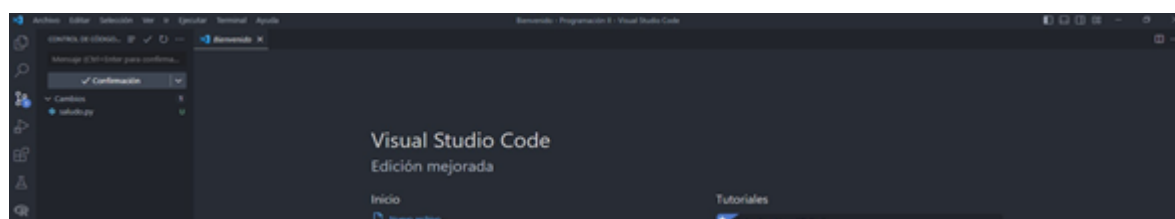
Nos dirigimos a la pestaña de “Control de código fuente” con la combinación de teclas CTRL + SHIFT + G o haciendo click en ella. De los dos botones, utilizamos el de “Abrir carpeta”.



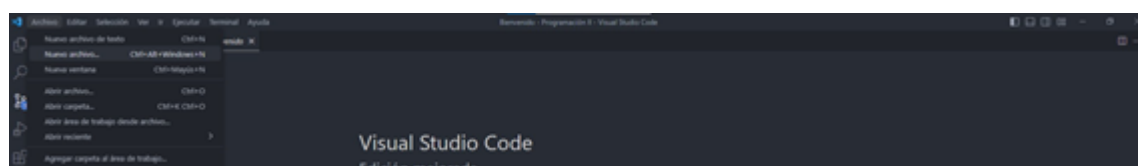
Acá seleccionaremos la carpeta que contenga todos los archivos de nuestro proyecto o crearemos una nueva, en mi caso ***C:\Politecnico Malvinas\Técnicatura en Ciencia de Datos\Programación II***. Una vez que hemos seleccionado nuestra carpeta de proyectos, nos dirigimos a la opción “Control de código fuente” y ya tenemos la opción de iniciar nuestra carpeta como repositorio de Git.



Hacemos click en el botón, “Iniciar repositorio” y ya estamos listo para trabajar con nuestro sistema de control de versiones de código 🙌



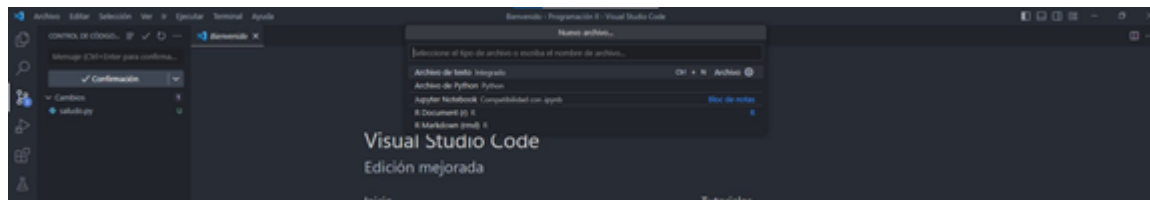
Ahora ya tenemos nuestro repositorio y vamos a crear nuestro primer archivo, para esto vamos a ir al menú “Archivo” y dentro a “Nuevo Archivo”



una vez hecho esto y si ya hemos instalado las extensiones de la clase pasada,

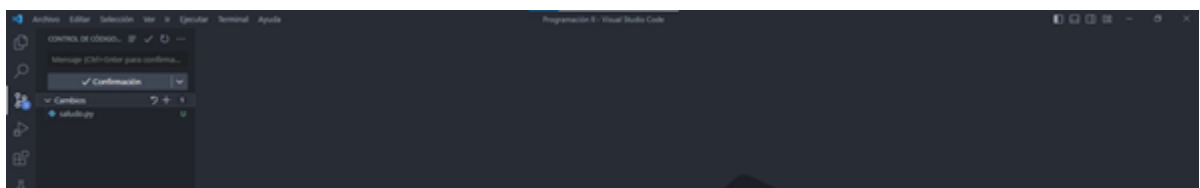


nos mostrará las siguientes opciones:



Podemos generar el archivo como script de python, con extensión **.py** o podemos generar un Jupyter Notebook con extensión **.ipynb**. Para este primer ejemplo vamos a elegir la opción “Archivo de Python” y los vamos a nombrar como “saludo.py”

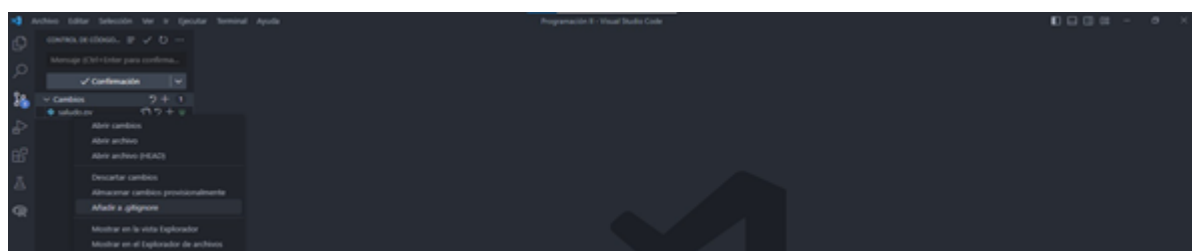
Ya hemos creado un archivo nuevo en el repositorio y en la pestaña de “Control de código fuente” nos aparece ya un cambio.

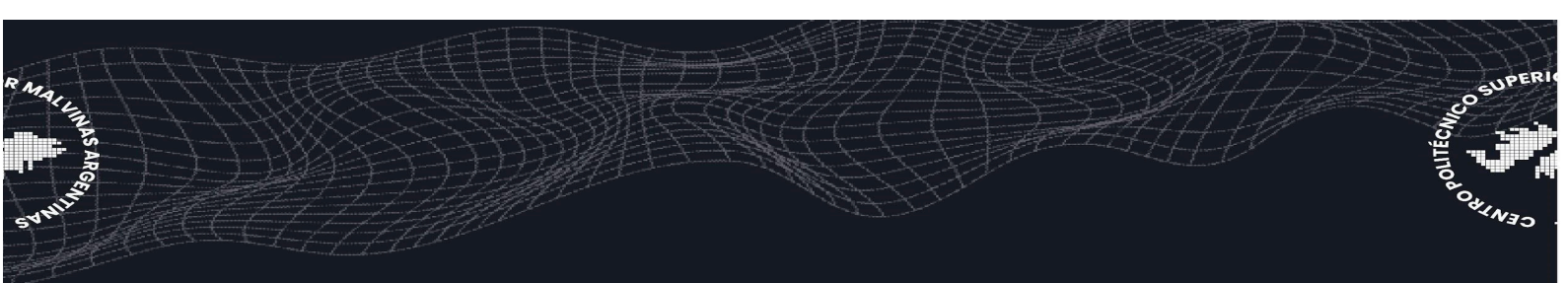


Si miramos a ver que es, veremos que nos aparece un archivo con cambios y una “U” que representa el estado “Untracked” (sin seguimiento).

Añadir archivo a .gitignore

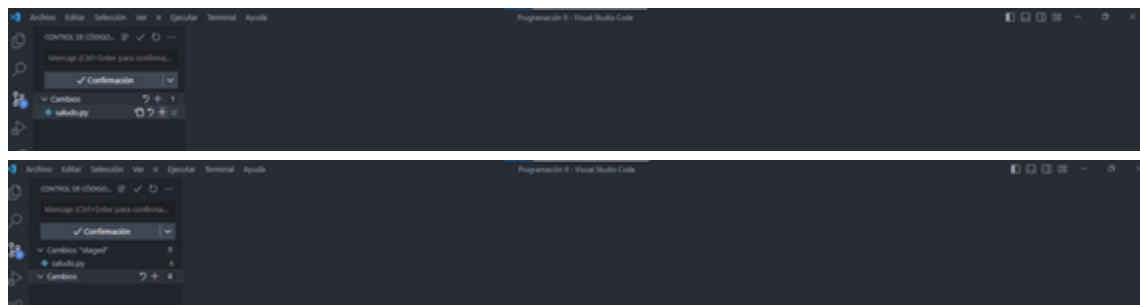
gitignore es un archivo de texto que le dice a Git qué otros archivos y carpetas de un proyecto debe ignorar. Si hacemos click derecho sobre un archivo, lo podemos añadir a **.gitignore** para que se ignore en el repositorio.





Añadir archivos al área de staging

Para añadir archivos al área de staging (área de ensayo) desde la interfaz gráfica de Visual Studio Code, solo tienes que pulsar el botón “+”:



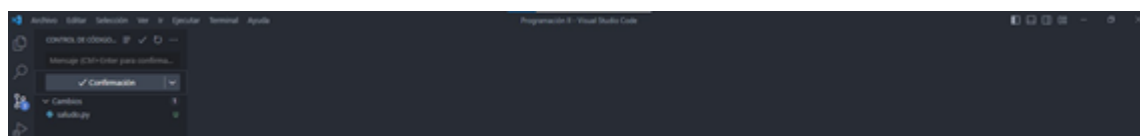
Al hacerlo, vemos que el archivo cambia de estado para Git y pasa de no seguido, a agregado. El archivo nos sale como “staged” en estado “A” addition of a file (adición de un archivo). En este estado, el archivo ya se encuentra en condiciones de ser versionado.

Quitar archivos del área de staging

Si por algún motivo, nosotros quisiéramos que un archivo que se encuentra en estado de seguido, volviera a un estado de no seguido para evitar que sea versionado, lo podemos hacer posicionándonos sobre el archivo y presionando el signo “-”



Al hacerlo nuestro archivo volverá al estado “U”



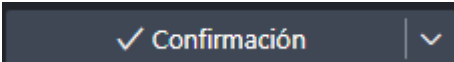
Cómo hacer un commit

El comando git commit captura una instantánea de los cambios preparados en



ese momento del proyecto. Las instantáneas confirmadas pueden considerarse como versiones "seguras" de un proyecto: Git no las cambiará nunca a no ser que se le solicite expresamente.

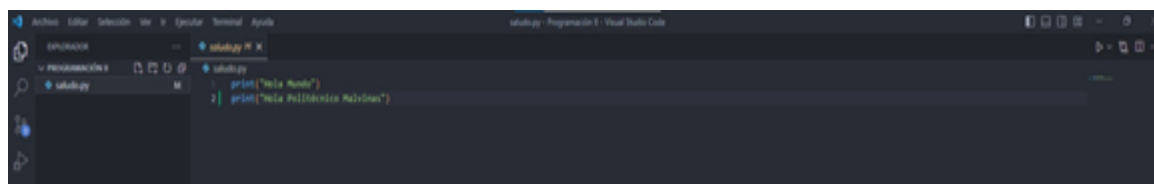
Para confirmar (commit) todo lo que hay en el área de staging, se escribe un mensaje en el cuadro de texto y pulsamos las teclas CTRL + ENTER o bien en

el botón 



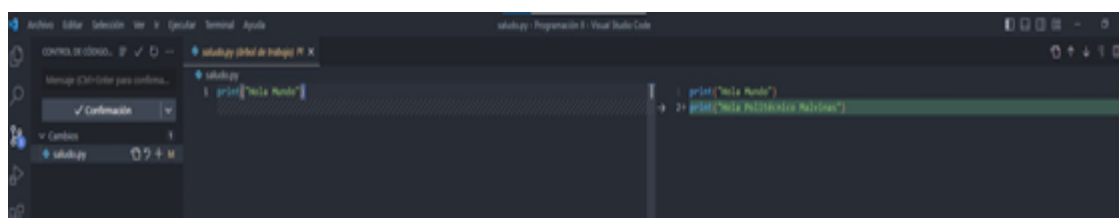
Gestionar archivos modificados

Ahora que el archivo ya tiene seguimiento, cuando añadamos una modificación en el archivo, nos aparecerá con la “M” de modificado (“modified”).



Ver las modificaciones de los archivos

Para ver las modificaciones en los archivos, solo tienes que hacer click en cada archivo y te aparecerá una sección con el antes y otra con el después



Actividad:

Como actividad de la clase vamos a instalar Git, configurar nuestra cuenta de GitHub y configurar nuestro entorno de trabajo en VsCode para trabar con Git y GitHub



Instancia de Autoevaluación:

En esta instancia de autoevaluación te pedimos que puedas poner en juego lo que has aprendido hasta este momento. Recordá que siempre podrás volver a la teoría presentada en esta clase o hacer las preguntas pertinentes en los encuentros sincrónicos o las tutorías presenciales.

Es importante tener en cuenta que para aprobar esta instancia deberás:

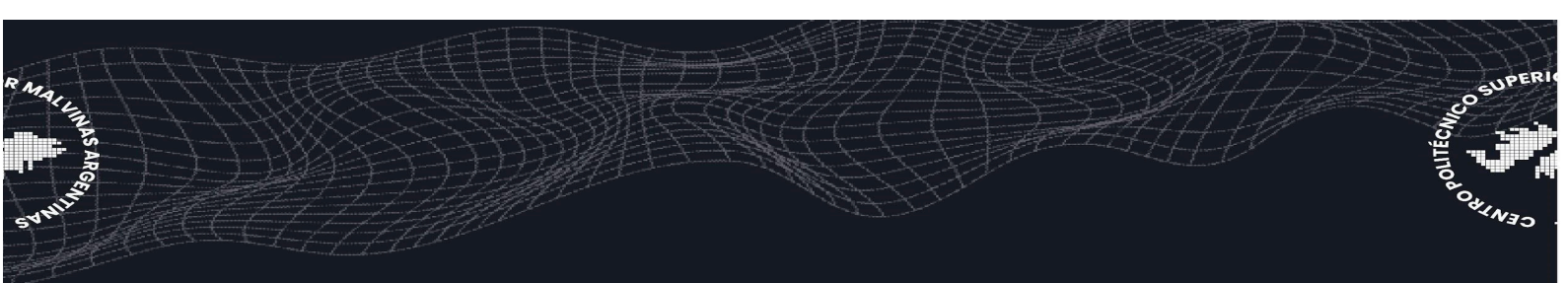
- Entregar en tiempo y forma
- Realizar un 60% de la actividad correctamente.
- Cumplir con las consignas
- Cumplir con el formato requerido.

Cierre:

Llegamos al final de la segunda clase 😞. En esta clase vimos el sistema de control de versiones más utilizado en el mundo del desarrollo y vimos la plataforma para trabajar con repositorios remotos más utilizada. También vimos cómo integrar esto a nuestro entorno de trabajo para tener todo en un solo lugar

En la próxima clase con nuestro entorno de trabajo ya configurado, vamos a seguir trabajando Git y Github 💪, veremos como crear un nuevo repositorio de Git en GitHub, crear un nuevo repositorio de Git en local y en GitHub a través de visual studio, eliminación de archivos en GitHub, añadir colaboradores a un repositorios y varios temas más vinculados al trabajo con repositorio de versiones . Que tengan una excelente semana. Los espero con mucha energía 🍷 listos para programar. Saludos a todos!





Bibliografía Obligatoria:

- Ramírez Jiménez, Ó. (2021). *Python a fondo*. Marcombo.

Recursos adicionales:

- ¿Qué es un Jupyter Notebook? (2021, October 4). YouTube.
Retrieved April 7, 2023, from <https://www.youtube.com/watch?v=G4K1jKZH7nE>
- Entrenamiento de programación en Python 3 - Nivel básico.
(2023, January 1). Programación en Python. Retrieved April 7, 2023, from <https://entrenamiento-python-basico.readthedocs.io/es/3.7/>
- (n.d.). El Libro De Python: 🏠 Inicio. Retrieved April 7, 2023, from <https://ellibrodepython.com/>
- Curso Python. Video 1. (2017, January 24). YouTube. Retrieved April 7, 2023, from <https://www.youtube.com/watch?v=G2FCfQj-9ig>
- (n.d.). Visualize Code Line-by-Line. Retrieved April 7, 2023, from <https://memlayout.com/>
- Python in Visual Studio Code. (n.d.). Visual Studio Code. Retrieved April 7, 2023, from <https://code.visualstudio.com/docs/languages/python>