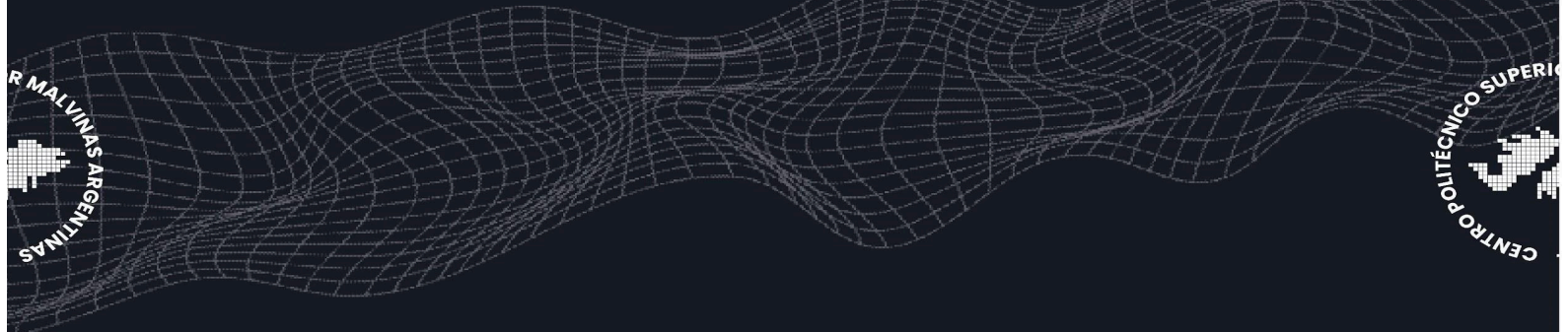


## Nombre del bloque PROGRAMACIÓN II

Año de cursada 1° AÑO

### Clase N° 4: Inteligencia Artificial para potenciar tu desarrollo en Python con Visual Studio Code





## **Contenido:**

En la clase de hoy trabajaremos los siguientes temas:

- Gestión y utilización de ramas.
- Puesta a punto de un repositorio.
- Extensiones de IA para Python en VSCode.
- Beneficios de usar IA en Python.

## **Presentación:**

¡Bienvenidos a la clase IV de Programación II! 🙌 En nuestra última clase, nos sumergimos en el fascinante mundo de Git y GitHub, explorando cómo crear y eliminar ramas tanto en GitHub como en Visual Studio Code. También aprendimos sobre la colaboración y la sincronización de repositorios entre estas plataformas. Hoy, continuaremos nuestro viaje a través del sistema de control de versiones mientras nos adentramos en temas más avanzados y emocionantes.

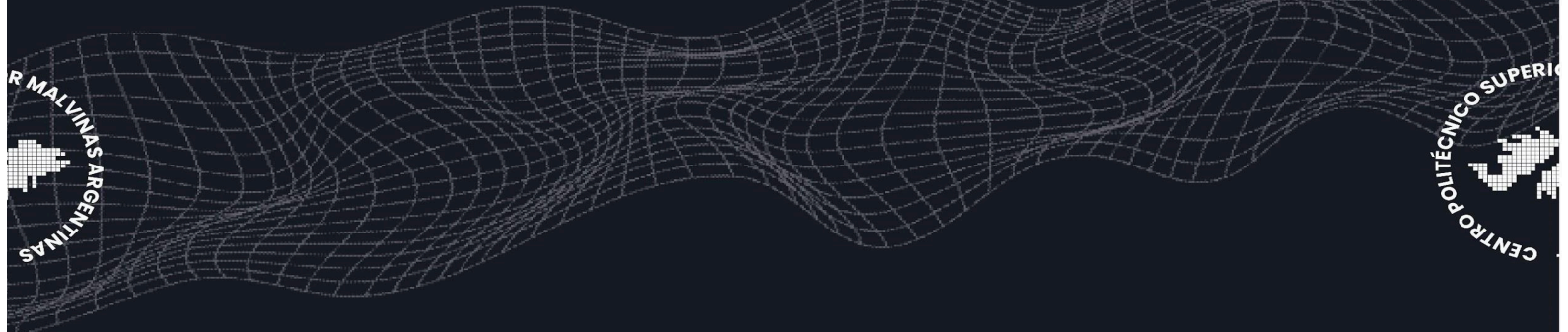
Repaso rápido:

En la clase anterior, establecimos una base sólida al aprender sobre Git, un sistema de control de versiones distribuido que nos permite rastrear y gestionar cambios en nuestro código de manera eficiente. Además, exploramos GitHub, una plataforma esencial para alojar proyectos de software y fomentar la colaboración entre desarrolladores de todo el mundo. No olvidemos cómo integramos Git en Visual Studio Code para mejorar nuestra experiencia de desarrollo.

Temas de hoy:

Hoy nos sumergiremos aún más en el vasto océano de la programación, explorando los





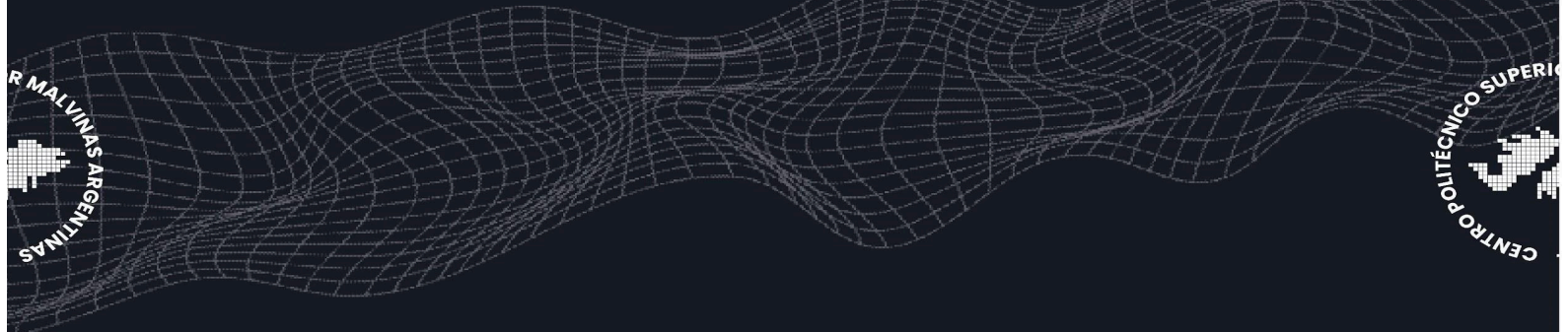
siguientes temas:

1. **Gestión y utilización de ramas:** Profundizaremos en el poder de las ramas en Git, aprendiendo cómo crear, fusionar y eliminar ramas de manera efectiva para mantener un flujo de trabajo organizado y eficiente.
2. **Puesta a punto de un repositorio:** Descubriremos la importancia de mantener una estructura de archivos organizada en nuestros repositorios tanto en GitHub como en Visual Studio. Aprenderemos a agregar, modificar y eliminar archivos de manera efectiva para garantizar un desarrollo fluido.
3. **Extensiones de IA para Python en VSCode:** Exploraremos cómo utilizar herramientas de inteligencia artificial dentro de Visual Studio Code para potenciar nuestro desarrollo en Python. Veremos cómo estas extensiones pueden mejorar nuestra productividad y eficiencia en el proceso de escritura de código.
4. **Beneficios de usar IA en Python:** Discutiremos los numerosos beneficios que ofrece la inteligencia artificial en el desarrollo de software en Python, desde la automatización de tareas repetitivas hasta la optimización de algoritmos y la mejora de la experiencia del usuario.

Objetivos de Aprendizaje:

En esta sesión, nuestro objetivo es consolidar y ampliar los conocimientos adquiridos en clases anteriores. Queremos:

- Reforzar nuestra comprensión de Git, GitHub y su integración con Visual Studio Code.
- Dominar técnicas avanzadas de colaboración y sincronización de repositorios.



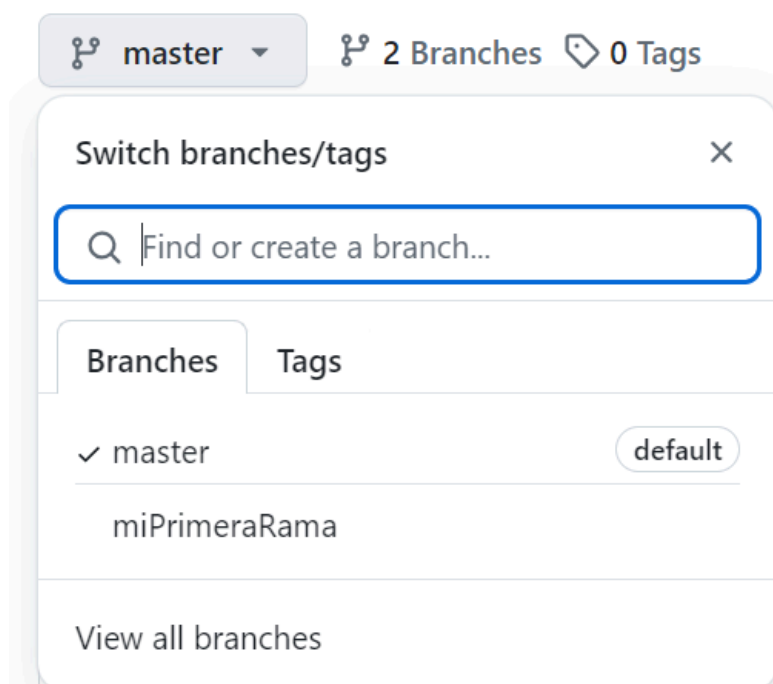
- Desarrollar habilidades para gestionar eficientemente ramas y resolver conflictos en el desarrollo de software colaborativo.
- Explorar las posibilidades que ofrece la inteligencia artificial para mejorar nuestro trabajo en Python.

Los animo a participar activamente, hacer preguntas y colaborar entre ustedes. Estoy aquí para ayudarlos a comprender y aprovechar al máximo estos conceptos. ¡Sigamos aprendiendo juntos y disfrutemos de esta clase! 🚀

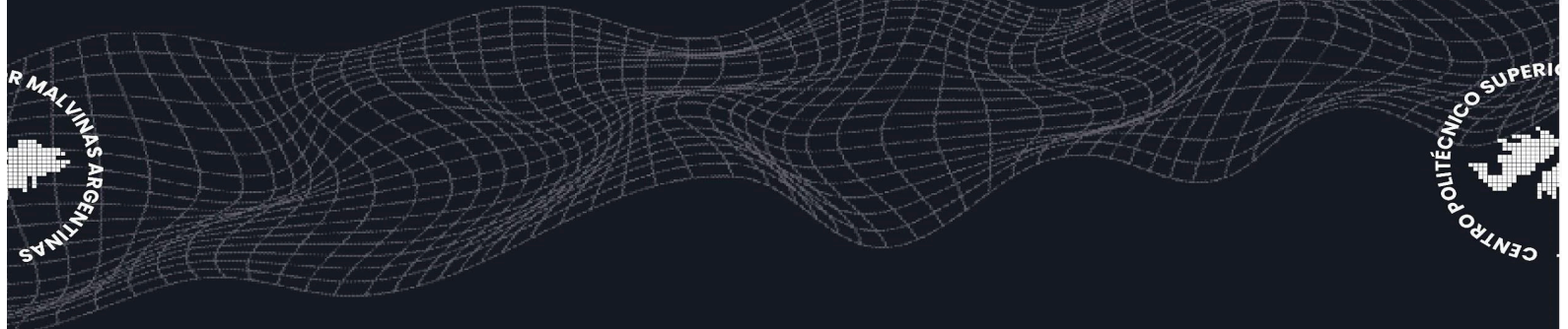
## Desarrollo y Actividades:

### Eliminación de una rama existente en GitHub

No solo podemos crear ramas, sino también eliminarlas. Para eliminar una rama, desde GitHub debemos haber iniciado sesión y disponer de permisos de escritura en el repositorio en el que vamos a eliminar la rama en cuestión. Accedemos a la URL del repositorio, hacemos clic en este desplegable



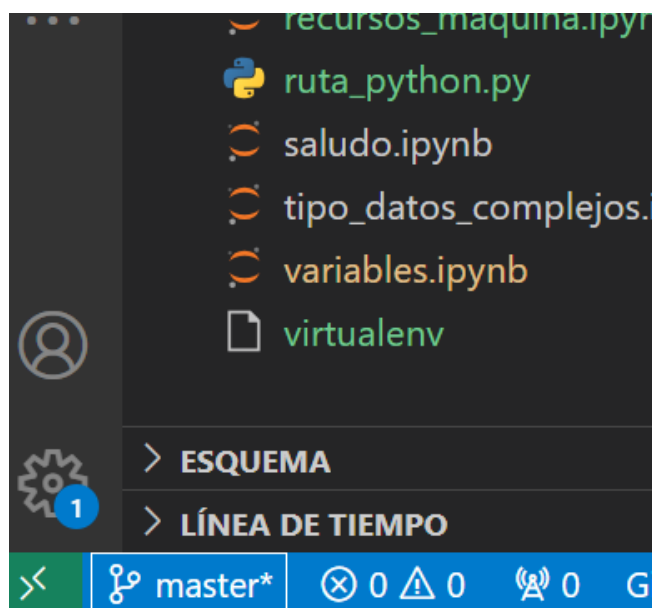
y, a continuación, View all branches (ver todas las ramas). Si ya disponemos de al menos una rama creada diferente a la principal, en la página a la que somos redirigidos podremos ver un listado con información muy interesante. En primer lugar encontramos una sección llamada Default branch (rama por defecto). En segundo lugar, encontramos Your branches (tus ramas), un listado con el resto de ramas disponibles en nuestro repositorio, aunque no tienen por qué estar activas. Y por último encontramos la sección Active branches, que son aquellas ramas que sí se encuentran activas en este momento.



Para eliminar una rama existente, tratamos de localizarla en la sección Your branches. Y a la derecha encontramos un icono de una papelera que nos permitirá, si hacemos clic en él, eliminar esta rama. Debemos tener mucho cuidado, ya que la eliminación de una rama puede provocar una pérdida muy significativa de información sobre nuestro repositorio. Tras hacer clic, nuestra rama habrá quedado borrada.

### **Cambiar de una rama a otra en Visual Studio**

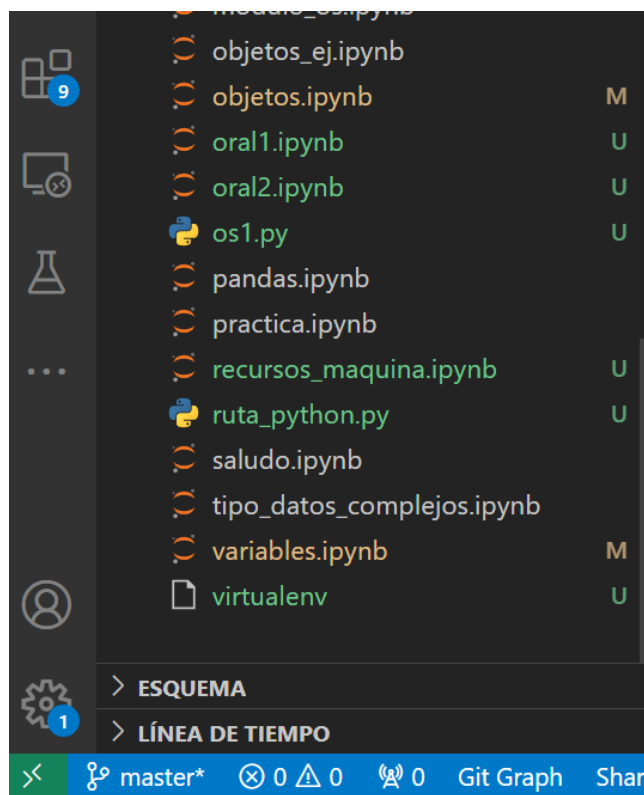
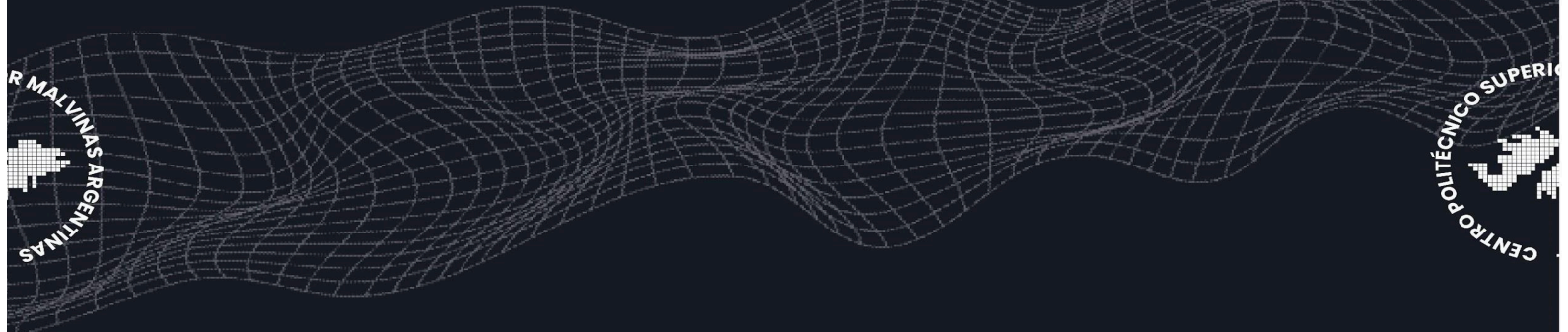
Una de las ventajas que Git nos ofrece es la de poder cambiar de una rama a otra para poder trabajar con diferentes estados del proyecto. Cambiar de una rama a otra en Visual Studio es realmente sencillo y vamos a ver las implicaciones que presenta. En estos momentos nos encontramos trabajando en la rama máster. Lo sabemos porque en la interfaz de Visual Studio podemos ver en esta zona el nombre de la rama en cuestión.



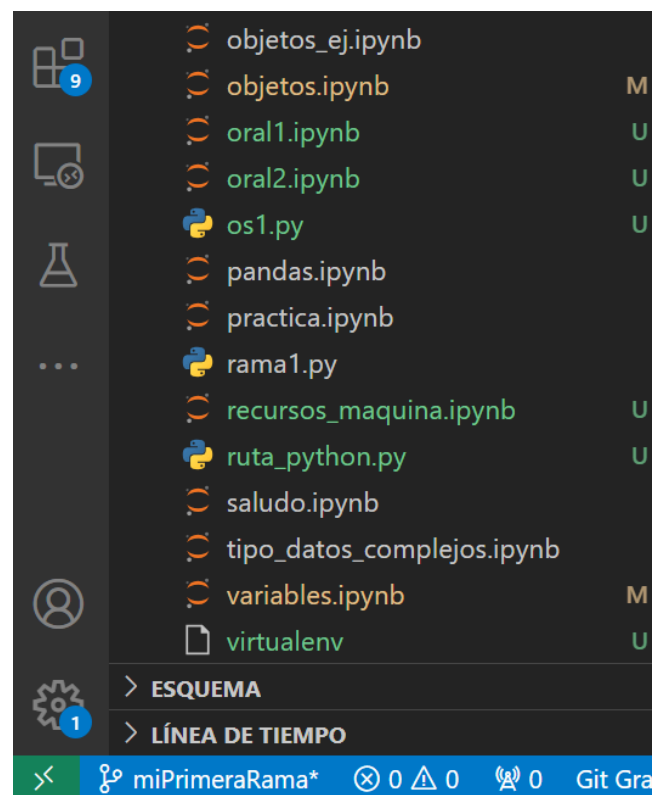
Para cambiar de rama, lo único que tenemos que hacer es precisamente en esta zona hacer clic y elegir la rama a la que queremos movernos. Pero antes de hacer clic fíjate muy bien qué es lo que ocurre en el explorador de soluciones. Cuando cambiamos de rama, el proyecto pasa al estado en el que se encuentra en la nueva rama y nuestro listado de archivos se actualiza en consecuencia. Como podemos

apreciar, en esta rama disponemos de un archivo que no estaba en la rama máster. A su vez, si volvemos a la rama máster, dicho archivo desaparecerá en consonancia con el estado del proyecto en la rama en la que nos encontramos.



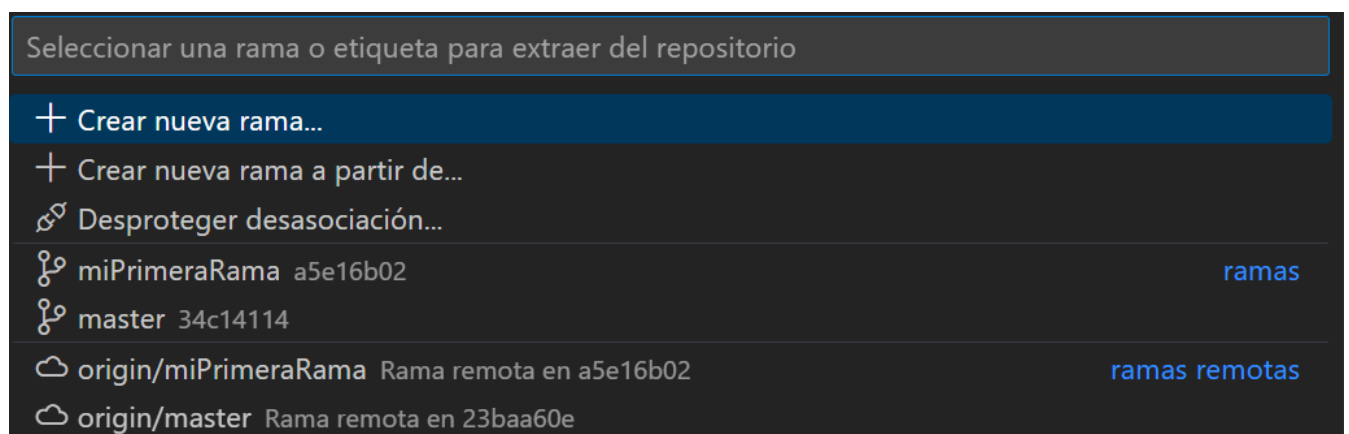


### Rama Master



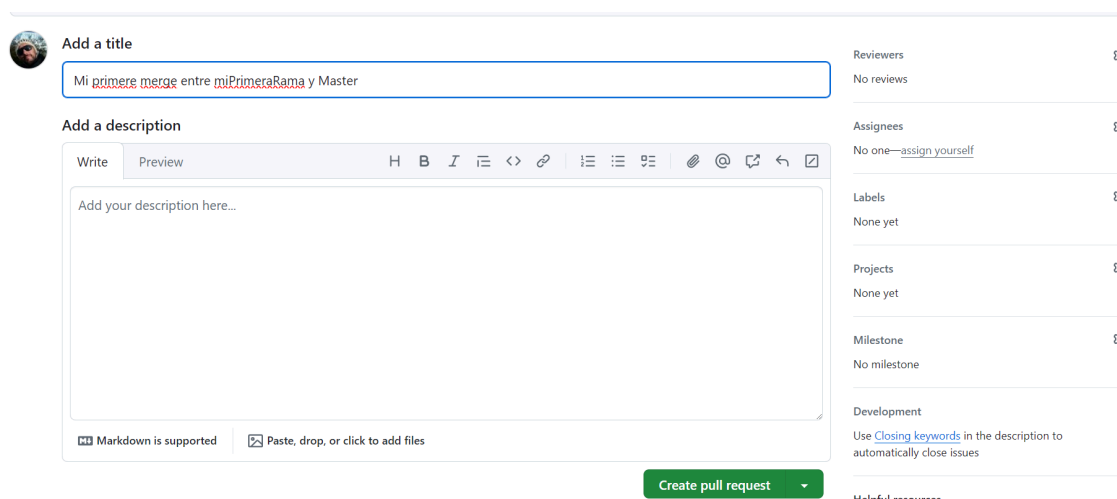
### Rama miPrimeraRama


También debemos tener en cuenta que si queremos cambiar a una rama que no aparece en este listado pero sabemos que existe, probablemente se encuentre en donde dice “ramas remotas”. En caso de que la localices ahí, solo tendrás que hacer doble clic en la rama en cuestión para empezar a trabajar sobre ella.



## Insertar los cambios de una rama en otra en GitHub

Dos ramas con dos estados diferentes de un mismo proyecto pueden ser unificadas. Para insertar los cambios de la rama en la que hemos estado trabajando en otra diferente, por ejemplo en la rama máster, a través de GitHub, accedemos a la URL en el repositorio en cuestión, hacemos clic en el cuadro que nos dice en qué rama nos encontramos y a continuación seleccionamos la opción View all branches (ver todas las ramas). En el listado de ramas que vemos en pantalla localizamos la rama en la que hemos estado trabajando. En este caso hemos elegido, por ejemplo, la rama 'miPrimeraRama'. Y a su derecha, los tres puntos horizontales a la derecha, localizamos la opción "New pull request". Hacemos clic y seremos redirigidos a una página que nos permitirá abrir lo que se conoce como un "pull request". Un "pull request" es la acción de validar un código que se va a unir de una rama a otra. Como podemos observar, el contenido de la rama 'miPrimeraRama' se uniría a la rama máster, en este caso nombrada "master" (rama principal). Se produciría lo que se conoce como un "merge", una unión de una rama a otra. Escribimos un título y una descripción para la solicitud de validación y esta solicitud se enviará de la rama 'miPrimeraRama' a la rama principal.



 Add a title

Mi primera merge entre miPrimeraRama y Master

Add a description

Write Preview

Add your description here...

Markdown is supported Paste, drop, or click to add files

Reviewers No reviews

Assignees No one—[assign yourself](#)

Labels None yet

Projects None yet

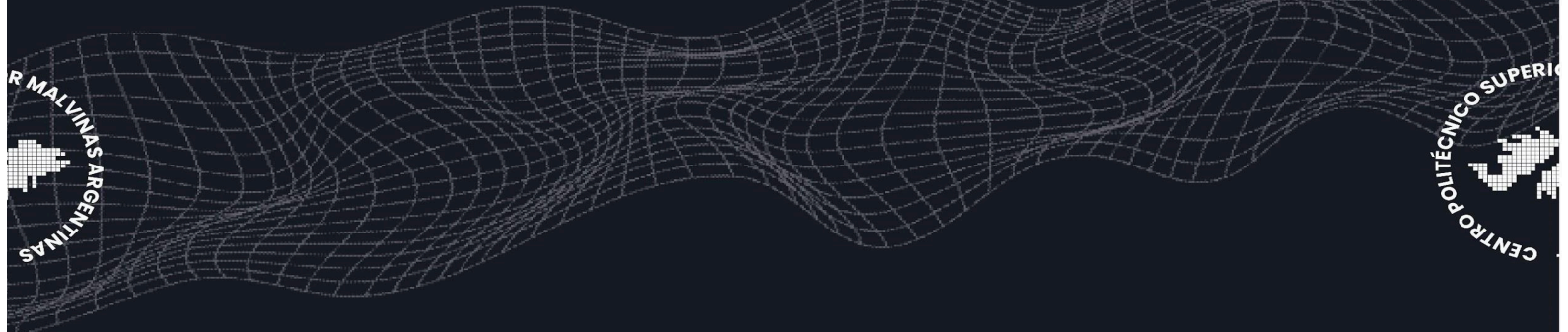
Milestone No milestone

Development Use [Closing keywords](#) in the description to automatically close issues

[Helpful resources](#)

Create pull request





Para ello hacemos clic en Create pull request. Si hubiera algún conflicto entre ramas, GitHub nos lo mostraría en este proceso de validación. Si todo está correcto, entonces nos aparecerá un mensaje similar al que vemos en pantalla. Hacemos clic en Merge pull request y a continuación confirmamos el "merge", confirmamos la unión.

The screenshot displays a GitHub pull request interface. At the top, the commit history shows two commits by 'gpimentel79' from 1 hour ago: 'Added and modified cells in variables.ipynb related to variable assi...' (94ee051) and 'agregando archivo rama1 a miPrimeraRama' (a5e16b0). Below the commits, a message indicates to 'Add more commits by pushing to the `miPrimeraRama` branch on `gpimentel79/Programacion-II-Repo`'. The main section shows three status checks: 'Require approval from specific reviewers before merging' (with a link to Rulesets), 'Continuous integration has not been set up' (with links to GitHub Actions and other apps), and a green checkmark indicating 'This branch has no conflicts with the base branch'. A 'Merge pull request' button is visible. To the right, there are sections for 'Assignees' (No one—assign yourself), 'Labels' (None yet), 'Projects' (None yet), 'Milestone' (No milestone), 'Development' (Successfully merging this pull request may close th issues.), 'Notifications' (Unsubscribe), and 'Custc'. Below the main section, a 'Merge pull request #1 from gpimentel79/miPrimeraRama' dialog is shown, with a text box containing 'Mi primere merge entre miPrimeraRama y Master' and a confirmation message: 'This commit will be authored by 36083041+gpimentel79@users.noreply.github.com'. At the bottom, there is an 'Add a comment' section with a 'Write' button and a rich text editor toolbar.

Esta petición de solicitud y posterior aceptación que hemos realizado nosotros mismos a menudo forma parte de procesos de trabajo dentro de equipos de desarrollo, en los que diferentes miembros envían sus solicitudes de confirmación a una rama concreta y el responsable técnico en cuestión valida el código implementado, pudiendo a la solicitud ser aprobada o denegada.

## Edición del archivo README.md en GitHub

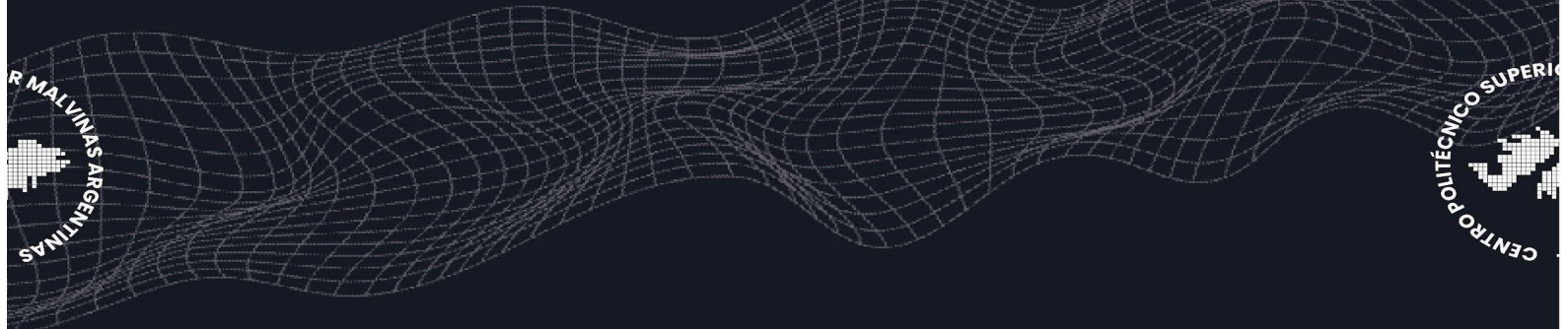
El archivo 'readme' es el primer documento que un desarrollador debería mirar antes de comenzar a trabajar en un proyecto. Para editar el archivo 'readme' desde la interfaz visual de GitHub, accedemos a la URL del repositorio en cuestión, hacemos clic sobre el archivo y a continuación pulsamos sobre el botón de editar.

The screenshot shows the GitHub interface for the repository 'Programacion-II-Repo' by user 'gpimentel79'. The repository has 1 pull request and various tabs for Code, Issues, Pull requests, Actions, Projects, Security, Insights, and Settings. The 'Code' tab is active, showing a file browser with folders 'clase2', 'clases\_actividades', 'img', and 'tkinter', and a file 'README.md'. The 'README.md' file is selected, and the 'Edit this file' button is visible. The file content is shown in a preview mode, displaying the title 'Programación II Politécnico Malvinas Argentinas'.

Name	Last commit message	Last commit date
clase2	agregando mensaje de cambio en repositorio remoto	5 days ago
clases_actividades	se agregó una variable	8 months ago
img	Mi primera versión de archivo de variables	8 months ago
tkinter	Mi primera versión de archivo de variables	8 months ago
README.md	Create README.md	2 minutes ago

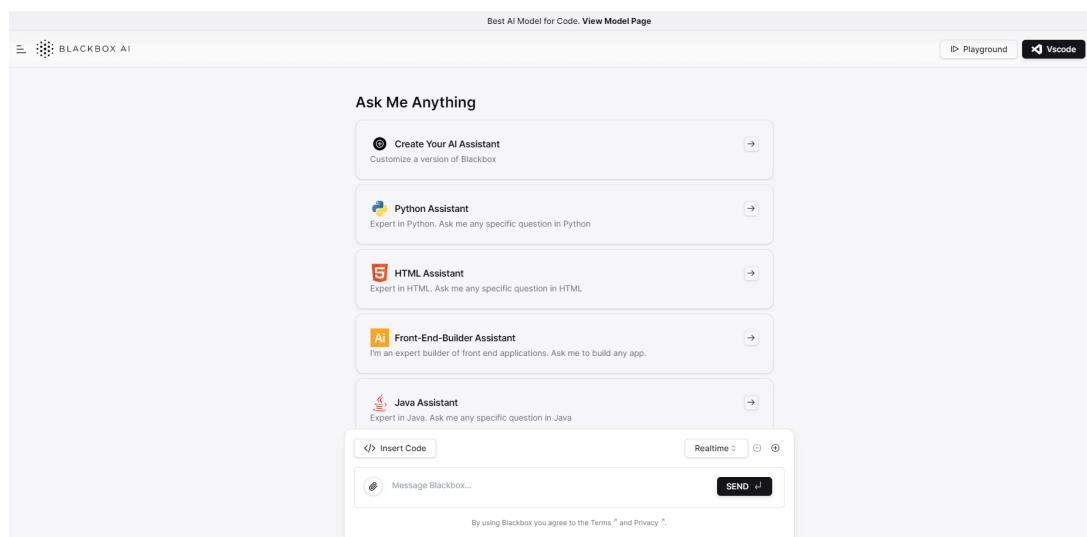
Programación II Politécnico Malvinas Argentinas

Los archivos 'readme' se escriben con formato "markdown". El formato "markdown" es un formato de escritura que permite la generación de una salida en HTML sin necesidad de aprender HTML. Editar un archivo 'readme' en formato "markdown" es realmente sencillo. Si nos fijamos, los títulos comienzan con #. Dependiendo del número de # que escribamos, tendremos un texto de nivel uno, un texto de nivel dos o un texto de nivel



tres, y podríamos llegar hasta el nivel seis. Los párrafos se escriben como cualquier otro texto y no necesitan de ningún símbolo especial. Lo único que es bueno saber es que el formato en negrita se indica con dos asteriscos al comienzo y al final de la palabra o frase que queramos marcar como negrita, la cursiva con solo un asterisco al comienzo y al final, y las negritas con cursiva con dos asteriscos y un guion bajo. E incluso también podemos utilizar listas ordenadas y no ordenadas. Las listas ordenadas se indican con números y las listas no ordenadas se indican con un asterisco. Existen multitud de signos que nos permitirán maquetar el archivo 'readme' a nuestro gusto. Te invito a que investigues todas las opciones disponibles y practiques creando archivos 'readme' para cada uno de tus repositorios. Para guardar las modificaciones realizadas en este archivo solo necesitamos anotar los cambios.

## **Blackbox AI: Un asistente de código impulsado por IA**

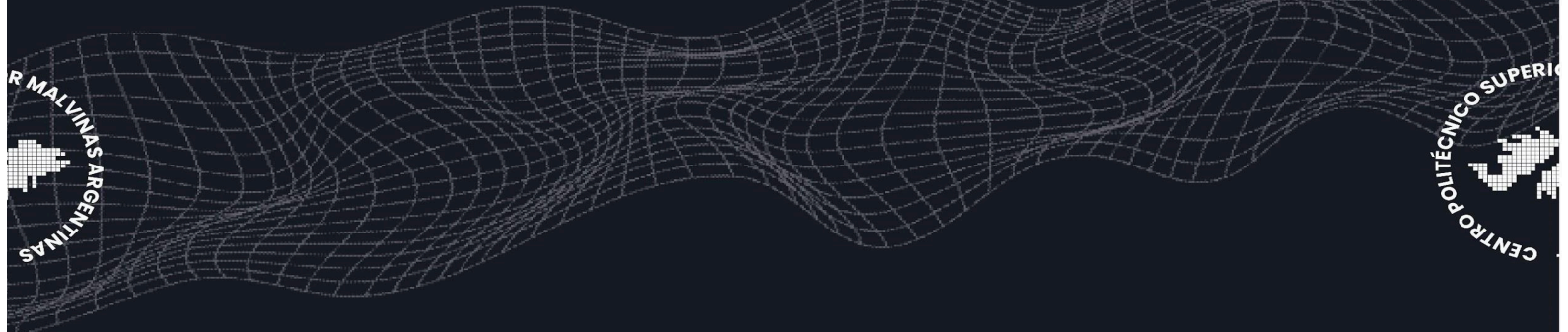


### **¿Qué es Blackbox AI?**

Blackbox AI es una extensión para Visual Studio Code que ofrece una variedad de funciones para ayudar a los programadores a escribir código más rápido y con menos errores. Algunas de las funciones más destacadas son:

- Autocompletado inteligente: Blackbox AI puede predecir qué código vas a escribir





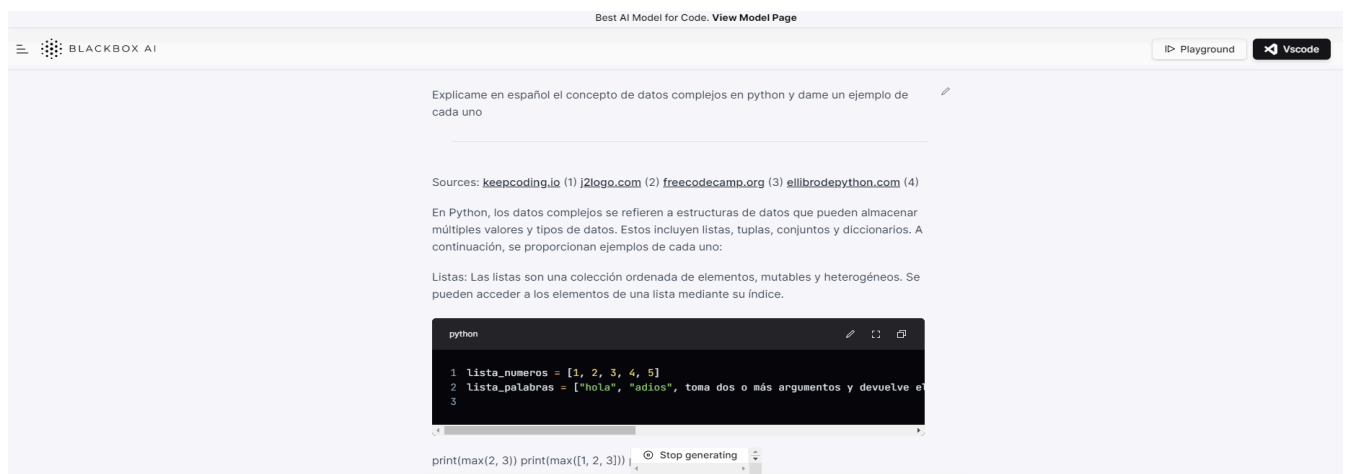
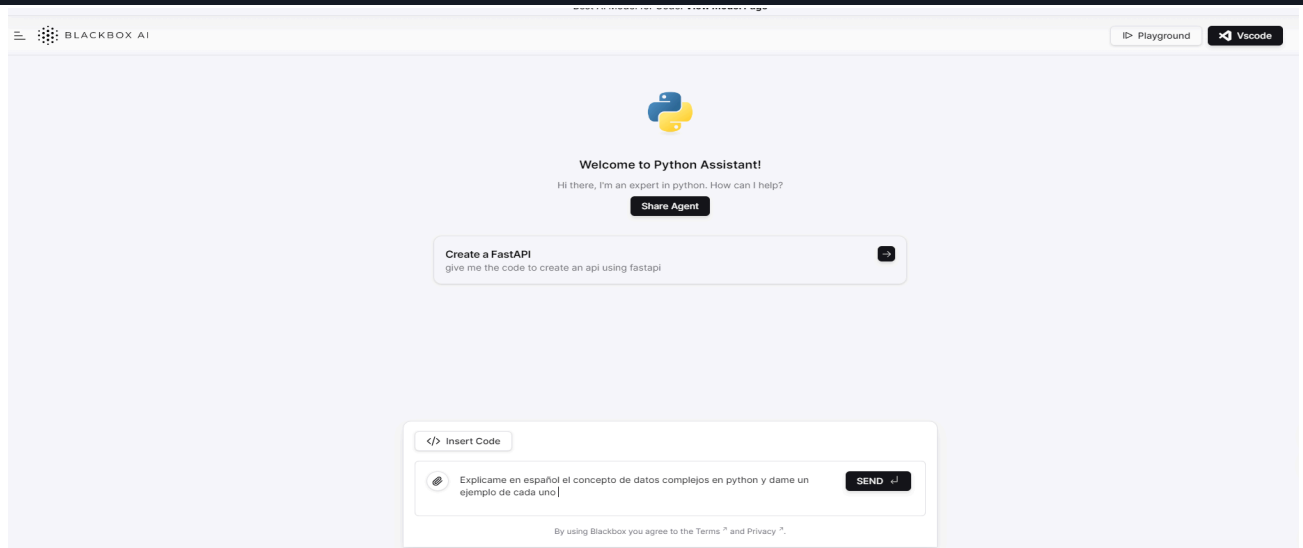
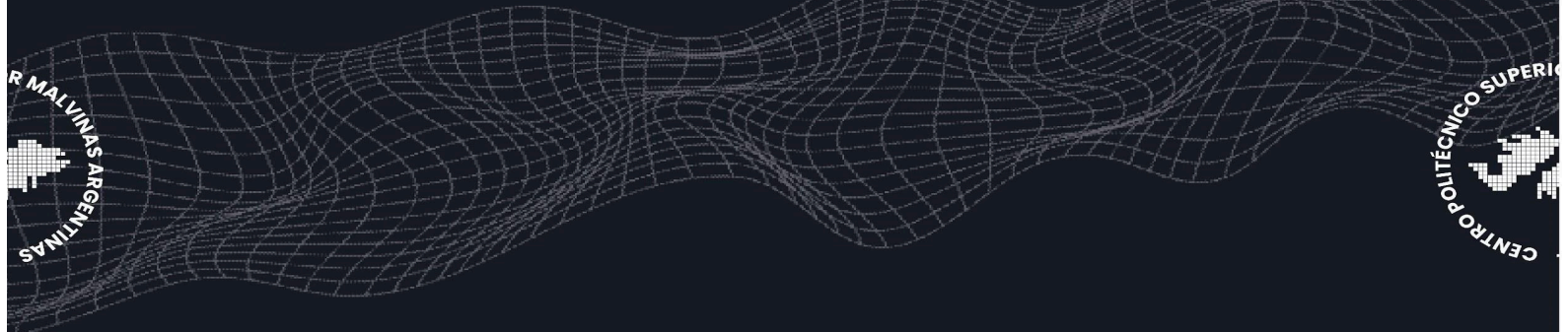
a continuación y ofrecerte sugerencias para completarlo automáticamente. Esto te ahorra tiempo y esfuerzo, y también te ayuda a evitar errores de sintaxis.

- Generación de código: Blackbox AI puede generar código completo a partir de una breve descripción de lo que quieres que haga. Esto es ideal para tareas repetitivas o para cuando no estás seguro de cómo implementar una función en particular.
- Búsqueda de código: Blackbox AI puede ayudarte a encontrar código similar al que estás escribiendo en una gran base de datos de código abierto. Esto es útil para aprender de otros programadores o para encontrar soluciones a problemas específicos.

### **¿Cómo usar Blackbox AI con Python?**

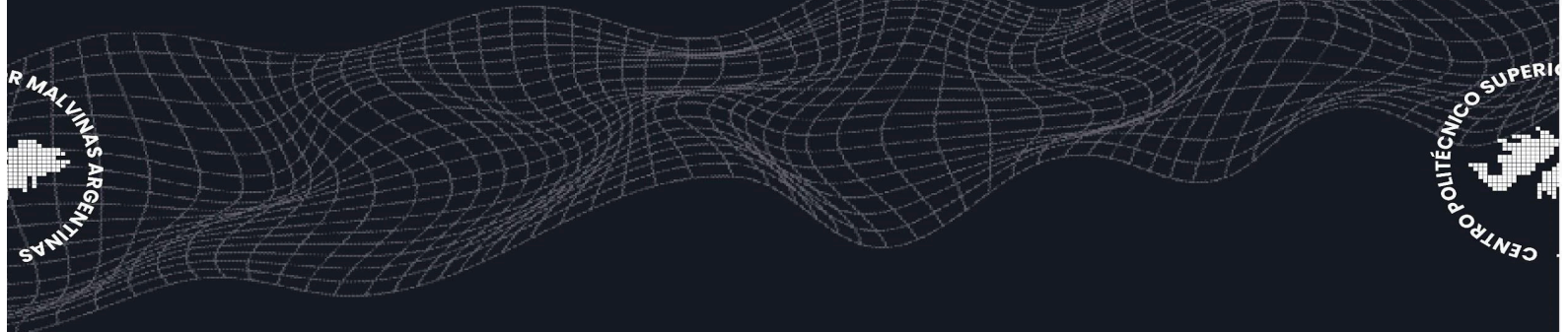
Blackbox AI es compatible con una gran variedad de lenguajes de programación, incluido Python. Para usar Blackbox AI con Python, simplemente debes instalar la extensión desde el Marketplace de Visual Studio Code.

Una vez que hayas instalado la extensión, puedes empezar a usarla de inmediato. Simplemente escribe el código que deseas completar y Blackbox AI te ofrecerá sugerencias. También puedes usar la función de generación de código para crear código a partir de una breve descripción.

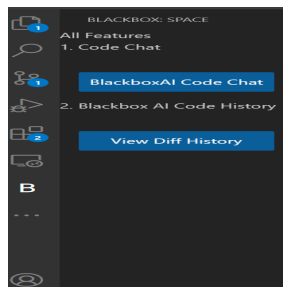


## ¿Cómo trabajar con Blackbox AI en Visual Studio Code?

Blackbox AI se integra perfectamente con Visual Studio Code. Vamos a instalar la extensiones de blackbox AI para VsCode.



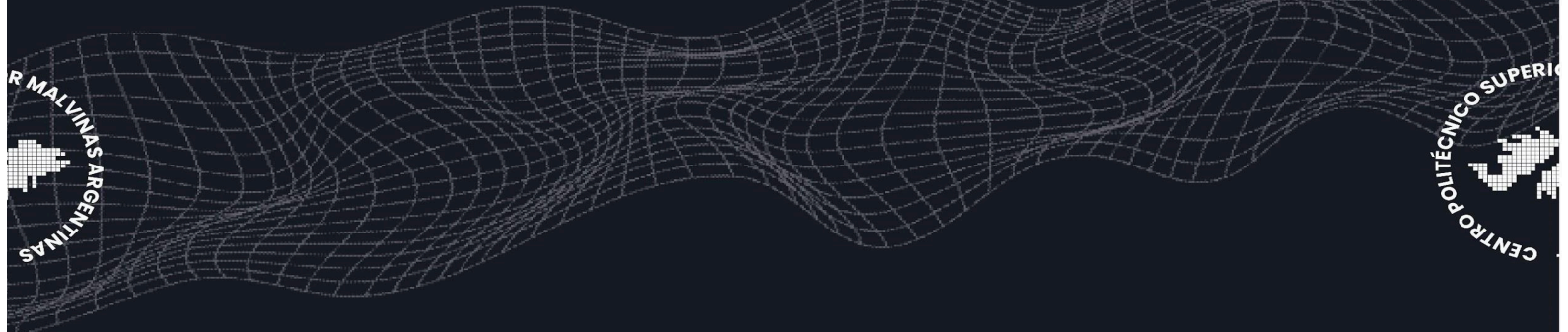
The screenshot shows the Visual Studio Code marketplace page for the 'Blackbox AI' extension. The page has a dark theme. At the top, the extension name 'Blackbox AI' is displayed with its icon, a star rating of 4.5 stars (168 reviews), and a download count of 2,179,436. Below this, a brief description states: 'BLACKBOX AI is an AI coding assistant that helps developers by providing real-time...'. A blue 'Instalando' button is visible. The main section features the title 'BLACKBOX AI — {Code} As Fast As You Think in VS Code' and a sub-header 'Write and Review Code with the help of our AI-powered Code Autocomplete and Code Chat. Seamlessly integrate BLACKBOX AI into your existing workflow to Supercharge your Coding Experience.' Below this, a 'Get Started' button is prominent. On the right side, there are sections for 'Categorías' (Programming Languages, Snippets, Machine Learning, Education) and 'Recursos' (Marketplace, Incidencias, Licencia, BLACKBOX AI).



Buscamos la extensión e instalamos la primera opción que nos aparece. Una vez instalado ya la podemos ver dentro de nuestras extensiones. Ya podemos habilitar alguna de las opciones por ejemplo habilitar el Code Chat que nos permitirá dentro de VsCode hacerle preguntas a la IA de blackbox sobre código de programación.

The screenshot shows the Blackbox AI chat interface. At the top, there is a toggle switch labeled 'Chat with Your Code Files'. Below this, a text input field contains the prompt 'dame un ejemplo sencillo de como utilizar un condicional en python'. To the right of the input field is a blue button with a white arrow icon. At the bottom, there is a disclaimer: 'By using Blackbox you agree to the Terms <sup>↗</sup> and Privacy <sup>↗</sup>.'





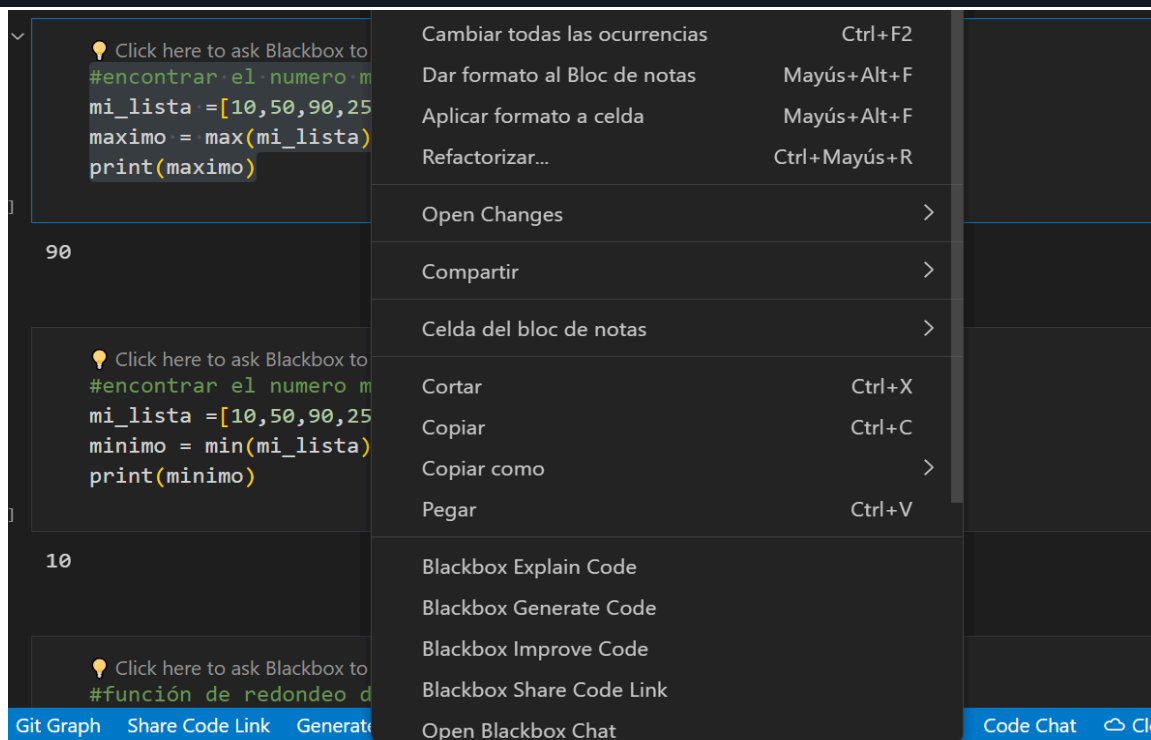
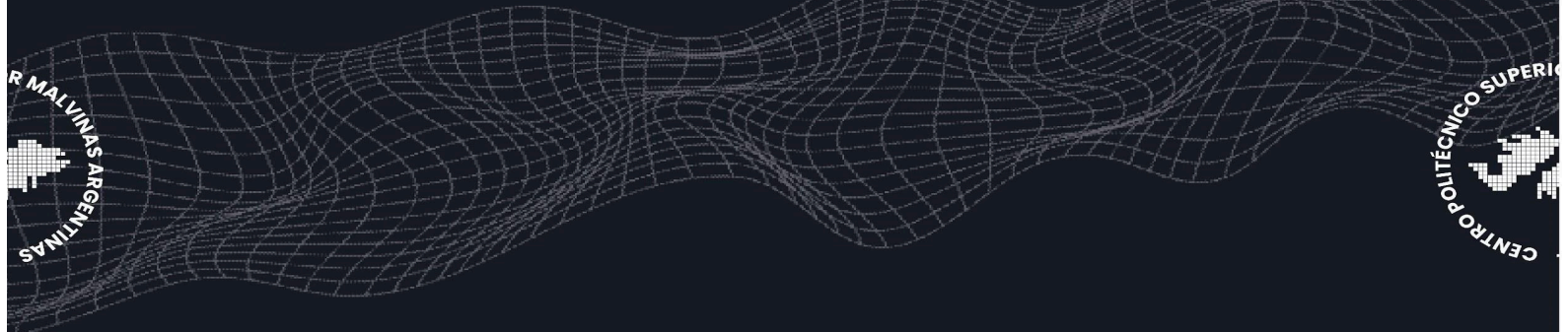
```
1 # Python program to illustrate if statement
2
3 x = 10
4 y = 20
5
6 if x < y:
7     print("x es menor que y")
8 else:
9     print("x es mayor o igual que y")
```

En este ejemplo, la sentencia `if` comprueba si el valor de `x` es menor que el

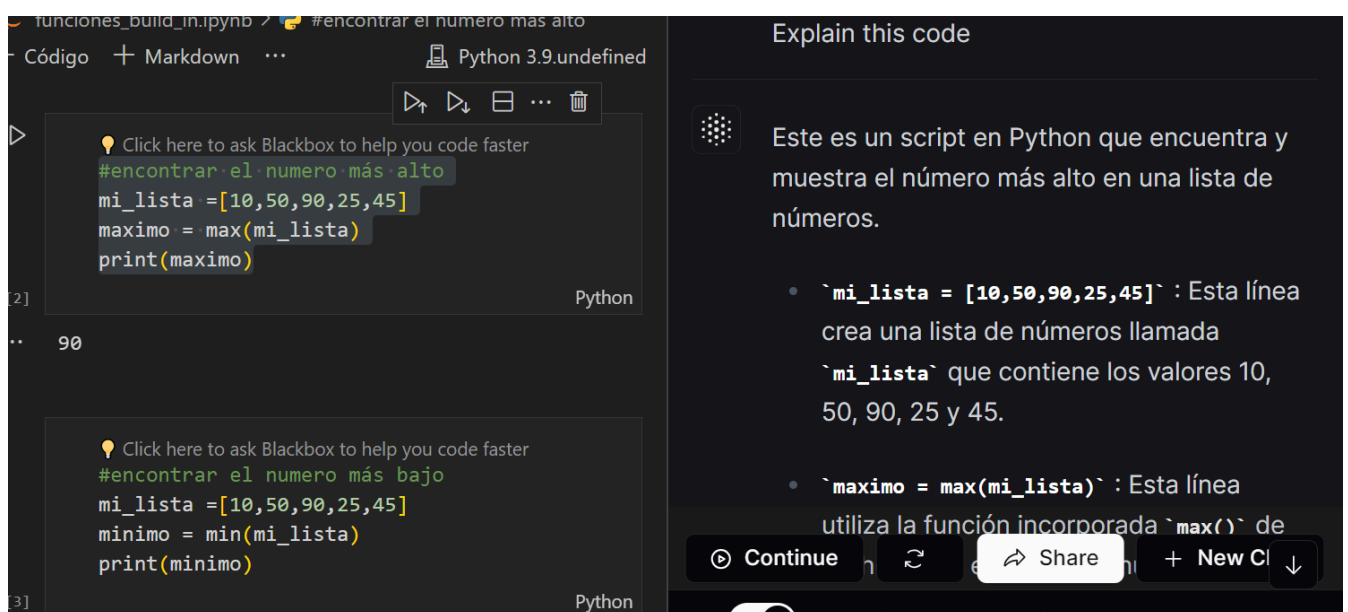
De esta manera vemos cómo a través del **code chat** podemos realizar consultas sobre código de programación en python y en otros lenguajes.

## Explain Code

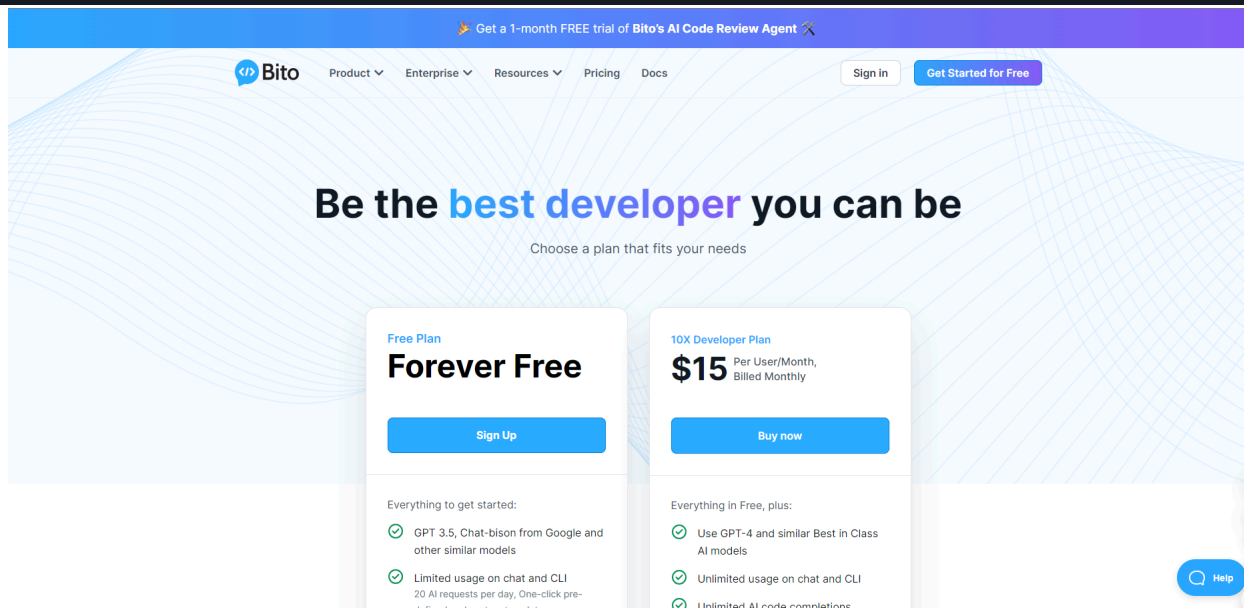
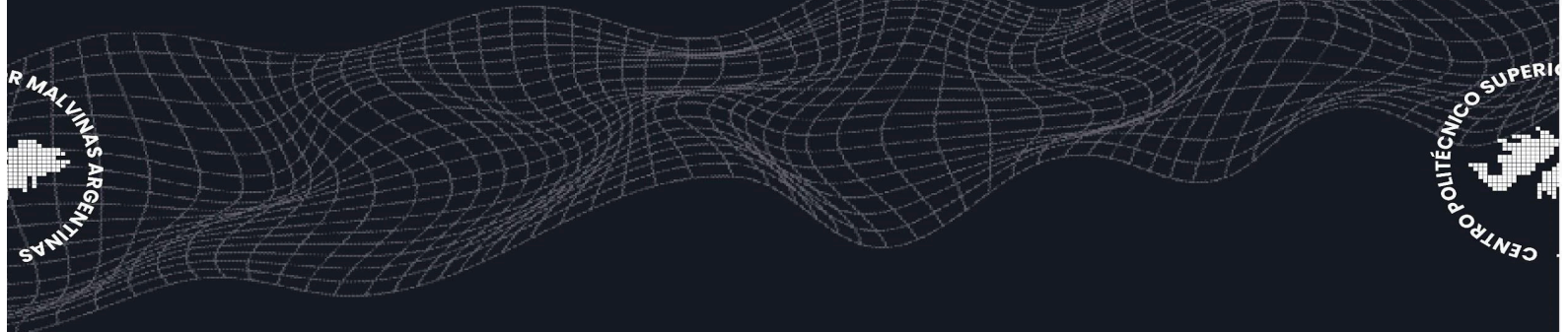
Otra opción interesante que tiene BlackBox AI es la posibilidad de explicar un código ya escrito.



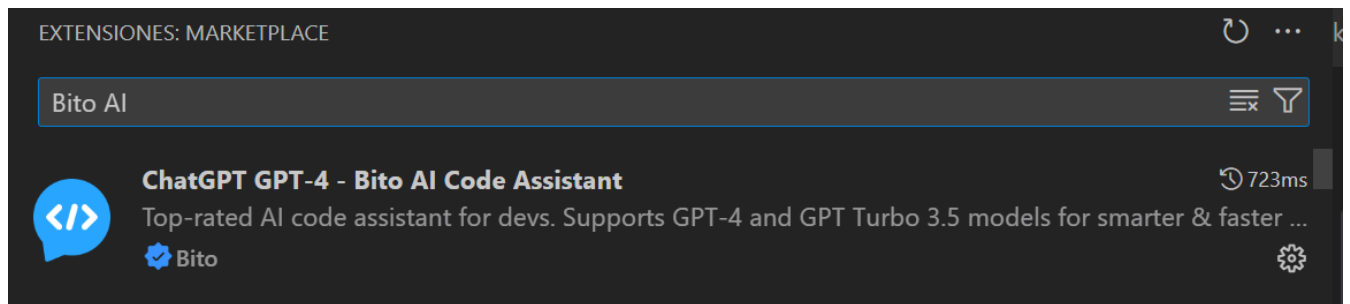
Abrimos un archivo de código y con el botón derecho del mouse vamos a ver las opciones que nos brinda BlackBox, entre ellas se encuentra **Explain code** (explicar el código)



## **Bit Code: Un asistente de código impulsado por IA**

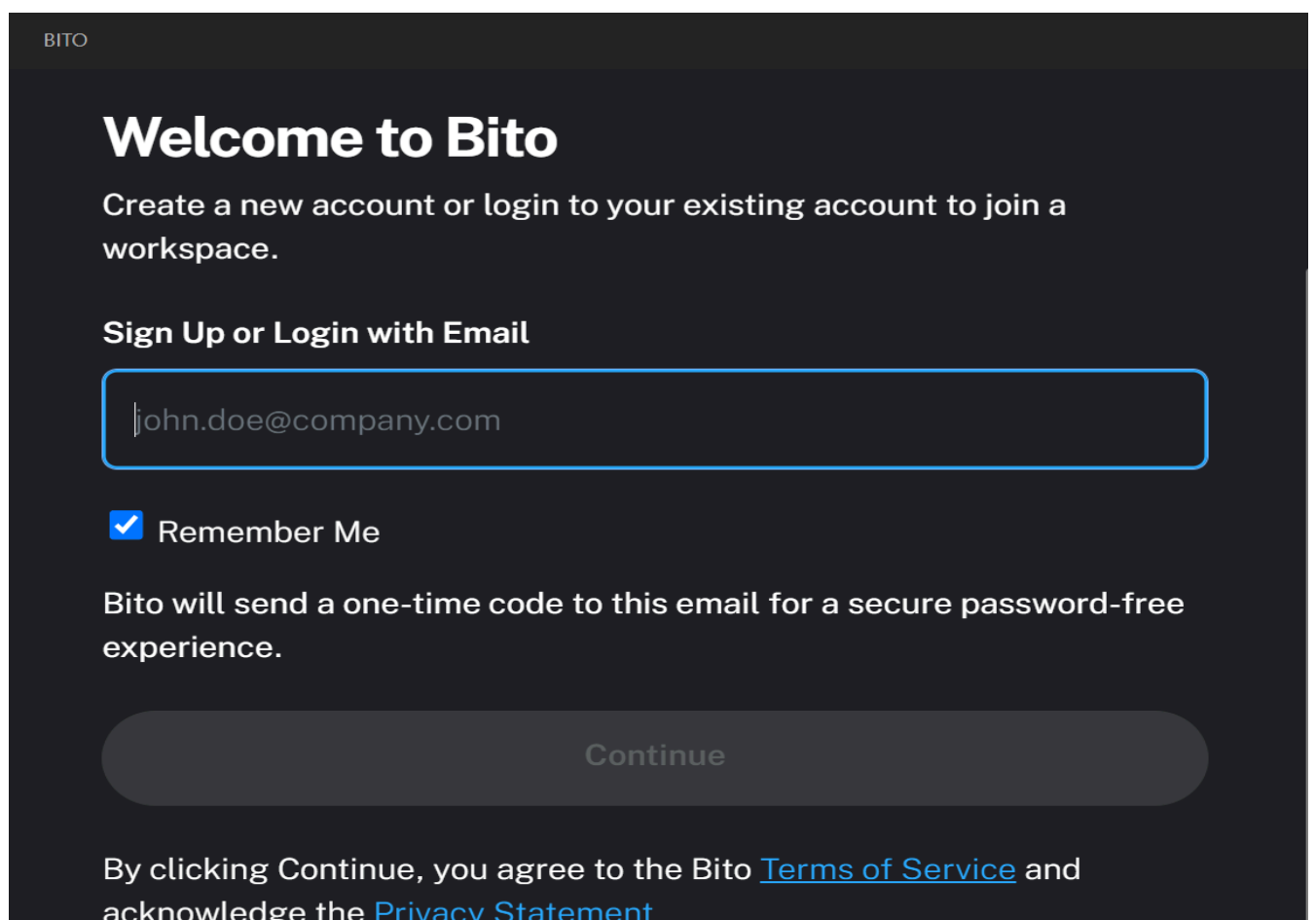
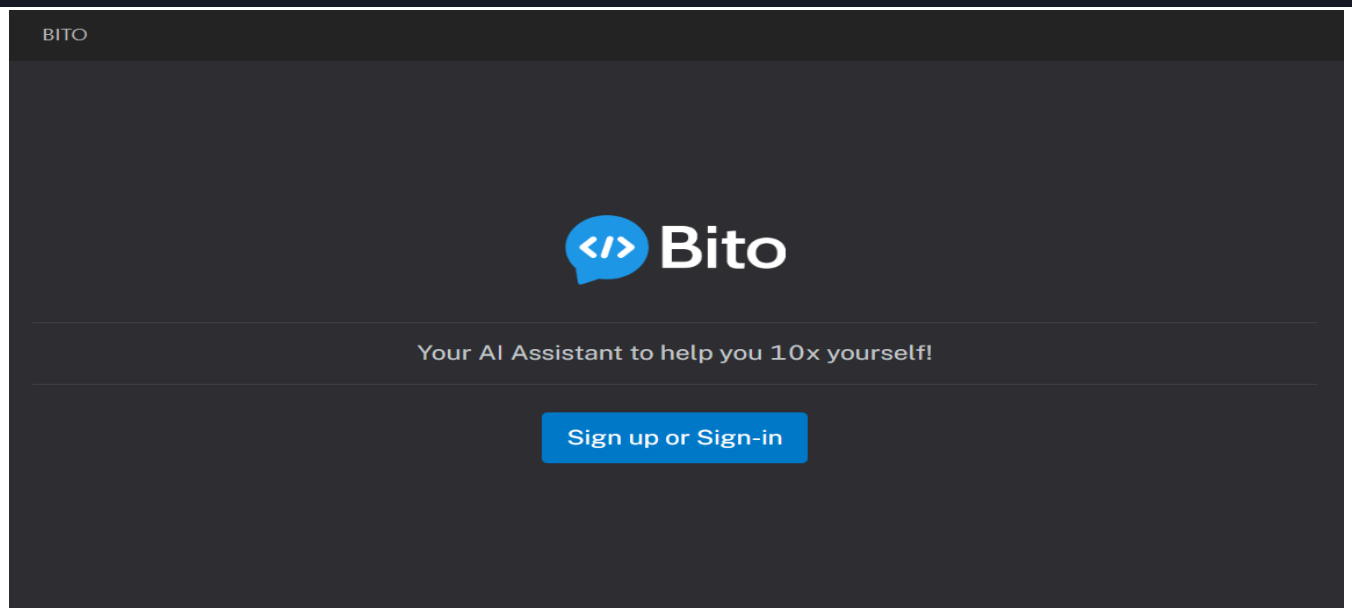
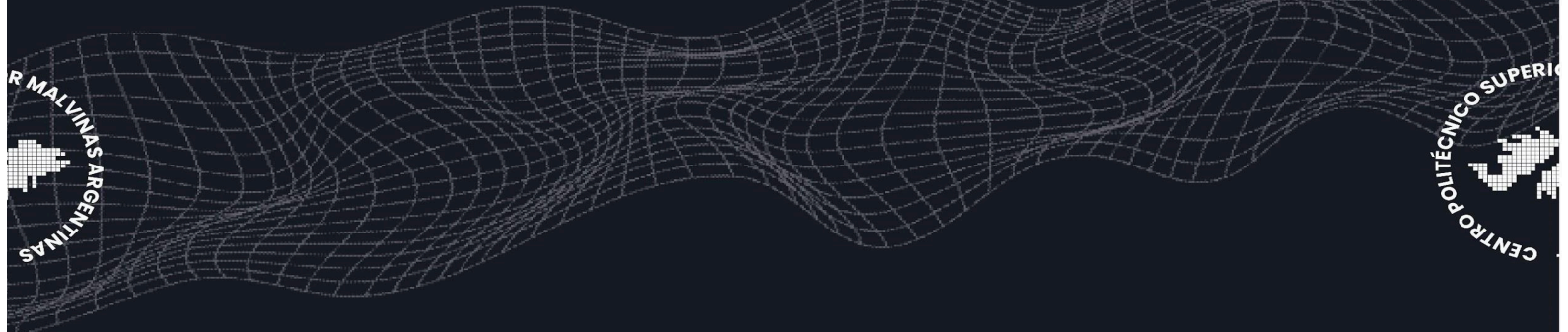


Para poder trabajar con Bito desde VsCode vamos a la opción de extensiones y en la campo de búsqueda ingresamos **Bito AI** la primera opción que nos aparece, le damos a instalar.



Una vez instalado vamos a la opción de Bito Code y se nos abrirá la ventana para que comencemos, para utilizarlo debemos registrar una cuenta de correo electrónico.





Una vez ingresado el mail, nos enviarán a nuestra casilla, un código de 6 dígitos que


debemos ingresar para que se nos habilite el chat de Bito

Please check your junk, spam, and promotions folders if you don't see it.

The code expires shortly, so please enter it soon.

Una vez ingresado el código que nos llegó a nuestra casilla de correo, Bito nos permitirá acceder a nuestro espacio de trabajo si ya hemos creado o crear un espacio de trabajo

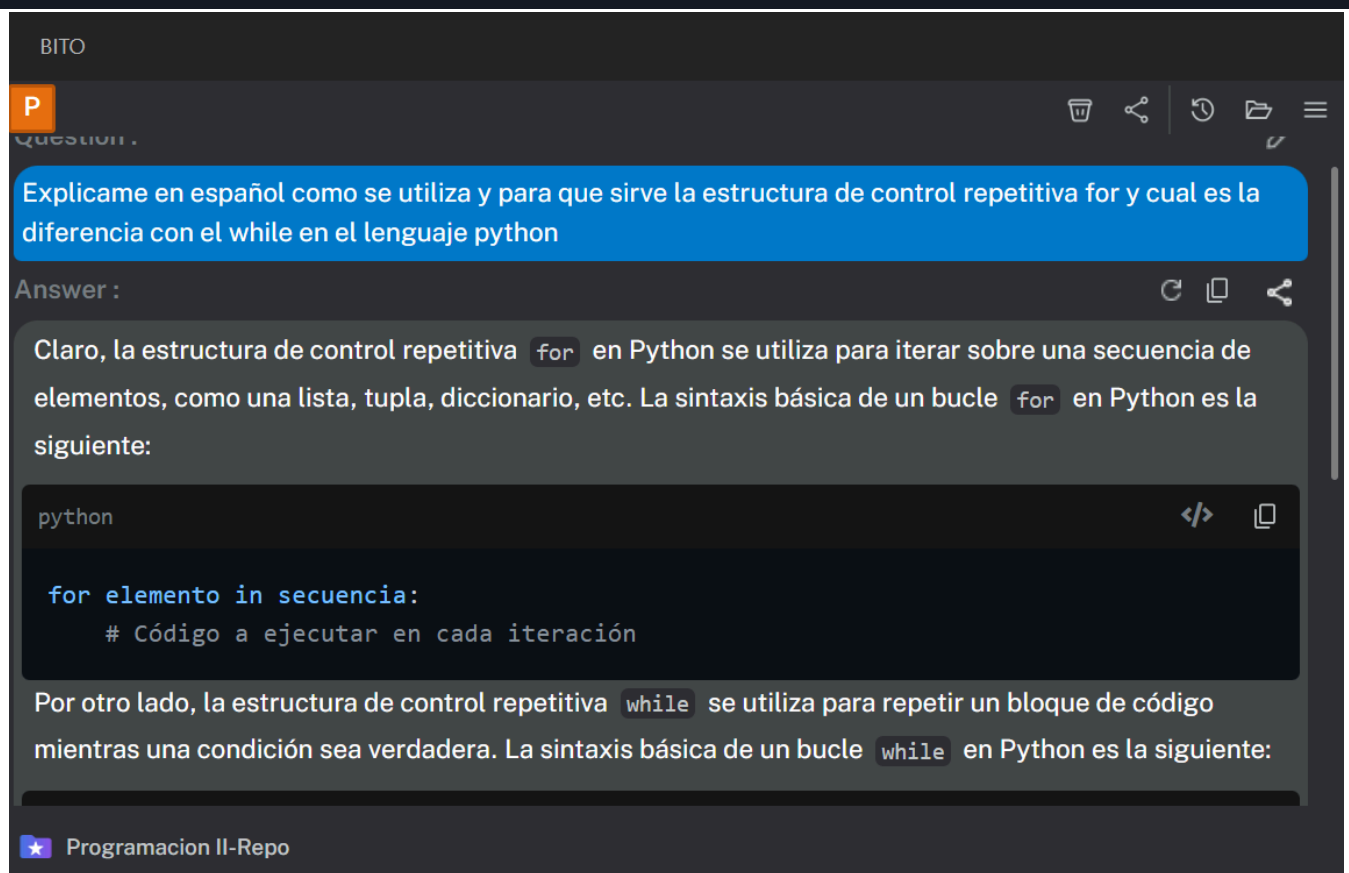
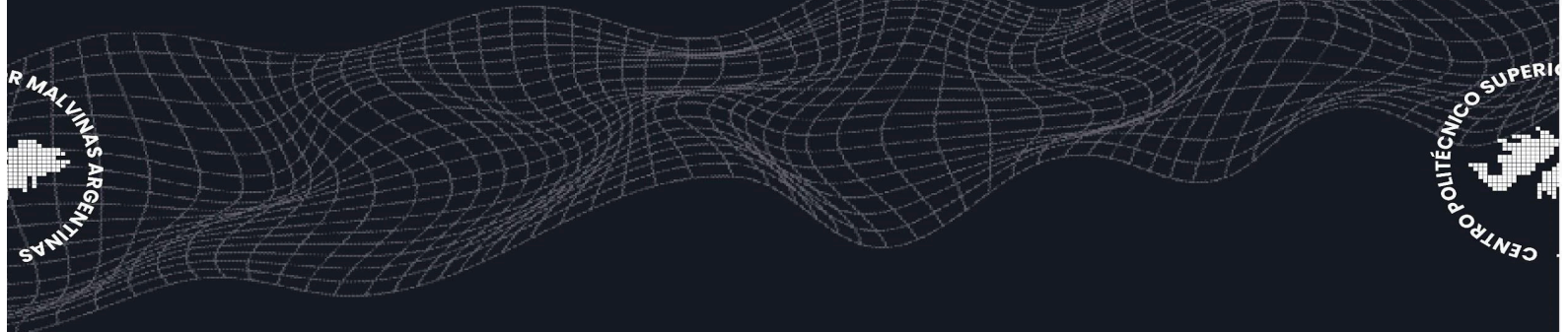
## Sign into your workspaces



**Python\_dev**  
Python\_dev

Sign in

Hecho esto, ya estamos listo para interactuar con Bito, haciéndole preguntas. Bito no trabaja en tiempo real en su versión libre. Pero es una opción válida que nos permite realizar preguntas y mejorar nuestro aprendizaje apoyándonos en una herramienta de inteligencia artificial 😊.



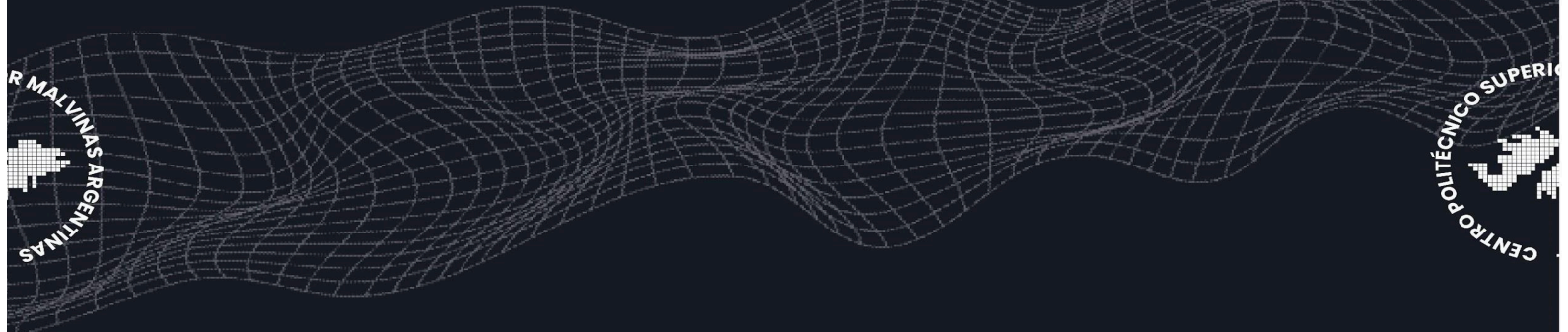
La inteligencia artificial (IA) está revolucionando el mundo, y la educación no es una excepción. Herramientas como Bito Code y Blackbox AI son ejemplos de cómo la IA puede ser un **asistente facilitador del aprendizaje** en la programación.

Sin embargo, es importante recordar que la IA no debe reemplazar nuestro propio **esfuerzo y dedicación**. No podemos delegar nuestro conocimiento a la IA. Debemos usarla como una herramienta para potenciar nuestro aprendizaje, no como un sustituto del mismo.

### ¿Cómo podemos usar la IA de forma responsable en el aprendizaje?

- **Ser conscientes de las limitaciones de la IA:** La IA no es perfecta. Puede cometer errores o proporcionar información incompleta. Es importante que siempre **verifiquemos y validemos** la información que obtenemos de la IA.





- **Utilizar la IA para complementar nuestro aprendizaje:** La IA puede ayudarnos a comprender conceptos complejos, encontrar soluciones a problemas y practicar nuestras habilidades. Sin embargo, no debemos depender de la IA para aprender todo. Es importante que también **realicemos nuestro propio trabajo** de investigación, estudio y práctica.
- **Desarrollar nuestro pensamiento crítico:** La IA puede ofrecernos información y sugerencias, pero es importante que **pongamos por nosotros mismos** y que no aceptemos ciegamente todo lo que la IA nos dice.
- **Ser responsables con la información que compartimos:** La IA aprende de la información que le proporcionamos. Es importante que seamos **cuidadosos con la información que compartimos** con la IA para asegurarnos de que está aprendiendo de información de calidad.

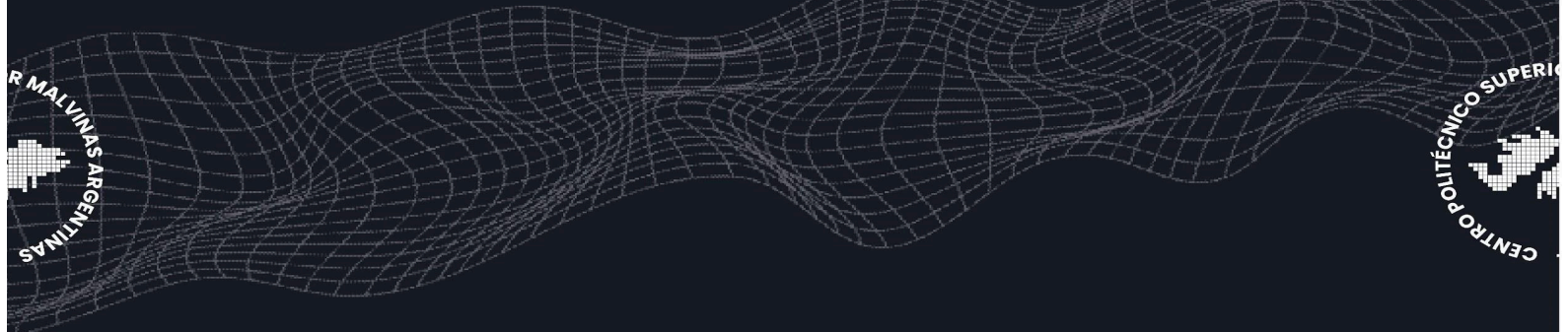
**En resumen, la IA puede ser una herramienta poderosa para el aprendizaje, pero debemos usarla de forma responsable.** No debemos delegar nuestro conocimiento a la IA, sino que debemos usarla como un **asistente para potenciar nuestro propio aprendizaje.**

### **Actividad:**

**Objetivo:** Practicar el uso del autocompletado y la generación de código con Blackbox AI y Bito Code en Python.

Instrucciones:

1. Abrir Visual Studio Code.
2. Crear un nuevo archivo Python.
3. Escribir un código simple, como una función para calcular el promedio de una



lista de números.

4. Observa cómo Blackbox AI y Bito Code te ayudan a en el desarrollo del código de manera rápida y precisa.
5. Prueben utilizar la función de generación de código para que Blackbox les escriba el código completo.
6. Comparen el código generado por la herramienta con el código que habrían escrito ustedes mismos.
7. Reflexionen sobre las ventajas y desventajas de usar la IA para generar código.

### **Instancia de Autoevaluación:**

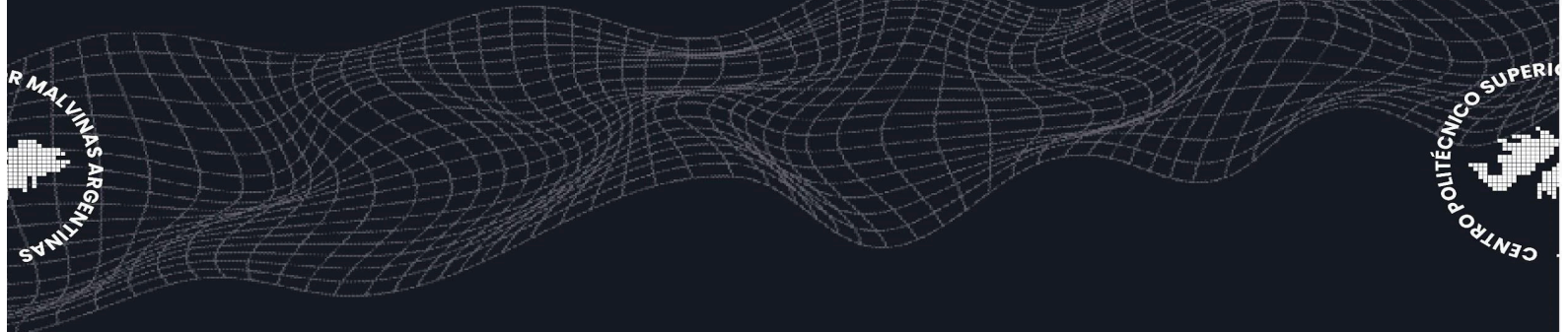
En esta instancia de autoevaluación te pedimos que puedas poner en juego lo que has aprendido hasta este momento. Recordá que siempre podrás volver a la teoría presentada en esta clase o hacer las preguntas pertinentes en los encuentros sincrónicos o las tutorías presenciales.

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:**

Hemos llegado al final de este recorrido por la clase IV de Programación II 🎓. Durante esta clase, exploramos temas fundamentales como la gestión y utilización de ramas en Git, la importancia de mantener repositorios organizados y la potenciación del desarrollo



en Python con extensiones de inteligencia artificial en Visual Studio Code. Al comprender la integración de Git, GitHub y Visual Studio Code, hemos fortalecido nuestras habilidades en colaboración y sincronización de repositorios, así como en la resolución de conflictos en proyectos colaborativos.

Recordemos que la práctica constante es esencial para afianzar lo aprendido. Los invito a seguir explorando las posibilidades que ofrece la inteligencia artificial en el desarrollo de software en Python, así como a aplicar eficientemente las técnicas avanzadas de gestión de ramas para mantener un flujo de trabajo organizado y eficiente.

En la próxima clase, continuaremos nuestro viaje en el mundo de la programación, ¡Prepárense para seguir ampliando sus conocimientos y habilidades en programación!

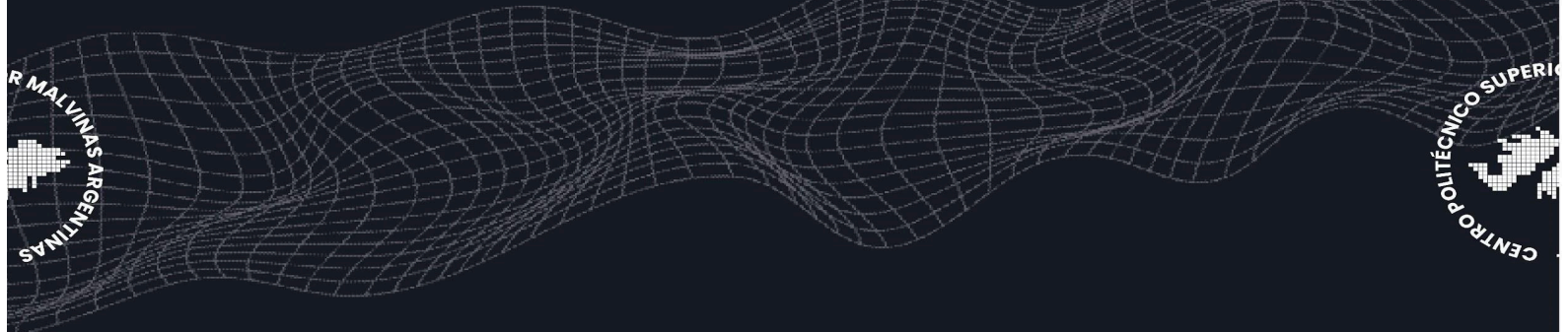


### **Bibliografía Obligatoria:**

- 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>