



GIT

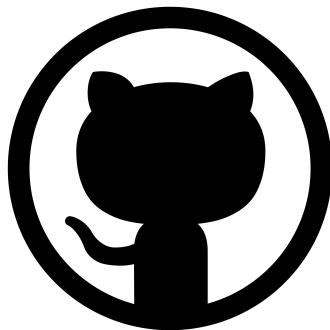
¿Qué es GIT?

- Es un sistema de control de versiones open source.
- Es distribuido, es decir que múltiples personas pueden trabajar en equipo a la vez.
- Se adapta a todo tipo de proyectos desde pequeños hasta grandes
- Se pueden fusionar archivos, guarda una línea de tiempo a lo largo de todo el proyecto.
- Maneja una interfaz por consola
- GIT es el software de control de versiones en el que se basa GitHub.



¿Qué es GitHub?

- A diferencia de Git, GitHub es un sitio web y un servicio en la nube que ayuda a los desarrolladores a almacenar y administrar su código.
- Lleva un registro y control de cualquier cambio sobre este código.
- En otras palabras, es una plataforma de desarrollo colaborativo donde se alojan los repositorios.
- El código se almacena de forma pública o privada.





Recursos

Sitio de descarga de GIT: <https://git-scm.com/downloads> (Descargar e instalar)

Documentación de Git: <https://git-scm.com/book/es/v2>

Sitio de GitHub: <https://github.com/> (Crear una cuenta)

Documentación de GitHub: <https://docs.github.com/es/get-started>



Ejercicios por consola (CMD)

- Antes de empezar a usar git es necesario configurarlo con el nombre de usuario y su correo electrónico:
 - `git config --global user.name "tu nombre completo"`
 - `git config --global user.email "tu email usado en github"`
- Mostrar la configuración
 - `git config --list`



Visual Studio Code y Git

Lo primero que debe hacer para aprovechar la integración del Control de código fuente es iniciar un proyecto como repositorio Git.

Abra Visual Studio Code y acceda al terminal integrado. Puede abrirlo usando el atajo de teclado ```CTRL + ``` en Linux, macOS o Windows.

En su terminal, cree un directorio para un nuevo proyecto y cambie a ese directorio:

- `mkdir git_test`
- `cd git_test`

A continuación, cree un repositorio Git:

- `git init`



Visual Studio y Git

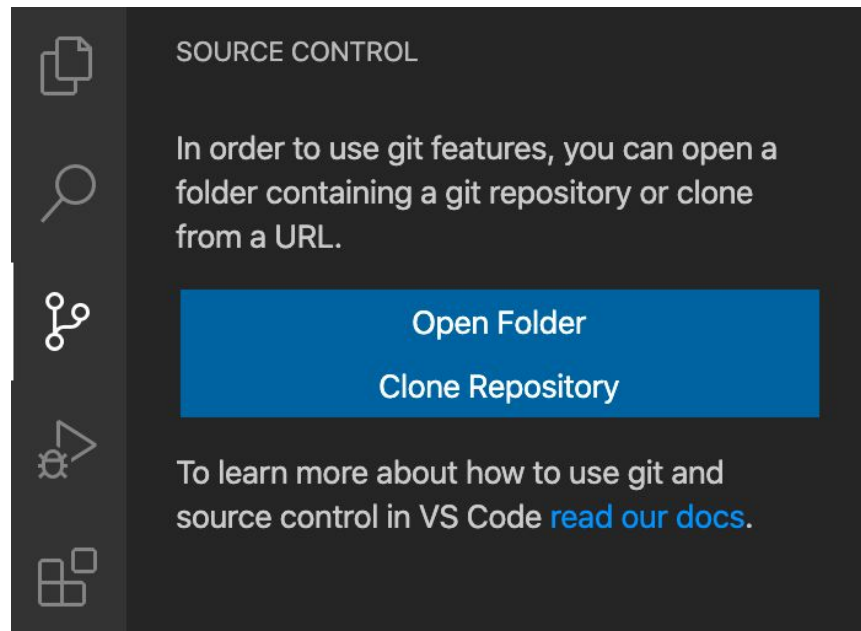
Otra forma de conseguir esto con Visual Studio Code es abriendo la pestaña Control de código fuente (el icono parece una brecha en la carretera) en el panel izquierdo:





Visual Studio y Git

A continuación, seleccione Abrir carpeta:

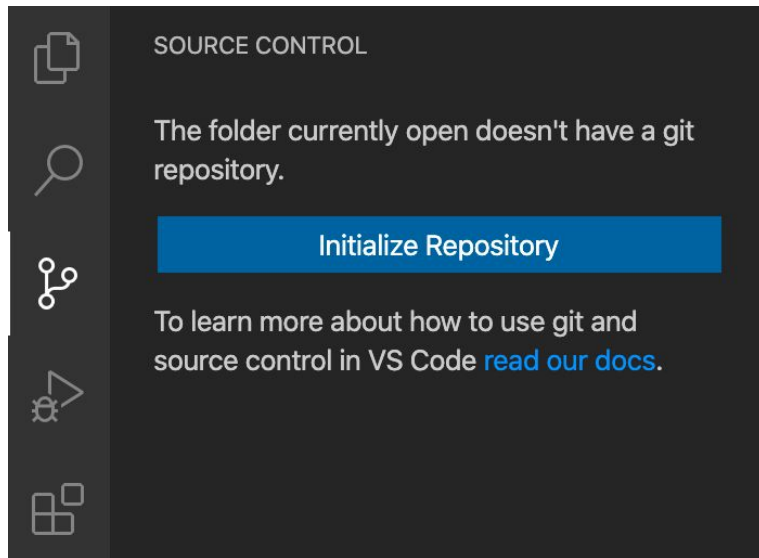




Visual Studio y Git

Esto abrirá su explorador de archivos al directorio actual. Seleccione el directorio de proyecto preferido y haga clic en Abrir.

A continuación, seleccione Iniciar repositorio:





Visual Studio y Git

Si ahora comprueba su sistema de archivos, verá que incluye un directorio `.git`. Para hacer esto, use el terminal para navegar al directorio de su proyecto y liste todo su contenido:

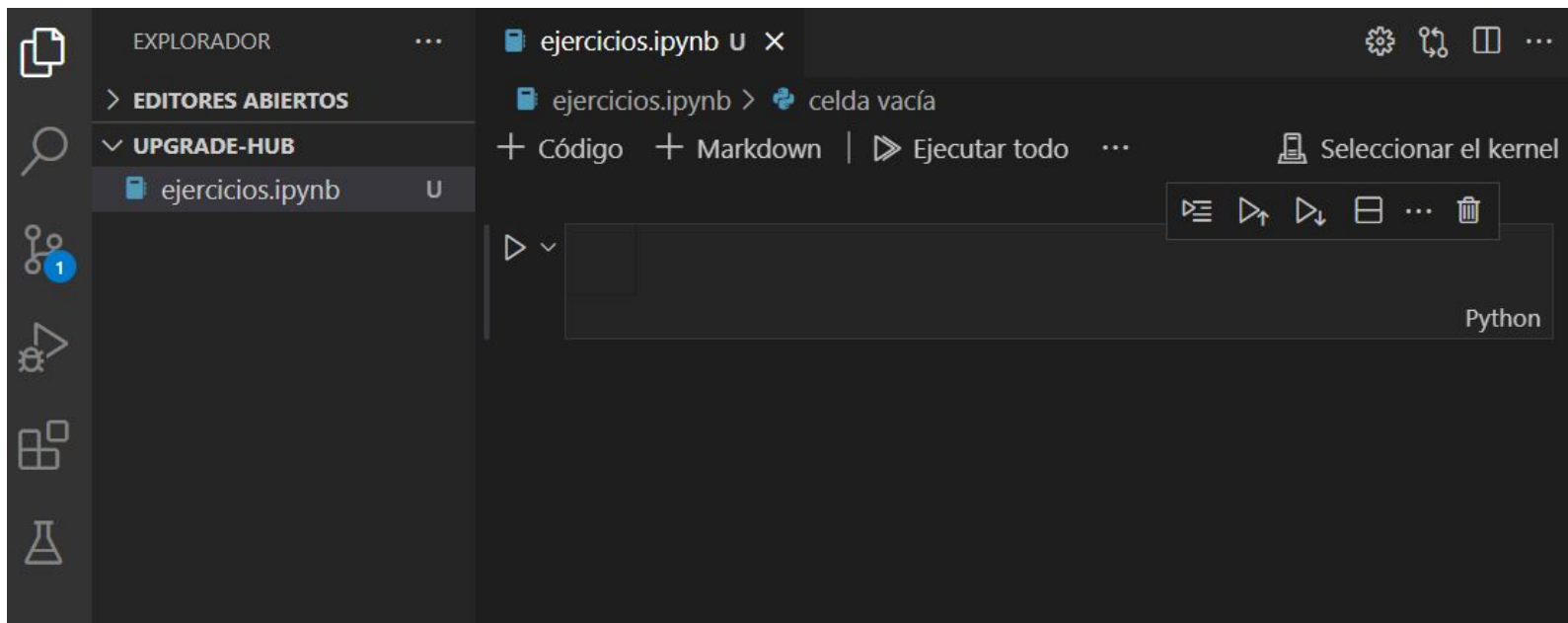
- `ls -la`

Verá el directorio `.git` que se creó.

Ahora que se ha iniciado el repositorio, añada un archivo llamado “ejercicios.ipynb” (sin comillas).

Tras hacer eso, verá en el panel Control de código fuente que su nuevo archivo se muestra con la letra U junto a él. U significa archivo sin seguimiento (untracked file), lo que significa que es nuevo o se ha cambiado, pero aún no ha sido añadido al repositorio.

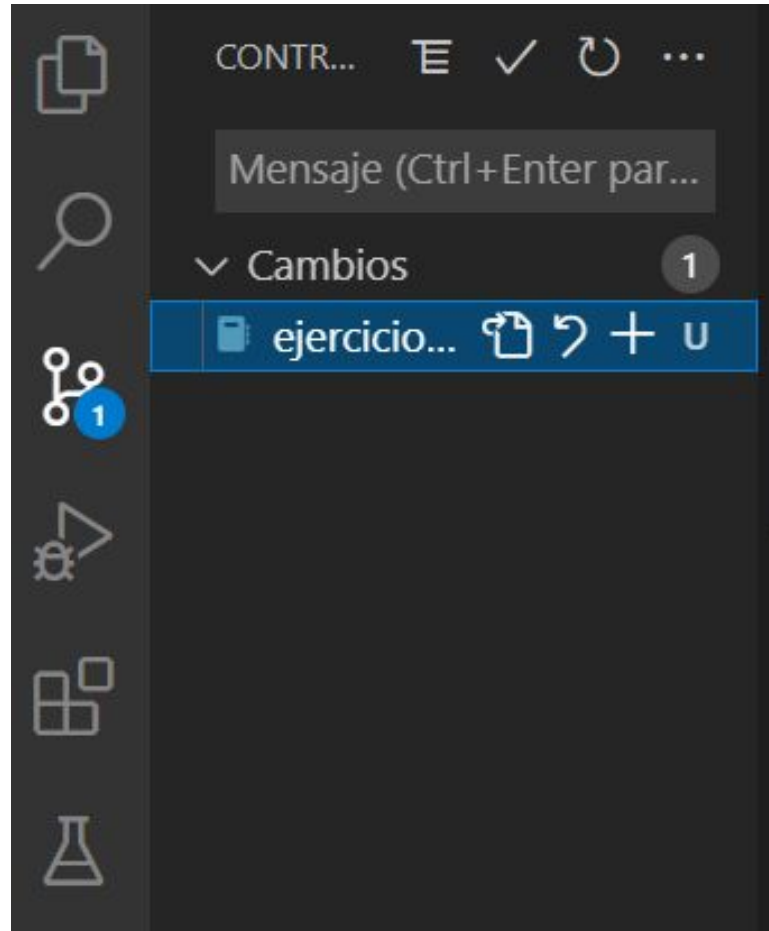
Visual Studio y Git



Visual Studio y Git

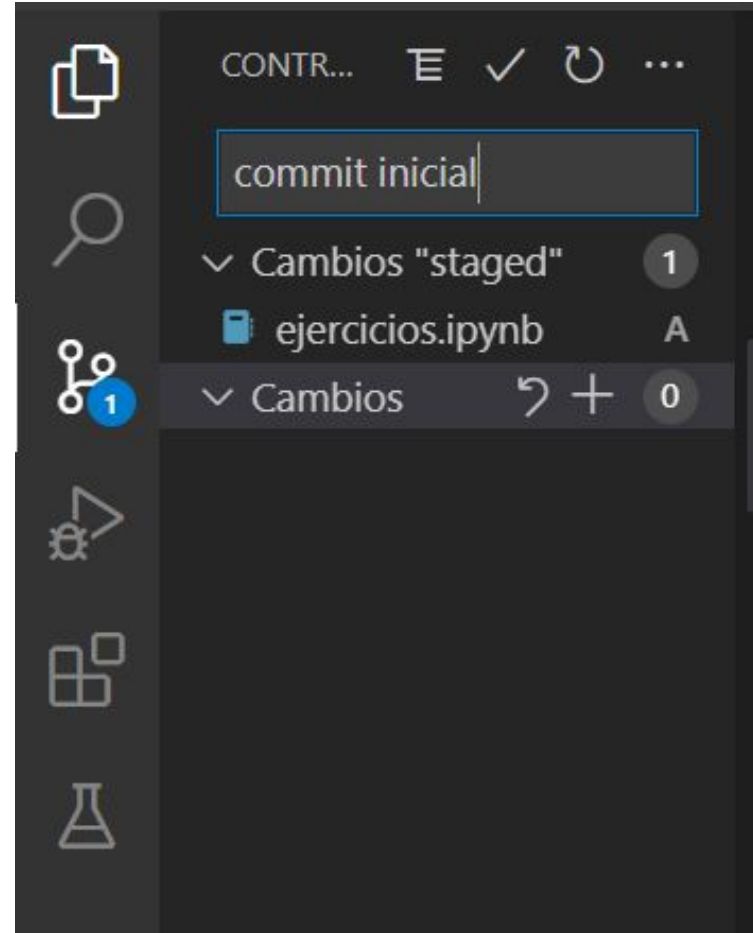
Ahora puede hacer clic en el icono plus (+) junto al listado de archivos para que el repositorio realice un seguimiento del archivo.

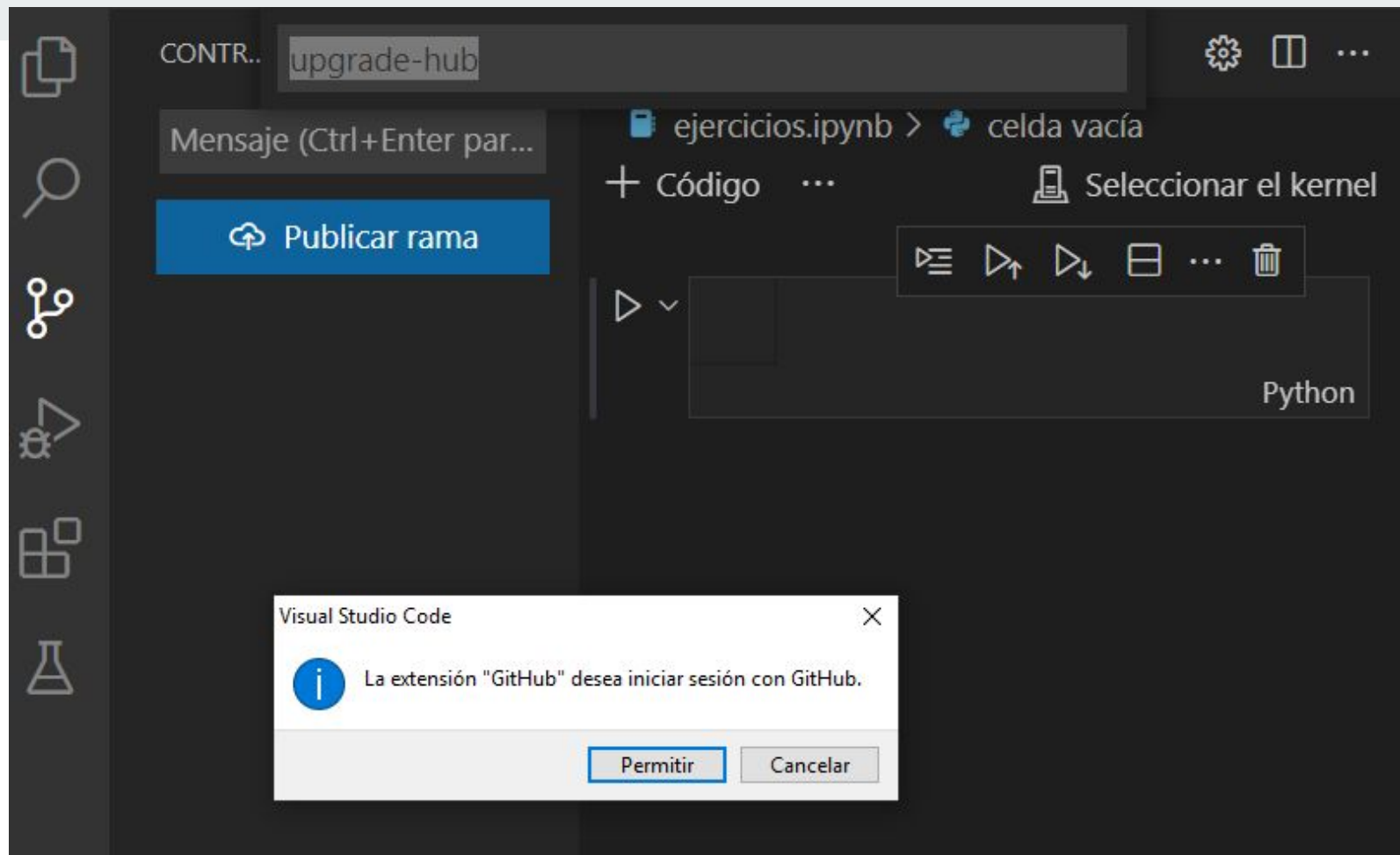
Una vez añadido, la letra junto al archivo cambiará a una A. A indica un nuevo archivo que se añadió al repositorio.



Visual Studio y Git

Para confirmar sus cambios, escriba un mensaje commit en el cuadro de entrada en la parte superior del panel Control de código fuente. A continuación, haga clic en el icono check para realizar la confirmación.







Sign in to GitHub
to continue to GitHub for VS Code

Username or email address

Password

[Forgot password?](#)

Sign in

New to GitHub? [Create an account.](#)



Authorize GitHub for VS Code



GitHub for VS Code by [Visual-Studio-Code](#)

wants to access your jalagrod account



Repositories

Public and private



Workflow

Update GitHub Action Workflow files.



Cancel

Authorize Visual-Studio-
Code

Authorizing will redirect to
<https://vscode.dev>



Owned & operated
by GitHub



Created 4 years ago



More than 1K
GitHub users

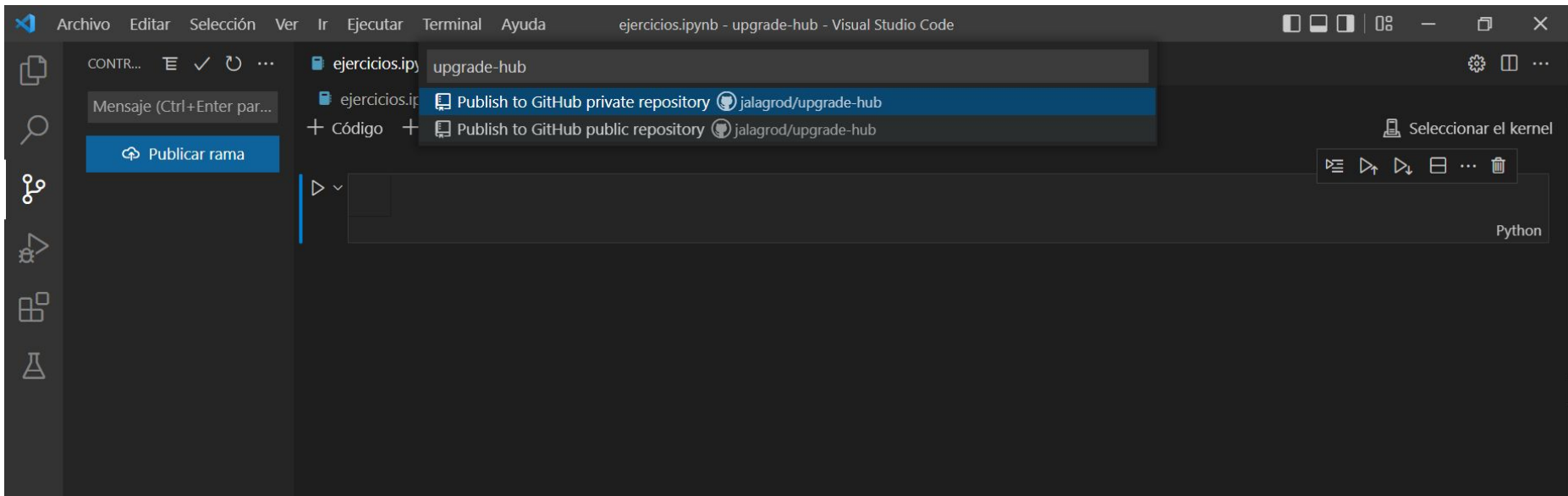
¿Abrir Visual Studio Code?


<https://vscode.dev> quiere abrir esta aplicación.


☐ Permitir siempre que vscode.dev abra este tipo de enlaces en la aplicación asociada

Abrir Visual Studio Code

Cancelar





 El repositorio "jalagrod/upgrade-hub" se ha publicado correctamente en GitHub.



Origen: GitHub (extensión).

[Abrir en GitHub](#)

[Pull requests](#)[Issues](#)[Marketplace](#)[Explore](#)

[jalagrod / upgrade-hub](#) Private

Unwatch **1**

Fork **0**

Star **0**

[Code](#)[Issues](#)[Pull requests](#)[Actions](#)[Projects](#)[Security](#)[Insights](#)[Settings](#)

master

1 branch

0 tags

[Go to file](#)[Add file](#)[Code](#)[About](#)

Jose commit inicial

f0f9259 6 minutes ago 1 commit



ejercicios.ipynb

commit inicial

6 minutes ago

Add a README with an overview of your project.

[Add a README](#)

No description, website, or topics provided.

0 stars

1 watching

0 forks

[Releases](#)

No releases published

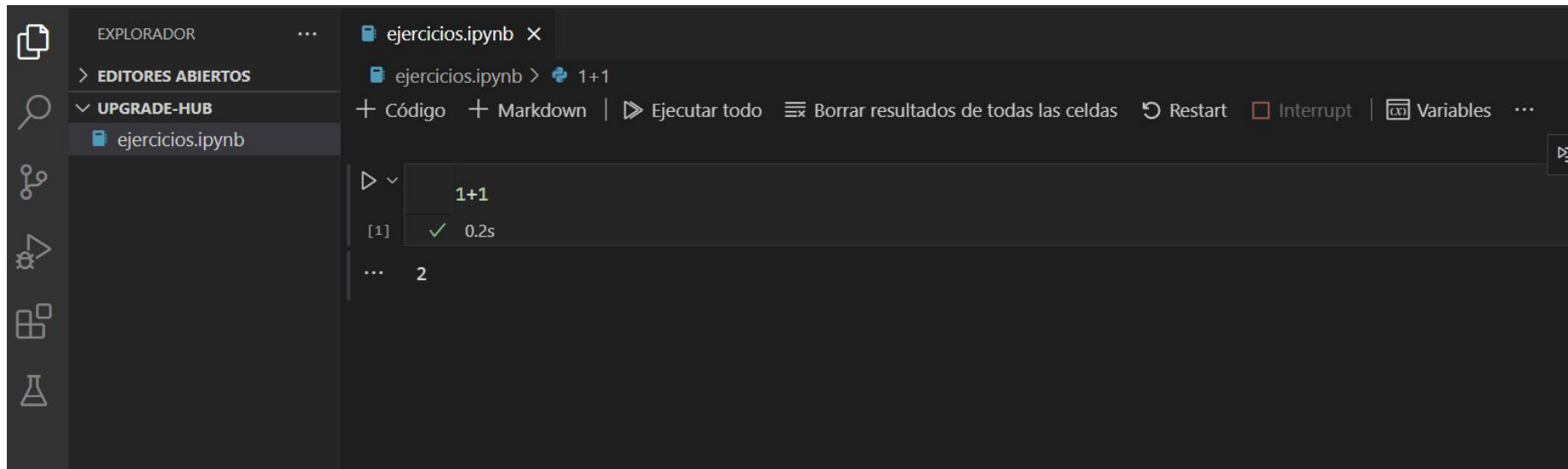
[Create a new release](#)[Packages](#)

No packages published

[Publish your first package](#)[Languages](#)

Jupyter Notebook 100.0%

Edite el archivo

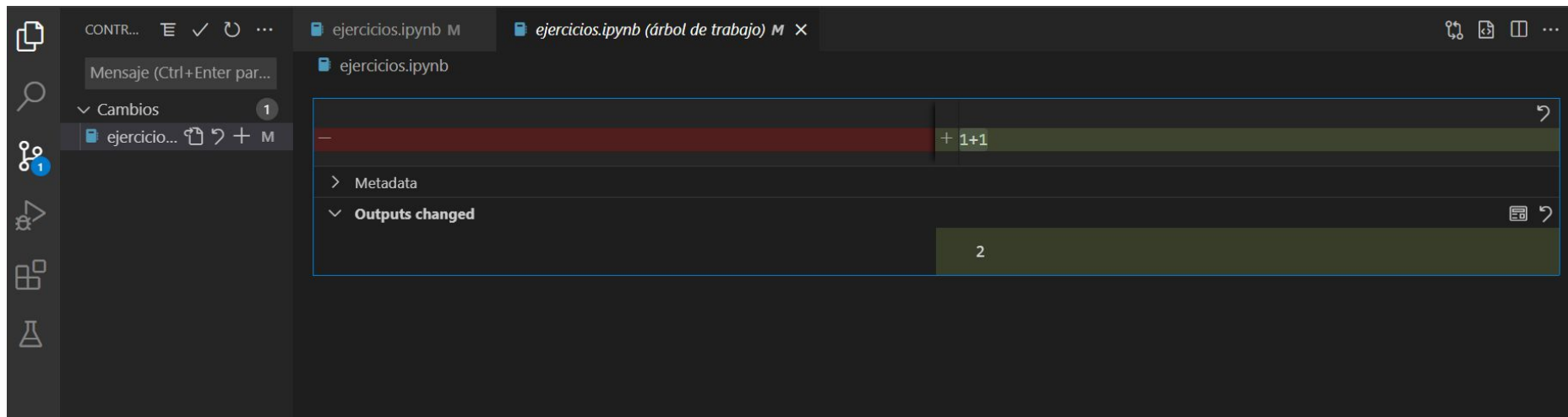




Visual Studio y Git - Modificaciones y Diff

- En el panel de control de código fuente, verá que su archivo ha cambiado. Mostrará la letra M junto a él, que representa un archivo que se ha modificado.
- VS Code también tiene la capacidad de realizar un **diff** sobre un archivo.
- Normalmente, tendría que descargar una herramienta diff independiente para hacer esto, de forma que esta función integrada puede ayudarle a trabajar de forma más eficiente.
- Para ver un diff, abra el panel de control del código fuente y haga doble clic en un archivo cambiado.
- En este caso, haga doble clic en el archivo creado. Se lo dirigirá a una vista diff normal con la versión actual del archivo a la izquierda y la versión previamente confirmada del archivo a la derecha.

Vuelva al control de versiones y vea los cambios



Trabajar con ramas

- Si nos movemos a la barra inferior, está presente la capacidad de crear y cambiar ramas.
- Echando un vistazo a la parte inferior izquierda del editor, debería ver el icono control de código fuente (el que parece una división en la carretera) seguido de probablemente el master o el nombre de la rama actualmente en funcionamiento.



- Para crear una rama, haga clic en ese nombre de la rama.
- Debería aparecer un menú que le da la capacidad para crear una nueva rama:



Crear rama

Select a ref to checkout

+ Create new branch...

+ Create new branch from...

master bc2d406a

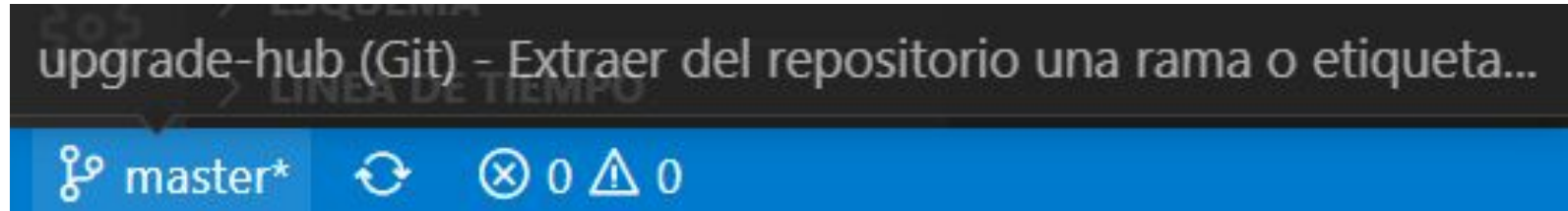


Trabajar con ramas

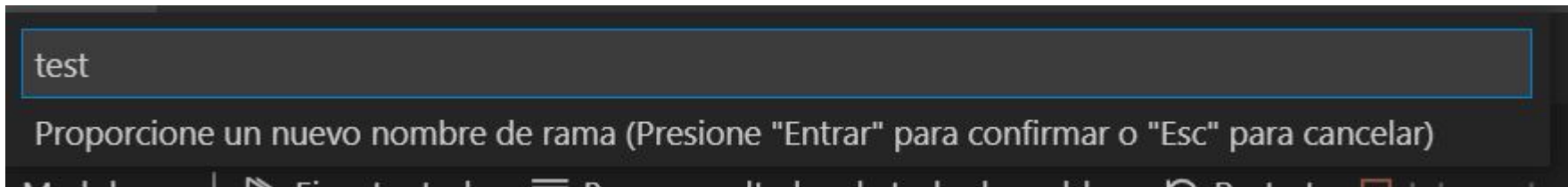
- Cree una rama llamada “test”.
- Ahora, realice un cambio a su archivo que significa que está en la nueva rama test
 - saludo = “Hola Mundo”
 - `print(saludo)`
- Confirme esos cambios a la rama test.
- A continuación, haga clic en el nombre de la rama en la parte inferior izquierda de nuevo para volver a la rama master.
- Tras volver a la rama master, observará que el que los cambios de la rama test ya no están presente.
- Veamos el ejemplo



Crear rama



Crear rama

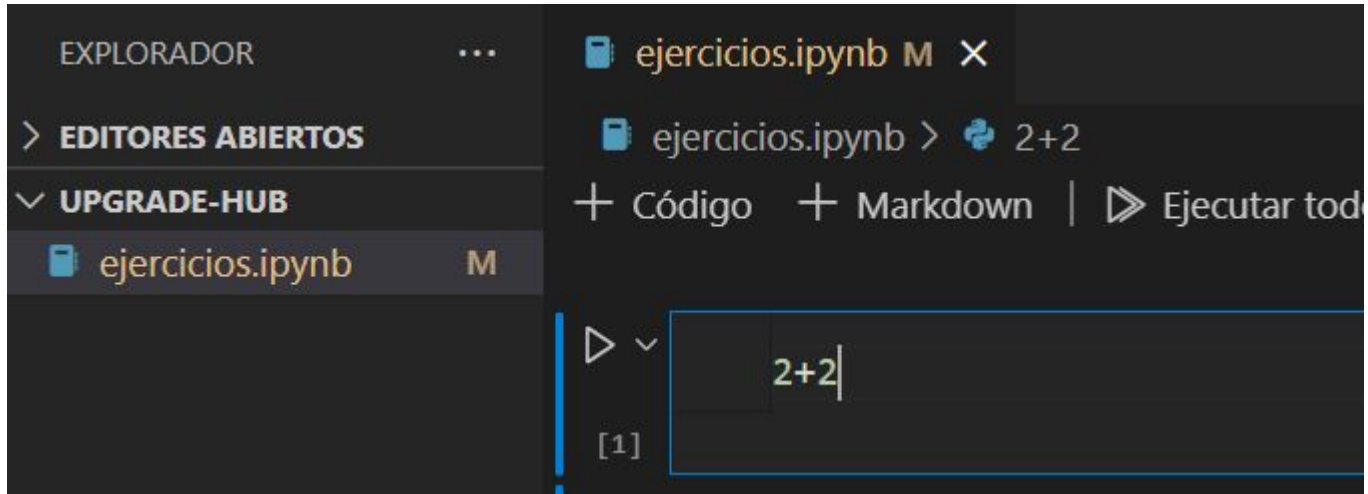


test

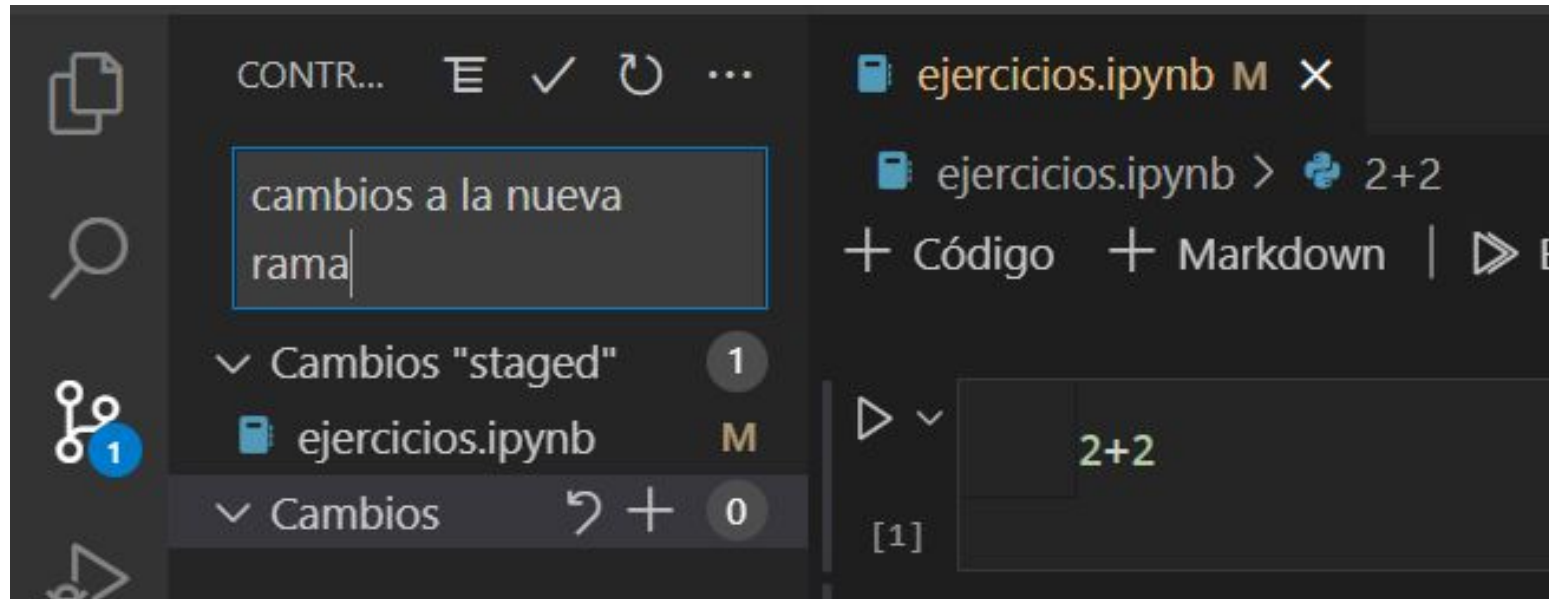
Proporcione un nuevo nombre de rama (Presione "Entrar" para confirmar o "Esc" para cancelar)



Crear rama



Subimos cambios a la nueva rama





Ve a GitHub y mira el contenido de ambas ramas

- Mira la siguiente captura para ver ambas ramas.
- El contenido de ambas ramas es diferente
- Las ramas son una división del estado del código, esto permite crear nuevos caminos a favor de la evolución del código.
- En Git, las ramas son parte diaria del desarrollo, son una guía instantánea para los cambios realizados.
- Imagina que quieres añadir una nueva función o tal vez arreglar un error, sin importar su tamaño, generas una nueva rama en la cual se alojan estos cambios que realizaste, al realizar esta acción va resultar más complicado que algún error o fallo del código inestable se incorpore al código base principal, dando la oportunidad de limpiar tu historial antes de fusionarlo todo con la rama principal, mejorando tu eficiencia de trabajo.

Ve a C

jalagrod / upgrade-hub Private Unwatch 1

<> Code Issues Pull requests Actions Projects Security

test had recent pushes 1 minute ago Compare & pull request

master Go to file Add file Code

Switch branches/tags

Find or create a branch...

Branches Tags

✓ master default

test

View all branches

3 hours ago 1


3 hours ago

Add a README



Ramas de código

- Las ramas en Git figuran como una línea independiente de desarrollo.
- Estas sirven como una abstracción en los procesos de edición, preparación y confirmación.
- Las puedes entender como una forma de solicitar un nuevo directorio de trabajo, un nuevo `workspace` de trabajo o un nuevo historial en el proyecto.
- Todas esas nuevas confirmaciones son registradas en el historial de la rama principal, creando una bifurcación en el historial del proyecto.
- Es posible que las ramas se integren (realizar un merge) en otras ramas.
- PD: A veces encontramos en la documentación que la rama principal es llamada “master” y otras “main”. Se refiere a lo mismo.



Commit, Pull y Push - Conceptos básicos

- Commit: Subir un cambio a una rama a través de un push.
- Push: Acción de subida de código. [Documentación.](#)
- Pull: Acción de bajada de código de una rama. [Documentación.](#)

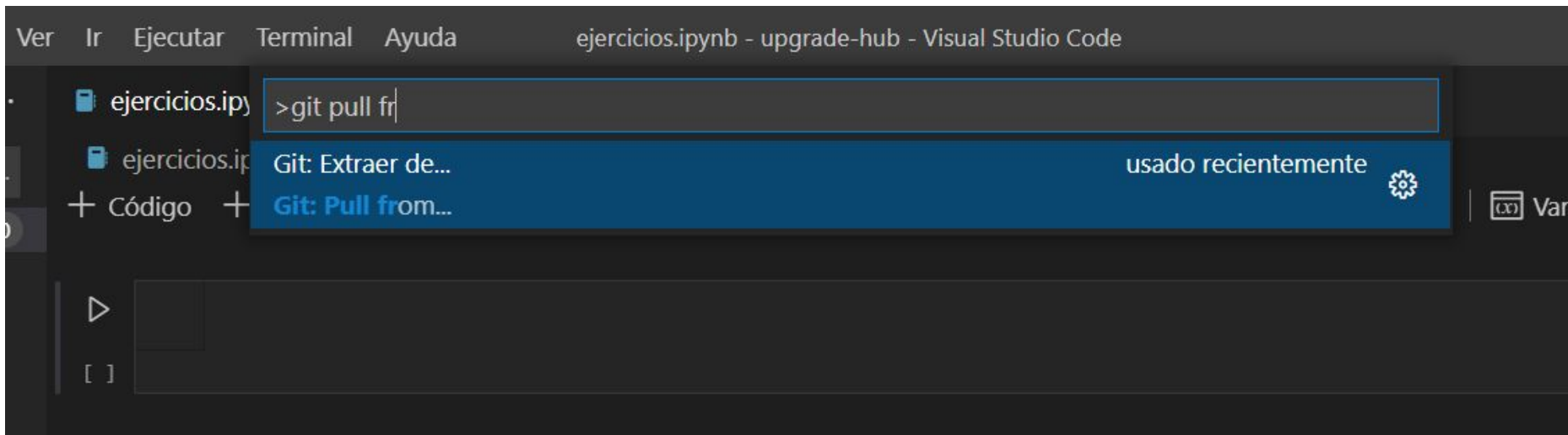
Dedica unos minutos a ver las documentaciones y entender qué ocurre en cada acción.

Merge y conflictos

- Ahora mismo debes tener cambios en la rama “master” y otros cambios en la rama “test”.
- Colócate en la rama “master”.
- Abre la consola de Git en VsCode. Pulsa:
 - MacOS: Command + Shift + p
 - Windows: Windows + Shift + p
- Haz un “git pull from” de la rama test.
- Veremos que existe un cambio para integrar o subir:
- Pulsa en la flecha arriba y estaremos realizando una integración de “test” en “master”. Ahora la rama “master” está actualizada. Veamos el ejemplo:

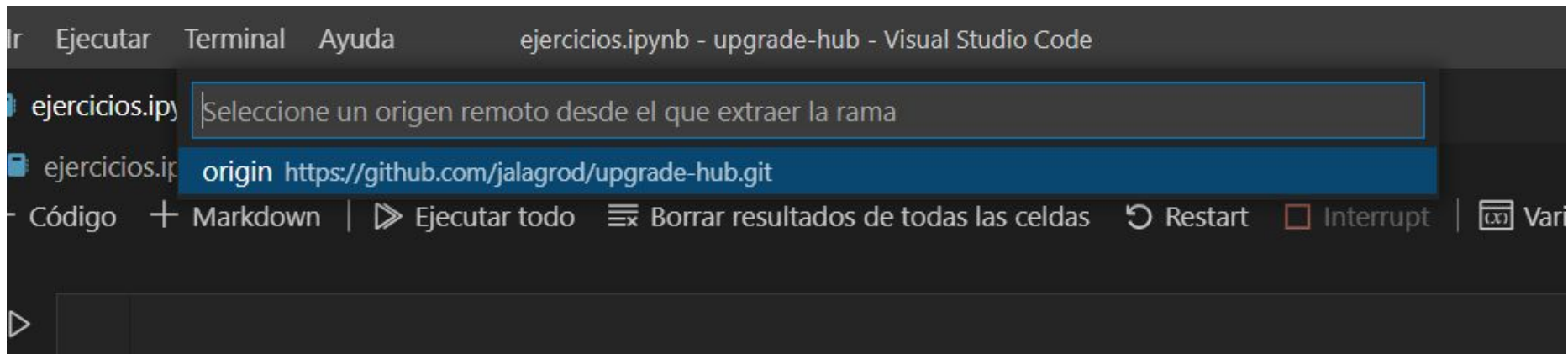


Merge y conflictos





Merge y conflictos





Merge y conflictos

Seleccionar una rama de la que extraer

origin/test

origin/master

 master  0↓ 1↑



Pull request y 'peer review'

- Las pull requests son una funcionalidad que facilita la colaboración entre desarrolladores.
- Es una solicitud de un desarrollador para que otro revise su trabajo y acepte los cambios.
- Las solicitudes de incorporación de cambios son un mecanismo para que los desarrolladores notifiquen a los miembros de su equipo que han terminado una función.
- Una vez la rama de función está lista, el desarrollador realiza la solicitud de incorporación de cambios mediante su cuenta de GitHub.
- Así, todas las personas involucradas saben que deben revisar el código y fusionarlo con la rama principal.
- Pero la solicitud de incorporación de cambios es mucho más que una notificación: es un foro especializado para debatir sobre una función propuesta. Si hay algún problema con los cambios, los miembros del equipo pueden publicar feedback en las solicitudes de incorporación de cambios e incluso modificar la función al enviar confirmaciones de seguimiento. El seguimiento de toda esta actividad se realiza directamente desde la solicitud de incorporación de cambios.

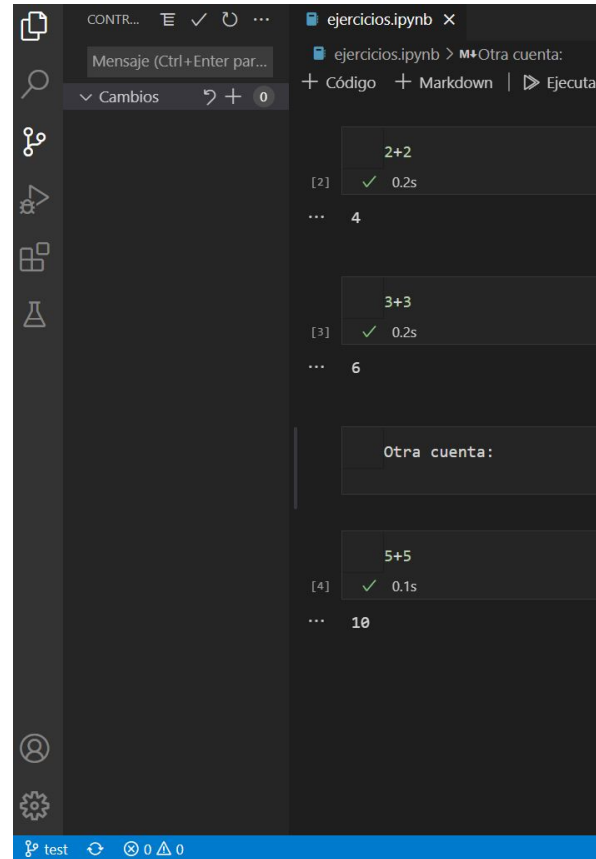


Pull Request - Lectura avanzada

- Invitad a un compañero a vuestro repositorio.
- Cread un cambio en una rama y cread un “pull request”.
- Asignad un “pull request” a vuestro compañero.
- Veamos el proceso a través de ejemplos:
 - Hemos creado nuevos cambios en “test”.
 - Vamos a realizar un “pull request” a través de la web.

[Lectura recomendada avanzada](#)

Cambios en “test”



Comprobar en la web los cambios

 test ▾ [upgrade-hub](#) / ejercicios.ipynb

 Jose Otro cambio en test

 0 contributors

96 lines (96 sloc) | 1.51 KB

In [2]: `2+2`

Out[2]: 4

In [3]: `3+3`

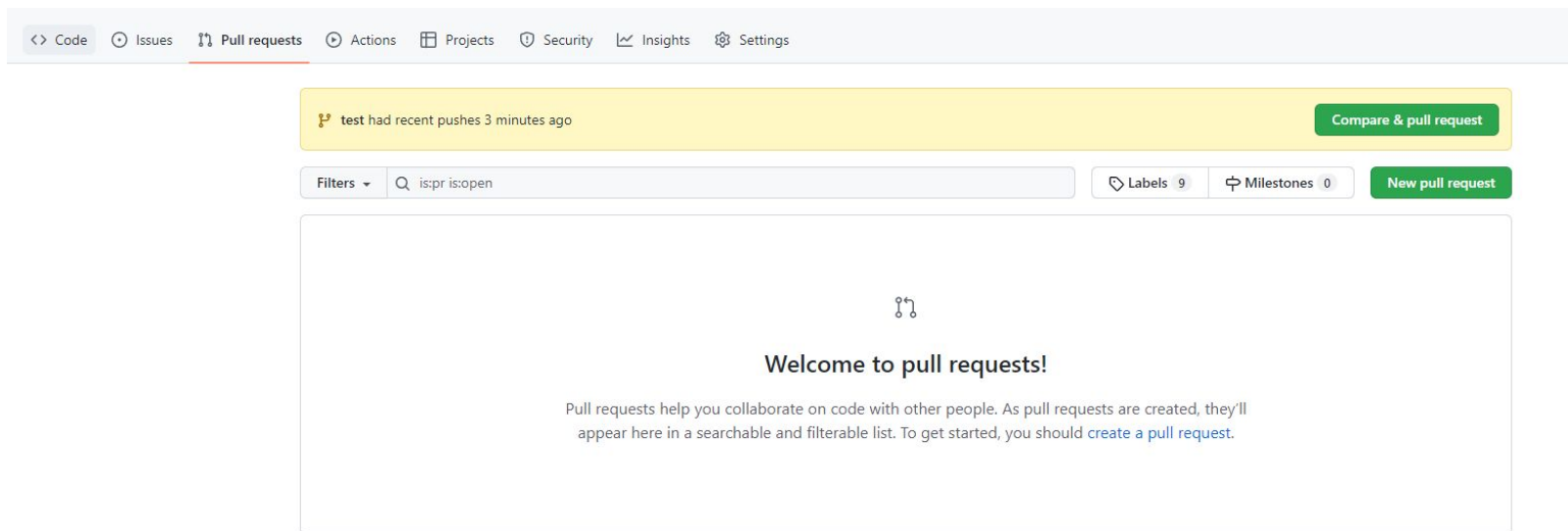
Out[3]: 6

Otra cuenta:

In [4]: `5+5`

Out[4]: 10

Sección “pull request” botón “Compare & pull request”




The screenshot shows the GitHub interface for the 'Pull requests' section. At the top, there's a navigation bar with tabs for Code, Issues, Pull requests (which is active), Actions, Projects, Security, Insights, and Settings. Below this, a yellow banner indicates that 'test' had recent pushes 3 minutes ago, with a 'Compare & pull request' button. Underneath the banner is a search bar with the filter 'is:pr is:open' and buttons for 'Labels 9' and 'Milestones 0'. A 'New pull request' button is also present. The main content area features a large icon of two branches merging and the text 'Welcome to pull requests!'. Below this, a paragraph explains that pull requests help with collaboration and that they will appear in a searchable and filterable list, with a link to 'create a pull request'.

<> Code Issues Pull requests Actions Projects Security Insights Settings

test had recent pushes 3 minutes ago Compare & pull request

Filters Q is:pr is:open Labels 9 Milestones 0 New pull request




Welcome to pull requests!

Pull requests help you collaborate on code with other people. As pull requests are created, they'll appear here in a searchable and filterable list. To get started, you should [create a pull request](#).

ProTip! Ears burning? Get @jalagrod mentions with [mentions:jalagrod](#).

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).

 base: master ← compare: test ✓ **Able to merge.** These branches can be automatically merged.



Write

Preview

Leave a comment

Attach files by dragging & dropping, selecting or pasting them.

Create pull request

- 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
[GitHub Community Guidelines](#)

Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).

1 commit

1 file changed

1 contributor

Commits on May 24, 2022

Otro cambio en test

Jose committed 4 minutes ago

Showing 1 **changed file** with 50 additions and 3 deletions.

Split

Unified

53 ejercicios.ipynb

↑

Crear pull request

- Añadir a vuestro compañero.
- Rellenad la información y dejad comentarios.
- Esta metodología se llama “peer review”

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

[GitHub Community Guidelines](#)

Crear pull request - final

The screenshot shows a GitHub Pull Request (PR) titled "Otro cambio en test #1". The interface includes a top navigation bar with links for Code, Issues, Pull requests (selected), Actions, Projects, Security, Insights, and Settings. The PR is from the `test` branch to the `master` branch, initiated by user `jalagrod`. The PR description is "Un comentario para la pull request". The interface shows the PR is ready for merge, with a green checkmark indicating "This branch has no conflicts with the base branch". A green "Merge pull request" button is visible. The right sidebar contains metadata for the PR, including Reviewers, Assignees, Labels, Projects, Milestone, and Development. The bottom section shows a comment input area with a "Write" tab and a "Preview" tab, and a "Close pull request" button.

Otro cambio en test #1

jalagrod wants to merge 1 commit into `master` from `test`

Conversation 0 Commits 1 Checks 0 Files changed 1 +50 -3

jalagrod commented now

Un comentario para la pull request

Otro cambio en test b0a19fe

Add more commits by pushing to the `test` branch on jalagrod/upgrade-hub.

Continuous integration has not been set up
GitHub Actions and several other apps can be used to automatically catch bugs and enforce style.

This branch has no conflicts with the base branch
Merging can be performed automatically.

Merge pull request You can also open this in GitHub Desktop or view command line instructions.

Write Preview H B I ≡ <> @ ↩ ↪

Leave a comment

Attach files by dragging & dropping, selecting or pasting them.

Close pull request Comment

1 participant



Readme y markdown

- README (como su nombre indica: «Léeme») es el primer archivo que se debe leer al iniciar un nuevo proyecto.
- Es un conjunto de información útil sobre un proyecto, y una especie de manual.
- Un archivo de texto README aparece en muchos lugares y se refiere no solo a la programación.
- Sin embargo, nos centraremos en el README de un programador.



¿De qué sirve escribir un buen README?

- Un buen README es para que otros entiendan lo que incluye nuestro código y por qué es notable.
- Un archivo README también es esencial para recuperar un proyecto, en GitHub pero también en navegadores.
- Cada vez los “recruiters” buscan a través de estos ficheros.
- Aunque estéis aprendiendo, es bueno usarlo para documentar.
- Intenta hacerlo en inglés para llegar a más público. Recordad que GitHub es vuestro futuro portfolio.
- Para escribir un buen “Readme.md” se usa lenguaje de marcado “markdown”.
- Puedes ver un resumen de las instrucciones que se pueden usar aquí:
 - <https://markdownlivepreview.com/>
- Dedicar un tiempo a ver este repositorio de ejemplo de ficheros “Readme.md”
 - <https://github.com/inessadl/readme>



Crea un fichero Readme.md y súbalo al repositorio

- Genere un fichero.
- Súbalo
- Mire en la portada del repositorio cómo el fichero representa la portada de su código.
- Tener buenos ficheros “Readme.md” ayuda a todas las partes.


Recordad

In case of fire



 **1. git commit**

 **2. git push**

 **3. leave building**