

---

# Algoritmos y Estructuras de Datos

## GitHub

---



### Contenidos del documento:

- ❖ ¿Qué es [GitHub](#) y para qué sirve?
- ❖ [Conceptos Basicos](#)
- ❖ Como crear [repositorios](#)
- ❖ Como crear [ramas](#)
- ❖ [Commit](#)
- ❖ [Pull Request](#)

## ¿Qué es GitHub y para qué sirve?:

Imagínate que estás trabajando en un proyecto grupal, como un trabajo de investigación o una presentación. Cada persona hace cambios en su propia copia del archivo, y luego tienen que unirlos manualmente, lo que puede ser un caos. GitHub resuelve este problema para el código y otros archivos.

GitHub es un servicio de alojamiento de repositorios: Aloja tus proyectos en la nube. Un repositorio es como una carpeta de proyecto que contiene todos tus archivos, el historial de cambios y la información de colaboración.

Git es un sistema de control de versiones distribuido. Esto significa que registra cada cambio que haces en tus archivos, permitiéndote volver a versiones anteriores, ver quién hizo qué cambio y fusionar el trabajo de diferentes personas sin problemas. GitHub es una interfaz web popular que utiliza Git.

GitHub es una plataforma colaborativa: Permite que múltiples personas trabajen en el mismo proyecto simultáneamente sin pisar los cambios. Facilita la revisión de código, la discusión de ideas y la integración de las contribuciones de todos.

### ¿Qué te permite hacer GitHub?

- ❖ **Control de versiones:** Evita la pesadilla de múltiples versiones de archivos y la pérdida de trabajo.
- ❖ **Colaboración:** Permite que equipos de cualquier tamaño trabajen juntos de forma eficiente, sin importar su ubicación.
- ❖ **Portafolio:** Muchos empleadores y universidades revisan los perfiles de GitHub para ver proyectos y contribuciones.
- ❖ **Comunidad:** GitHub es el hogar de millones de proyectos de código abierto, lo que permite a cualquiera aprender de otros, contribuir y ser parte de una comunidad global de desarrolladores.

## Conceptos Basicos:

- ❖ **Repository (Repo):** Es una carpeta que contiene todos los archivos de tu proyecto (código, imágenes, documentación, etc.) junto con el historial completo de todos los cambios realizados.
- ❖ **Commit:** Un "commit" es como tomar una instantánea de tus cambios en un momento dado. Cada vez que haces cambios en tus archivos y decides guardarlos en tu historial de Git, haces un commit. Cada commit tiene un mensaje que describe los cambios que hiciste, lo cual es crucial para entender el historial del proyecto.
- ❖ **Branch (Rama):** Las ramas te permiten trabajar en diferentes versiones de un proyecto al mismo tiempo. Imaginen que tienen una línea de tiempo principal de un proyecto. Si quieren añadir una nueva funcionalidad o corregir un error, crean una "rama" de esa línea principal. Trabajan en esa rama de forma independiente, y cuando terminan, la fusionan de nuevo con la línea principal. Esto evita que los cambios experimentales afecten la versión estable del proyecto.
- ❖ **Fork:** Hacer un "fork" de un repositorio es crear tu propia copia personal de un proyecto existente que está alojado en GitHub. Es útil cuando querés contribuir a un proyecto de otra persona sin modificar directamente el repositorio original. Haces tu propia copia, trabajas en ella, y luego puedes proponer tus cambios al creador original.
- ❖ **Pull Request (PR):** Después de hacer cambios en tu rama o en un fork de un proyecto, un "pull request" es tu forma de notificar a los demás que has terminado tus cambios y te gustaría que se revisaran e incorporaran al repositorio principal. Es el método estándar para colaborar en GitHub: la gente revisa tus cambios, los discuten y, si están de acuerdo, los fusionan.
- ❖ **Clone:** "Clonar" un repositorio significa descargar una copia completa de un repositorio de GitHub a tu computadora local. Esto te permite trabajar en el proyecto en tu propio entorno y luego subir tus cambios de nuevo a GitHub.

# Repositorios:

The screenshot shows the GitHub Home page. At the top, there's a search bar and various navigation icons. On the left, there's a sidebar with options like 'Create your first project', 'Create repository' (which is highlighted in green), and 'Import repository'. The main area has sections for 'Home' (with buttons for 'Ask Copilot', 'Make a Pong game', 'Suggest next steps for an issue', and 'Create an issue for a bug') and 'Feed' (showing 'Trending repositories' with links to 'bytedance/trae-agent' and 'coleam00/context-engineering-intro'). To the right, there's a 'Latest changes' sidebar with a list of recent updates.

Tocan donde dice Create repository luego les va a aparece la página de la imagen de abajo.

- ❖ **Owner:** El dueño del repositorio
- ❖ **Repository name:** Nombre del Repositorio
- ❖ **Descripción:** Una Descripción General del contenido del repositorio
- ❖ **Public o Private:** Esto determina quienes pueden ver el repositorio
- ❖ **README:** Un lugar donde podemos agregar una descripción mas extensa.
- ❖ **Gitignore:** Una opción para filtrar la cantidad de archivos necesarios del proyecto.
- ❖ **Licencia:** Podes elegir automáticamente entre diferentes licencias de uso para tu proyecto



**Owner \*** ignaciotraverso71 / **Repository name \***

Great repository names are short and memorable. Need inspiration? How about **fuzzy-octo-sniffle**?

**Description (optional)**

**Public**  
Anyone on the internet can see this repository. You choose who can commit.

**Private**  
You choose who can see and commit to this repository.

**Initialize this repository with:**

**Add a README file**  
This is where you can write a long description for your project. [Learn more about READMEs](#).

**Add .gitignore**

.gitignore template: **None**

Choose which files not to track from a list of templates. [Learn more about ignoring files](#).

**Choose a license**

License: **None**

A license tells others what they can and can't do with your code. [Learn more about licenses](#).

You are creating a public repository in your personal account.

**Create repository**

Test Public Pin Watch 0 Fork 0 Star 0

main 2 Branches 0 Tags Go to file Add file Code About Settings

ignaciotraverso71 Initial commit 5e061a4 · 20 hours ago 1 Commit

LICENSE Initial commit 20 hours ago

README GPL-3.0 license

**Add a README**

Help people interested in this repository understand your project by adding a README.

**Add a README**

**Hola**

GPL-3.0 license

Activity

0 stars

0 watching

0 forks

**Releases**

No releases published [Create a new release](#)

**Packages**

No packages published [Publish your first package](#)

Esta es la página principal de nuestro repositorio, en donde podemos ver la rama principal (main), al mismo tiempo podemos acceder desde aquí a todas las opciones que tenemos para nuestro repositorio.

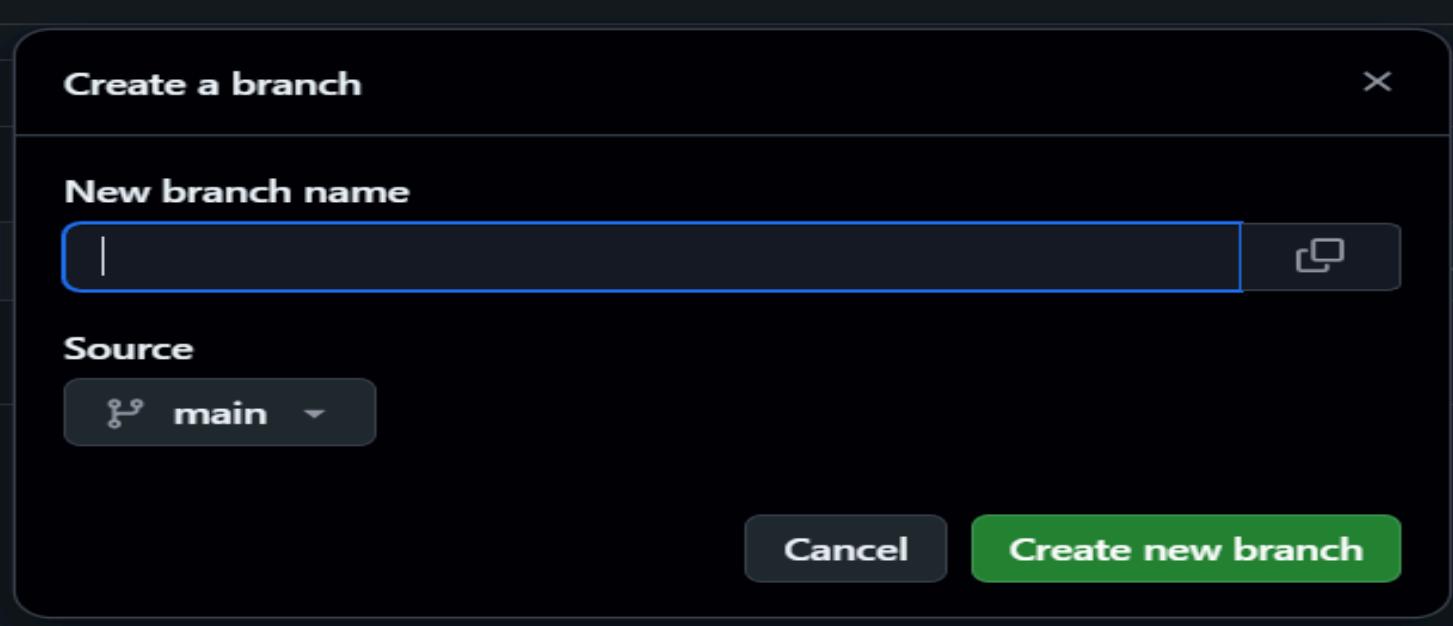
The screenshot shows a GitHub repository page for 'ignaciotraverso71/Test'. At the top, there are navigation links: 'Go to file' (with a search icon), 'Add file' (with a plus icon), and 'Code' (with a code icon). Below these are tabs for 'Local' and 'Codespaces'. A 'Clone' button is present, followed by three cloning methods: 'HTTPS' (selected), 'SSH', and 'GitHub CLI'. The HTTPS URL is highlighted with a red box: <https://github.com/ignaciotraverso71/Test.git>. Below the URL, there's a link to 'Clone using the web URL.' and two more options: 'Open with GitHub Desktop' and 'Download ZIP'. On the left side of the main content area, there are icons for 'I commit', 'I have a README', and 'Understand your project by adding a README file.'.

en la pestaña **<> Code** podemos encontrar un link para compartir el repositorio.

## Ramas:

The screenshot shows the 'Branches' page for the repository. It features a header with 'Branches' and a 'New branch' button. Below the header are tabs: 'Overview' (selected), 'Yours', 'Active', 'Stale', and 'All'. A search bar allows searching for branches. The main content area is titled 'Default' and lists a single branch: 'main'. The table columns include 'Branch', 'Updated', 'Check status', 'Behind/Ahead', and 'Pull request'. The 'main' branch is highlighted with a blue background. The 'Updated' column shows 'yesterday' with a small clock icon. The 'Check status' column shows 'Default'. The 'Behind/Ahead' and 'Pull request' columns are empty.

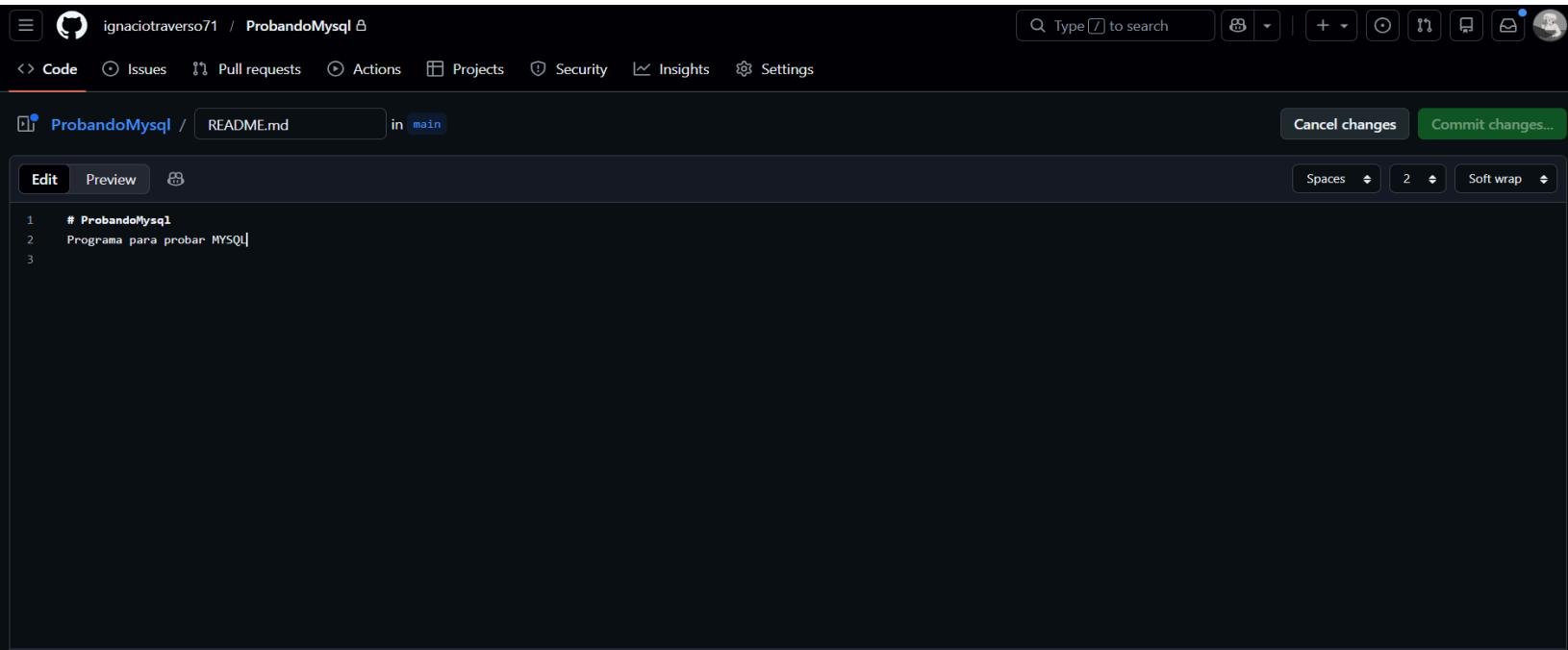
Una vez entramos a la pestaña de ramas vamos a ver la sección de arriba, en la que vamos a ver todas las ramas, si tienen pull request y también cuando fue la última vez que se actualizo.



Al tocar el botón de crear rama aparecerá esta pestaña, en la que podemos poner el nombre de la rama nueva y a partir de qué rama se va a crear.

## Commit:

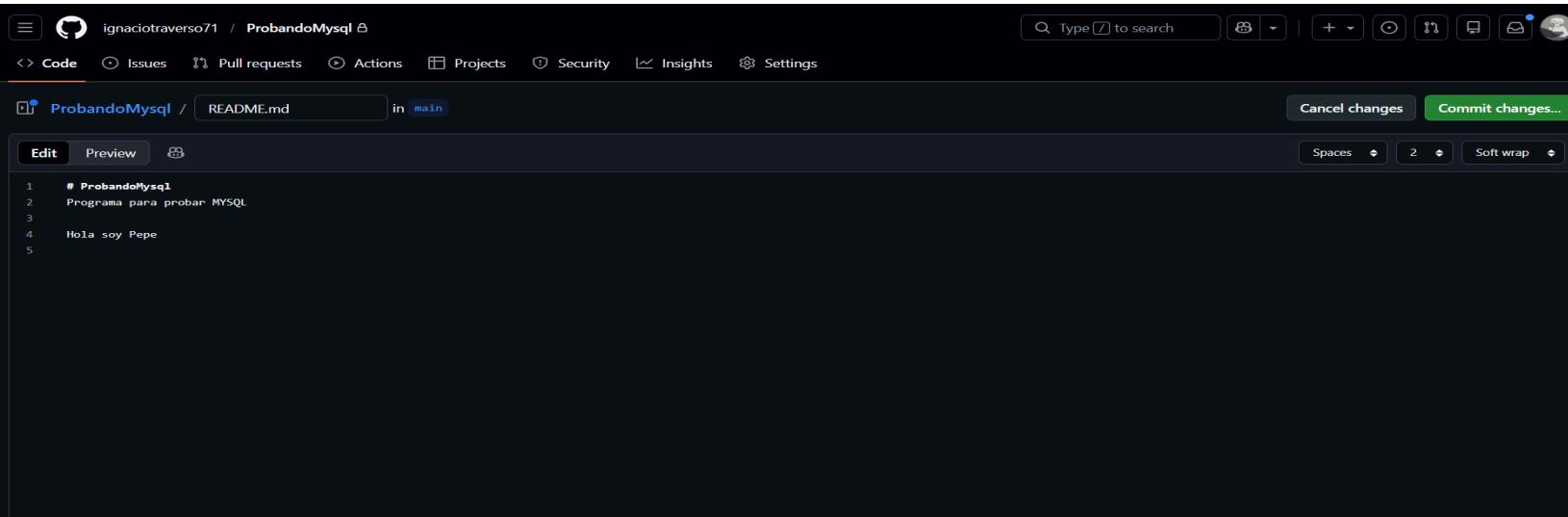
El commit funciona como una confirmación de guardar los cambios que hagas en el repositorio, siempre que hagan un cambio en el código, les va a aparecer un botón que dice commit, una vez lo aprieten y confirmen el cambio va a quedar guardado.



A screenshot of a GitHub code editor interface. The top navigation bar shows the repository "ignaciotraverso71 / ProbandoMysql". The main menu includes "Code", "Issues", "Pull requests", "Actions", "Projects", "Security", "Insights", and "Settings". Below the menu, the file "README.md" is selected, and the branch "main" is shown. The code editor displays the following content:

```
1 # ProbandoMysql
2 Programa para probar MYSQL
```

The editor has a toolbar with "Edit", "Preview", and "diff" icons. On the right, there are buttons for "Cancel changes" and "Commit changes...". The bottom of the editor shows standard text editing options: "Spaces", "2", and "Soft wrap".



A screenshot of a GitHub code editor interface, identical to the one above but with an additional line of text added to the README file.

```
1 # ProbandoMysql
2 Programa para probar MYSQL
3
4 Hola soy Pepe
5
```

## Commit changes

X

### Commit message

Le hice una update al Readme

### Extended description

se explica solo hermano que mas explicación querés?

- Commit directly to the `main` branch
- Create a new branch for this commit and start a pull request [Learn more about pull requests](#)

Cancel

Commit changes

como dueños del repositorio van a tener la opcion de directamente guardar los cambios o crear una nueva rama para iniciar un pull request con sus compañeros de repositorio.

## Pull Request:

Para hacer un pull request vamos a necesitar dos ramas, una rama original y una rama hija de esa misma.

Lo que vamos a hacer es cambiar la rama hija a nuestro gusto para hacer luego un pull request a la rama padre. Esto lo que va a hacer es mezclar ambas ramas en la rama padre.

The screenshot shows a GitHub repository page for 'ipowerstore'. At the top, it says 'La-Nueva-Branch had recent pushes 7 seconds ago'. Below that, there are buttons for 'Compare & pull request', 'main', '2 Branches', '0 Tags', 'Go to file', 'Add file', and 'Code'. The repository has 6 commits from 'mototruco'. The files listed are 'recursos', 'ipowerstore.html', and 'styles.css', all updated 3 months ago by 'xxpkpmc'. Below the files, there's a section for 'README' with a button to 'Add a README'. A message encourages adding a README to help people understand the project.

The screenshot shows the same GitHub repository page for 'ipowerstore'. Now, 'La-Nueva-Branch' is selected, indicated by a dropdown arrow next to its name. It says 'La-Nueva-Branch had recent pushes 4 minutes ago'. The 'Compare & pull request' button is still present. The repository now has 7 commits from 'ignaciotraverso71'. The files listed are 'recursos', 'README.md', 'ipowerstore.html', and 'styles.css', with 'README.md' being the most recent commit (4 minutes ago) and others from 3 months ago. Below the files, there's a section for 'README' with a button to 'Edit'. A message at the bottom says 'Kasane Teto y Hatsune Miku.'

Una vez se toque el botón de pull request ya va a estar listo. Lo único que falta es que el dueño del repositorio acepte que se haga el pull.

