

Especialización en
Programador Web

GIT & GITHUB



Prof. Katherine Molina

Guía de GitHub

Contenido

1. ¿Qué es GitHub?	3
¿Por qué conviene usar control de versiones?	3
2. Git y GitHub: ¿es lo mismo?	3
3. Conceptos básicos	3
4. Crear tu cuenta	3
5. Crear un repositorio: público o privado	4
.....	5
6. Subir tus archivos (HTML, CSS, JS)	6
7. Editar o actualizar archivos	6
8. Comandos básicos de Git (opcional, para trabajar desde tu computadora)	7
9. Ramas (branches)	7
Crear una rama desde la web	7
10. Combinar cambios (merge)	8
11. Resolver conflictos	8
12. Deshacer cambios	8
13. Ignorar archivos con .gitignore	8
Cómo crearlo	8
14. Agregar colaboradores	9
15. Fork y Pull Request: colaborar en proyectos de otras personas	9
Fork	9
Pull Request (PR)	10
16. Seguir personas y repositorios	10
Seguir a una persona (Follow)	10
Seguir un repositorio (Watch)	10
17. Ponerle una estrella (Star) a un repositorio	11
18. Buscar repositorios en GitHub	11
19. Publicar tu sitio con GitHub Pages	11
20. Glosario rápido	12

1. ¿Qué es GitHub?

GitHub es una página web donde podés guardar tus proyectos de código (por ejemplo, los archivos HTML, CSS y JavaScript de una página web) para no perderlos, mostrarlos a otras personas o trabajar en equipo. (o lo que quieras)

Pensalo como una carpeta en la nube (parecida a Google Drive), pero pensada especialmente para código, con funciones extra como guardar el historial de cambios, trabajar con otras personas y publicar sitios web gratis.

Cada proyecto que subís se guarda dentro de un repositorio (repo), que es simplemente la carpeta de ese proyecto dentro de GitHub.

¿Por qué conviene usar control de versiones?

"Control de versiones" es la idea de ir guardando distintas versiones de tu proyecto a medida que trabajás, en vez de tener un único archivo que se va sobrescribiendo. Esto trae varias ventajas:

- Nunca perdés una versión anterior de tu trabajo: siempre podés volver atrás.
- Podés ver exactamente qué cambió, cuándo y quién lo hizo.
- Varias personas pueden trabajar en el mismo proyecto sin pisarse el trabajo entre sí.
- Podés probar ideas nuevas en una copia aparte (una rama) sin arriesgar la versión que ya funciona.

2. Git y GitHub: ¿es lo mismo?



Es una confusión muy habitual, pero son dos cosas relacionadas y distintas:

Herramienta	Qué es
Git	Es el sistema que guarda las versiones de tus archivos. Se puede usar incluso sin internet, trabajando solo en tu computadora.
GitHub	Es la página web donde alojás tus repositorios de Git en la nube y accedés a herramientas de colaboración: invitar colaboradores, Pull Requests, GitHub Pages, etc.

3. Conceptos básicos

Estas palabras aparecen todo el tiempo al usar GitHub. No hace falta memorizarlas, pero ayuda tenerlas presentes:

Palabra	Qué significa
Repositorio (repo)	La "carpeta" de un proyecto dentro de GitHub.
Commit	Una versión guardada de tus archivos, con un mensaje que describe el cambio. Es como sacarle una "foto" al proyecto en ese momento.
Rama (branch)	Una copia paralela del proyecto para probar o desarrollar algo sin afectar la versión principal.
main	El nombre habitual de la rama principal: la versión "oficial" del proyecto.

4. Crear tu cuenta

1. Entrar a github.com
2. Hacer clic en "Sign up" (Registrarse).
3. Completar correo electrónico, contraseña y un nombre de usuario.



Sign up for GitHub

Continue with Google

Continue with Apple

or

Email*

Email

Password*

Password

Password should be at least 15 characters OR at least 8 characters including a number and a lowercase letter.

Username*

Username

Username may only contain alphanumeric characters or single hyphens, and cannot begin or end with a hyphen.

Your Country/Region*

Argentina

For compliance reasons, we're required to collect country information to send you occasional updates and announcements.

GitHub Copilot

☐ Sign up for Copilot Free for coding assistance on the web, in the IDE, and in the CLI. [See terms](#)

Email preferences

☐ Receive occasional product updates and announcements

Create account >

4. Confirmar el correo electrónico cuando GitHub lo solicite.

El nombre de usuario que elijas va a formar parte de la dirección de tus repositorios y de tu futura página publicada, así que conviene elegir uno simple y fácil de recordar.

5. Crear un repositorio: público o privado

1. Hacer clic en el botón verde "New" (o el símbolo "+" arriba a la derecha → "New repository").

Find a repository...

Type ▾

Language ▾

Sort ▾

New

2. Ponerle un nombre al repositorio (por ejemplo: mi-primera-web).
3. Elegir si va a ser público o privado.
4. Opcional: tildar "Add a README file" para tener una descripción inicial.
5. Hacer clic en "Create repository".

Término	Qué significa
Repositorio público	Cualquier persona en internet puede ver el código, aunque solo vos (y las personas que invites) pueden modificarlo.
Repositorio privado	Solo vos y las personas que invites como colaboradoras pueden ver y modificar el código.
Para tener en cuenta Si querés publicar tu sitio con GitHub Pages usando una cuenta gratuita, el repositorio generalmente debe ser público.	

Create a new repository

Repositories contain a project's files and version history. Have a project elsewhere? [Import a repository](#).
Required fields are marked with an asterisk (*).

1 General

Owner * katheemolina / Repository name *

Great repository names are short and memorable. How about [animated-carnival](#)?

Description

0 / 350 characters

2 Configuration

Choose visibility * Public

Choose who can see and commit to this repository

Start with a template No template

Templates pre-configure your repository with files.

Add README Off

READMEs can be used as longer descriptions. [About READMEs](#)

Add .gitignore No .gitignore

.gitignore tells git which files not to track. [About ignoring files](#)

Add license No license

Licenses explain how others can use your code. [About licenses](#)

Create repository

Crear un nuevo repositorio

Los repositorios contienen los archivos y el historial de versiones de un proyecto. ¿Tienes un proyecto en otro lugar? [Importa un repositorio](#).
Los campos obligatorios están marcados con un asterisco (*).

1 General

Dueño * katheemolina / Nombre del repositorio *

Los buenos nombres de repositorios son cortos y fáciles de recordar. ¿Qué tal si...? [carnaval animado](#)?

Descripción

0 / 350 caracteres

2 Configuración

Seleccione la visibilidad * Público

Elige quién puede ver y realizar cambios en este repositorio.

Comience con una plantilla Sin plantilla

Las plantillas preconfiguran tu repositorio con archivos.

Agregar README Apagado

Los archivos README pueden utilizarse como descripciones más extensas. [Acerca de los archivos README](#)

Agregar .gitignore No .gitignore

.gitignore le indica a git qué archivos no debe rastrear. [Acerca de los archivos ignorados](#)

Agregar licencia Sin licencia

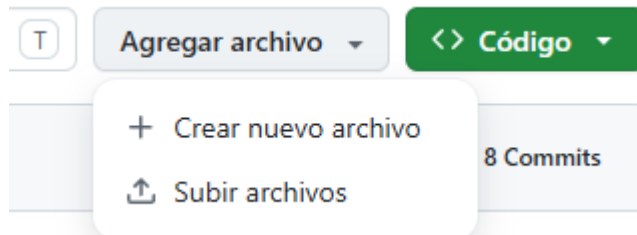
Las licencias explican cómo otros pueden usar tu código. [Acerca de las licencias](#)

Crear repositorio

6. Subir tus archivos (HTML, CSS, JS)

No hace falta instalar nada ni usar la terminal: se puede subir archivos directamente desde el navegador.

1. Entrar al repositorio.
2. Hacer clic en "Add file" → "Upload files".



3. Arrastrar los archivos (index.html, style.css, script.js, etc.) a la ventana, o hacer clic para buscarlos en tu computadora.
4. Escribir un breve mensaje describiendo qué subiste (por ejemplo: "Primera versión de la página").
5. Hacer clic en "Commit changes" para guardar los archivos.

A screenshot of the GitHub 'Confirmar cambios' (Commit changes) form. At the top, there is a large box with a file icon and the text 'Arrastra los archivos aquí para agregarlos a tu repositorio.' and a link 'elige tus archivos'. Below this is the 'Confirmar cambios' section, which includes a profile picture, a title 'Confirmar cambios', a text input field for 'Agregar archivos mediante carga', a larger text area for 'Agregue una descripción extendida opcional...', and two radio buttons for 'Comprométete directamente con el principal rama.' (selected) and 'Crea una nueva rama para esta confirmación e inicia una solicitud de extracción.' Below the form are two buttons: 'Confirmar cambios' and 'Cancelar'.

¿Qué es un "commit"?

Es simplemente el nombre que GitHub le da a "guardar una versión" de tus archivos. Cada vez que subís o modificás algo y confirmás el cambio, se crea un commit, y podés volver a versiones anteriores si lo necesitás.

7. Editar o actualizar archivos

Para modificar un archivo ya subido sin tener que volver a subirlo entero:

1. Entrar al archivo dentro del repositorio.
2. Hacer clic en el ícono del lápiz (Edit this file).
3. Hacer los cambios directamente en el navegador.
4. Bajar hasta el final de la página y hacer clic en "Commit changes" para guardarlos.

8. Comandos básicos de Git (opcional, para trabajar desde tu computadora)

Todo lo anterior se puede hacer sin comandos, subiendo archivos desde la web. Pero si en algún momento trabajás con un editor de código (como Visual Studio Code) y querés subir tus cambios desde ahí en lugar de arrastrar archivos a la web, estos son los comandos más usados:

Comando	Qué hace
git clone <url>	Descarga a tu computadora una copia completa de un repositorio de GitHub.
git add .	Prepara todos los archivos que modificaste para guardarlos en la próxima versión.
git commit -m "mensaje"	Guarda esa versión con un mensaje que describe qué cambiaste.
git push	Sube esos cambios guardados a GitHub.
git pull	Trae a tu computadora los cambios más recientes que estén en GitHub.
No es obligatorio Si te sentís más cómoda o cómodo subiendo archivos por la web, podés seguir así perfectamente. Estos comandos son solo una forma alternativa de hacer lo mismo, más rápida cuando ya tenés práctica.	

9. Ramas (branches)

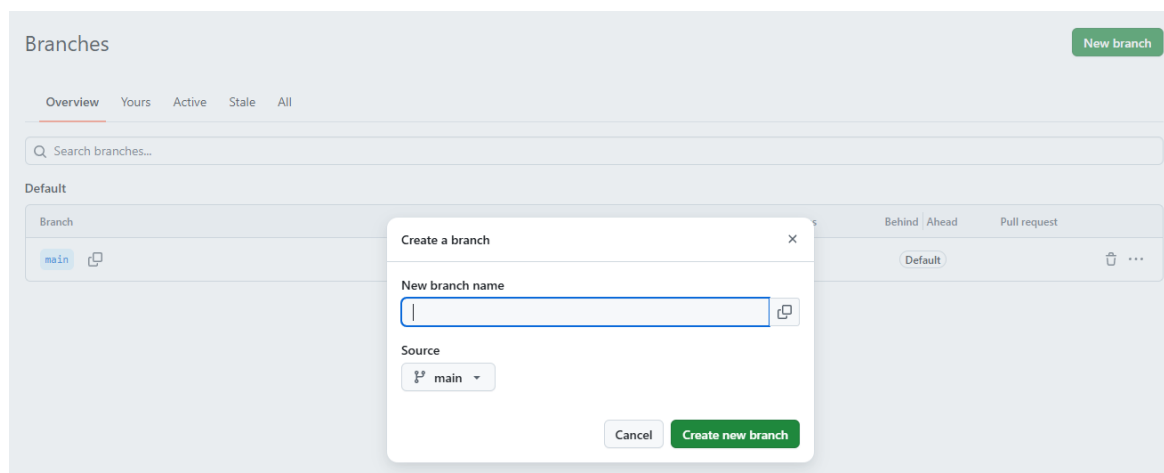
Una rama es una copia paralela de tu proyecto. Sirve para probar algo nuevo (un cambio de diseño, una función nueva) sin tocar la versión principal (main) hasta estar segura o seguro de que funciona bien.

Crear una rama desde la web

1. Entrar al repositorio.
2. Hacer clic en el selector de ramas, donde dice "main" (arriba a la izquierda, sobre la lista de archivos).



3. Escribir el nombre de la nueva rama (por ejemplo: nuevo-diseño) y confirmar.
4. A partir de ahí, los cambios que hagas dentro de esa rama no van a afectar a main.



10. Combinar cambios (merge)

Cuando terminás de trabajar en una rama y los cambios funcionan bien, el paso siguiente es incorporarlos a la rama principal. A eso se lo llama fusionar o hacer merge.

La forma más habitual de hacerlo en GitHub es a través de un Pull Request (lo vemos en la sección 15): se abre un Pull Request desde la rama nueva hacia main, y al aprobarlo aparece un botón "Merge pull request" que junta los cambios automáticamente.

Con comandos

Si estás trabajando desde tu computadora, el comando equivalente es `git merge <rama>`, ejecutado estando parada o parado sobre la rama main.

11. Resolver conflictos

A veces, si dos personas (o dos ramas) modificaron exactamente las mismas líneas de un archivo, GitHub no sabe cuál de las dos versiones conservar. A eso se lo llama conflicto.

Cuando esto pasa al intentar fusionar un Pull Request, GitHub te avisa que hay conflictos y te ofrece un editor donde se ven ambas versiones marcadas. Ahí podés elegir qué contenido dejar (uno de los dos, una combinación de ambos, o algo nuevo) y luego confirmar la resolución.

- Revisar con calma cada conflicto marcado, sin apurarse.
- Si no estás segura o seguro de qué versión conservar, es buena idea consultarlo con quien hizo el otro cambio.

12. Deshacer cambios

Como cada commit queda guardado en el historial, siempre podés ver cómo estaba el proyecto antes. Para consultarlo: entrar al repositorio → pestaña "Commits", donde aparece la lista completa de versiones guardadas, cada una con fecha y mensaje.

Si un cambio reciente rompió algo y querés deshacerlo, GitHub permite revertir un commit (crear automáticamente uno nuevo que deja todo como estaba antes) desde el historial de un Pull Request ya fusionado, con el botón "Revert".

Con comandos

El comando equivalente es `git revert <commit>`, que crea un nuevo commit deshaciendo los cambios de otro, sin borrar el historial.

13. Ignorar archivos con .gitignore

El archivo .gitignore le indica a GitHub qué archivos o carpetas no subir nunca al repositorio. Es útil para dejar afuera archivos que no hacen falta compartir, como configuraciones del editor de código o archivos temporales que genera tu sistema operativo.

Cómo crearlo

1. Entrar al repositorio → "Add file" → "Create new file".
2. Nombrar el archivo exactamente .gitignore
3. Escribir, uno por línea, qué querés ignorar.
4. Hacer clic en "Commit changes".

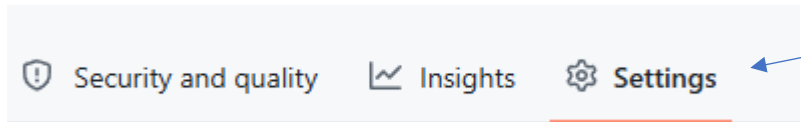
Ejemplo típico para un proyecto simple de HTML, CSS y JS:

```
.DS_Store
Thumbs.db
.vscode/
```

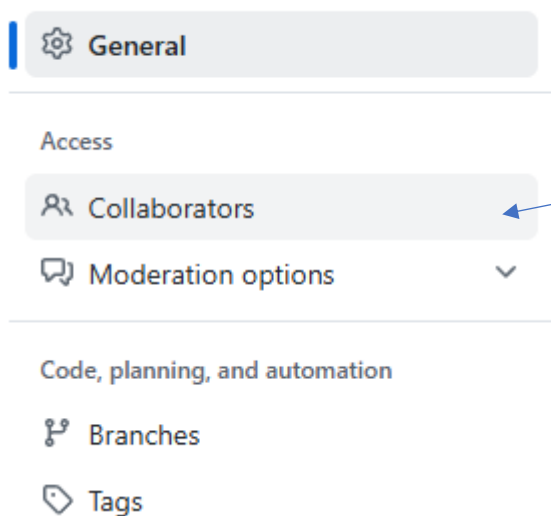
14. Agregar colaboradores

Si querés que otra persona pueda modificar tu repositorio (por ejemplo, un compañero de trabajo), podés invitarla como colaboradora:

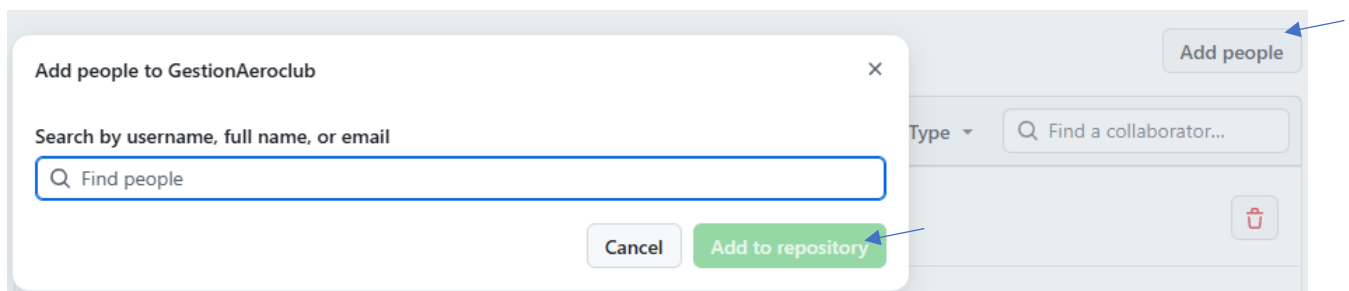
1. Entrar al repositorio → pestaña "Settings".



2. En el menú de la izquierda, hacer clic en "Collaborators".



3. Hacer clic en "Add people".



4. Escribir el nombre de usuario o el correo de GitHub de esa persona.
5. Esa persona va a recibir una invitación por correo, que debe aceptar.

Una vez aceptada la invitación, la persona colaboradora puede ver (y, si el repositorio es privado, también editar) el proyecto igual que vos.

15. Fork y Pull Request: colaborar en proyectos de otras personas

Cuando querés proponerle cambios a un repositorio que no es tuyo (y sobre el que no tenés permiso de colaboradora o colaborador), se usa el siguiente circuito:

Fork

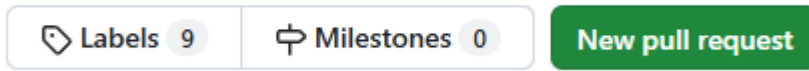
Un fork es una copia de un repositorio ajeno, dentro de tu propia cuenta. Se hace con el botón "Fork" arriba a la derecha del repositorio original. A partir de ahí, esa copia es tuya y podés modificarla libremente.

Pull Request (PR)

Un Pull Request es una solicitud para que tus cambios (hechos en tu fork o en una rama) se incorporen al repositorio original. Permite que la otra persona revise tus cambios, comente y los acepte o no.

Pasos habituales:

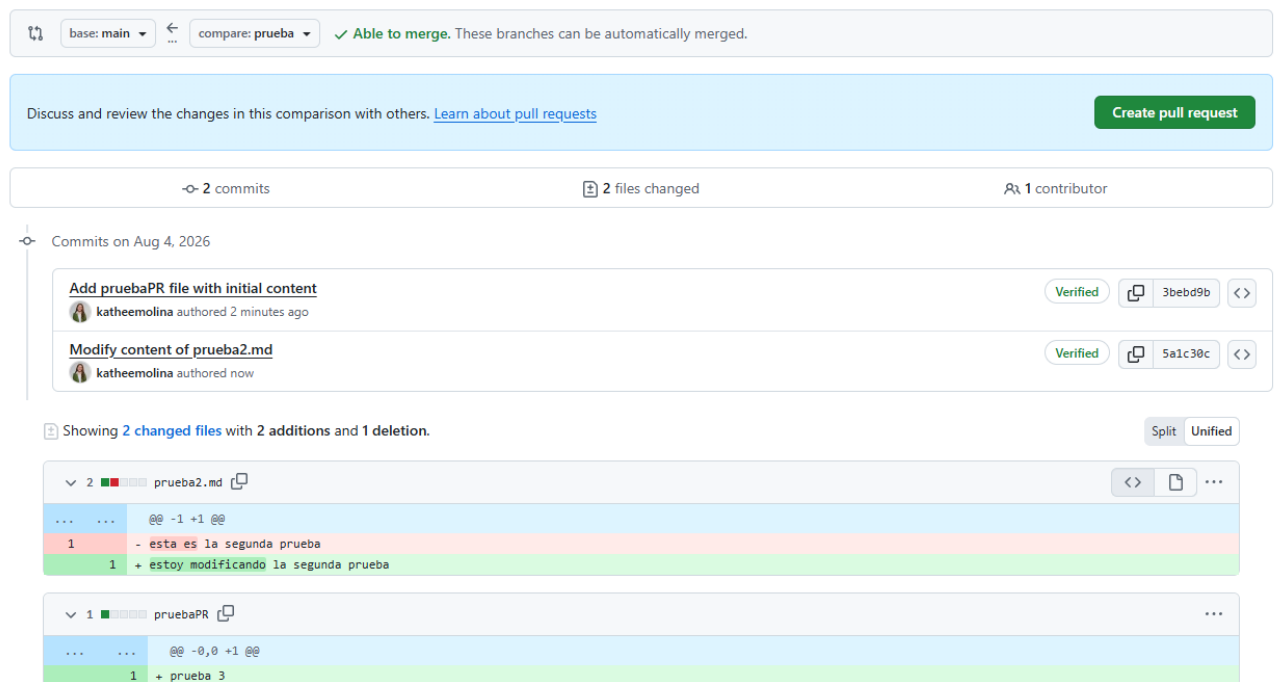
1. Hacer fork del repositorio original.
2. Hacer los cambios que quieras dentro de tu copia (podés crear una rama para eso).
3. Ir a la pestaña "Pull requests" y hacer clic en "New pull request".



4. Revisar los cambios propuestos y hacer clic en "Create pull request", agregando un mensaje que explique qué hiciste.
5. Esperar a que la persona dueña del repositorio original lo revise y lo acepte (merge).

Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also [compare across forks](#) or [learn more about diff comparisons](#).



16. Seguir personas y repositorios

Seguir a una persona (Follow)

Entrando al perfil de otro usuario o usuaria, vas a ver un botón "Follow". Al seguir a alguien, vas a poder ver sus novedades (por ejemplo, nuevos repositorios) en tu feed de GitHub, algo parecido a seguir a alguien en una red social.

Seguir un repositorio (Watch)

Dentro de cualquier repositorio hay un botón "Watch". Al activarlo, GitHub te va a avisar cuando haya novedades en ese proyecto (cambios, discusiones, etc.), aunque no seas colaborador.

17. Ponerle una estrella (Star) a un repositorio

El botón "Star" (con forma de estrellita) que aparece arriba a la derecha de cada repositorio sirve para marcarlo como favorito. Es útil para guardar proyectos que te interesan y encontrarlos fácilmente después, en la sección "Your stars" de tu perfil. Además, muchos repositorios usan la cantidad de estrellas como una forma de mostrar qué tan popular o útil es un proyecto.

18. Buscar repositorios en GitHub

Arriba de cualquier página de GitHub hay una barra de búsqueda. Ahí podés escribir palabras clave relacionadas con lo que necesitás (por ejemplo: "plantilla portfolio html css" o "juego javascript simple") para encontrar repositorios públicos creados por otras personas.

Algunos filtros útiles al buscar:

- Language: para filtrar por lenguaje (HTML, CSS, JavaScript, etc.).
- Sort → Most stars: para ver primero los proyectos más populares.
- Sort → Recently updated: para ver los proyectos más activos.

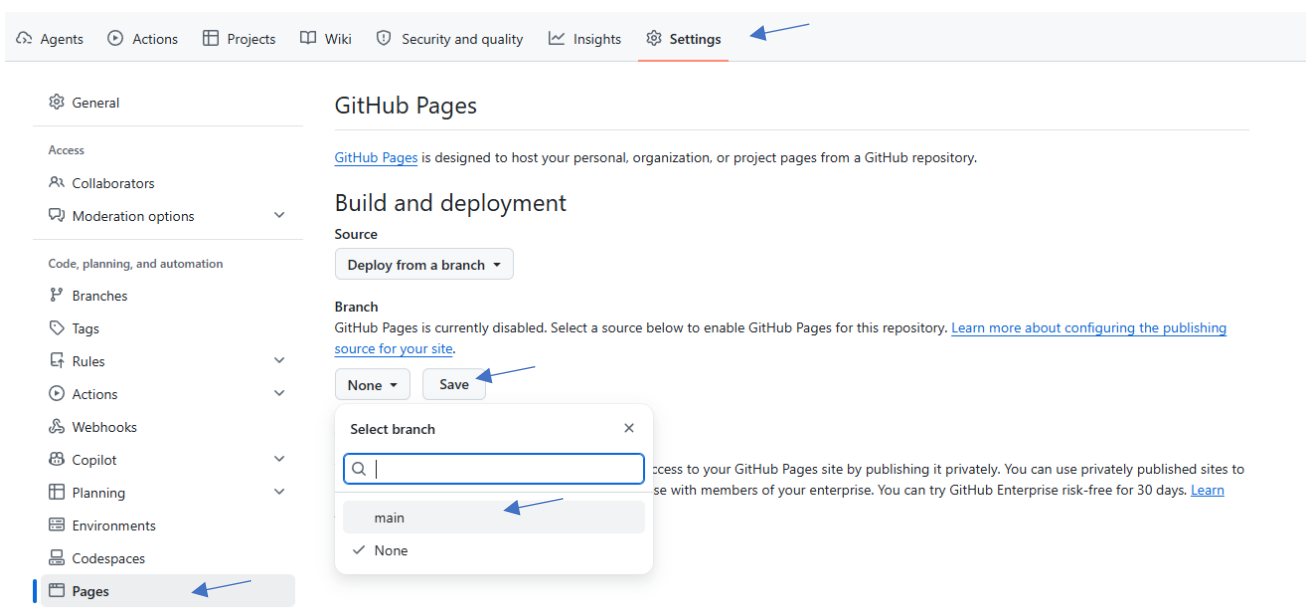
Buena práctica

Antes de copiar código de un repositorio ajeno, conviene revisar si tiene una licencia y qué permite hacer con ese código.

19. Publicar tu sitio con GitHub Pages

GitHub Pages es una función gratuita que permite publicar en internet un sitio hecho con HTML, CSS y JavaScript, usando directamente los archivos de tu repositorio.

1. Entrar al repositorio → pestaña "Settings".
2. En el menú de la izquierda, hacer clic en "Pages".
3. En "Branch", elegir la rama main y la carpeta / (root).
4. Hacer clic en "Save".
5. Esperar uno o dos minutos: GitHub va a mostrar la dirección donde quedó publicado el sitio (algo como `usuario.github.io/nombre-del-repositorio`).



Importante

El archivo principal de tu sitio debe llamarse exactamente index.html para que GitHub Pages lo reconozca como la página de inicio.

20. Glosario rápido

Término	Qué significa
Repositorio (repo)	La "carpeta" de un proyecto dentro de GitHub.
Commit	Una versión guardada de tus archivos, con un mensaje que describe el cambio.
Rama (branch)	Una copia paralela del proyecto para trabajar sin afectar la versión principal.
Merge	Combinar los cambios de una rama con otra (por ejemplo, con main).
Conflicto	Cuando GitHub no puede combinar automáticamente dos cambios sobre las mismas líneas.
Revert	Deshacer un commit creando uno nuevo que restaura el estado anterior.
.gitignore	Archivo que indica qué archivos no subir al repositorio.
Público / Privado	Quién puede ver el repositorio: cualquiera, o solo las personas invitadas.
Colaborador/a	Persona invitada a editar un repositorio (además del dueño/a).
Fork	Copia de un repositorio ajeno dentro de tu propia cuenta.
Pull Request (PR)	Solicitud para incorporar tus cambios al repositorio original.
Follow	Seguir a una persona para ver sus novedades en GitHub.
Watch	Recibir notificaciones sobre la actividad de un repositorio.
Star	Marcar un repositorio como favorito (estrellita).
GitHub Pages	Función para publicar gratis un sitio web a partir de un repositorio.
README	Archivo de texto que explica de qué trata el repositorio.