

# Github



# ¿Qué es?

GITHUB es un portal creado para alojar el código de los sitios y aplicaciones web de cualquier desarrollador, y que fue comprada por Microsoft en junio del 2018.

La plataforma está creada para que los desarrolladores suban el código web, y que como usuario no sólo puedas descargar la aplicación, sino también entrar a su perfil para leer sobre ella o colaborar con el desarrollo.

Existen 2 estilos: publico y privado

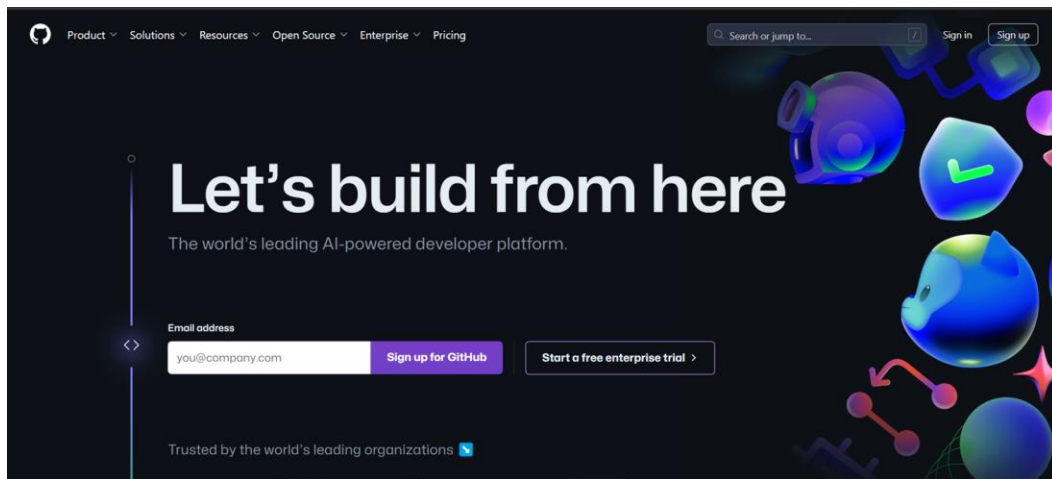
# ¿Ventajas?

- Colaboración: facilita la colaboración entre equipos de desarrollo, permitiendo que múltiples personas trabajen en un proyecto de manera simultánea. Las funciones de solicitud de extracción (pull requests) permiten revisar y discutir los cambios antes de que se integren en la versión principal del proyecto.
- Repositorios públicos y privados: ofrece la posibilidad de tener repositorios públicos (visibles para todos) o privados (acceso restringido). Esto es útil para proyectos de código abierto y para proyectos privados de empresas.
- Portafolio personal: Para los desarrolladores, GitHub sirve como una especie de portafolio en línea, donde pueden mostrar su trabajo, contribuciones a proyectos de código abierto y habilidades técnicas a posibles empleadores.

# ¿Cómo usarlo desde la web?

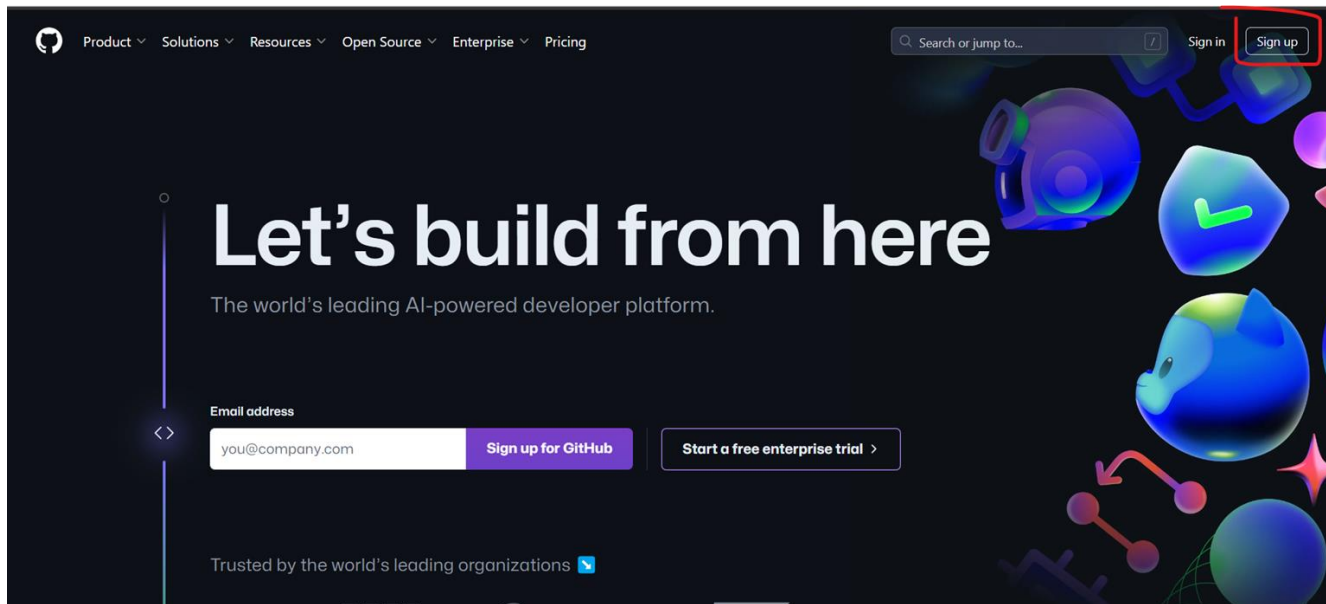
1- Se debe ingresar al siguiente enlace

<https://github.com/>



# ¿Cómo usarlo desde la web?

2- Damos clic a **Sign up**

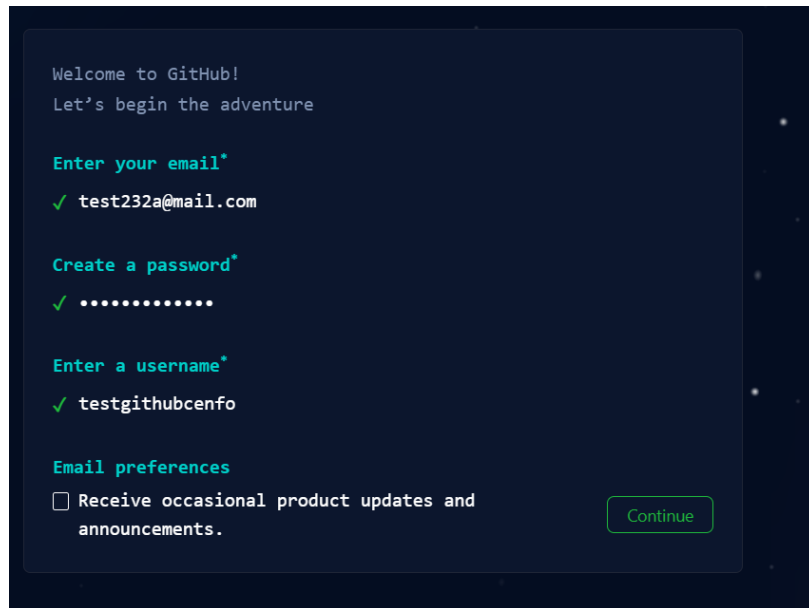


# ¿Cómo usarlo desde la web?

3- Nos creamos una cuenta con los siguientes datos:

- Email
- Contraseña
- Username
- El checkbox es opcional, si desean recibir novedades lo pueden marcar, de lo contrario lo dejan desmarcado

Una vez rellenado, damos en continuar



Wellcome to GitHub!  
Let's begin the adventure

Enter your email\*  
✓ test232a@mail.com

Create a password\*  
✓ .....

Enter a username\*  
✓ testgithubcenfo

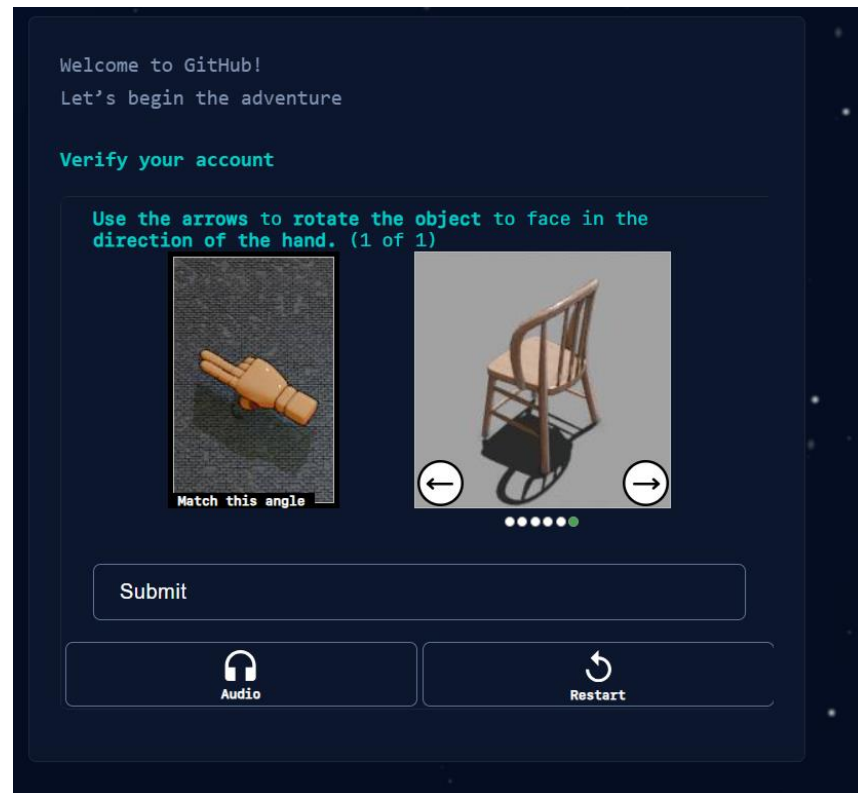
Email preferences  
☐ Receive occasional product updates and announcements.

Continue

# ¿Cómo usarlo desde la web?

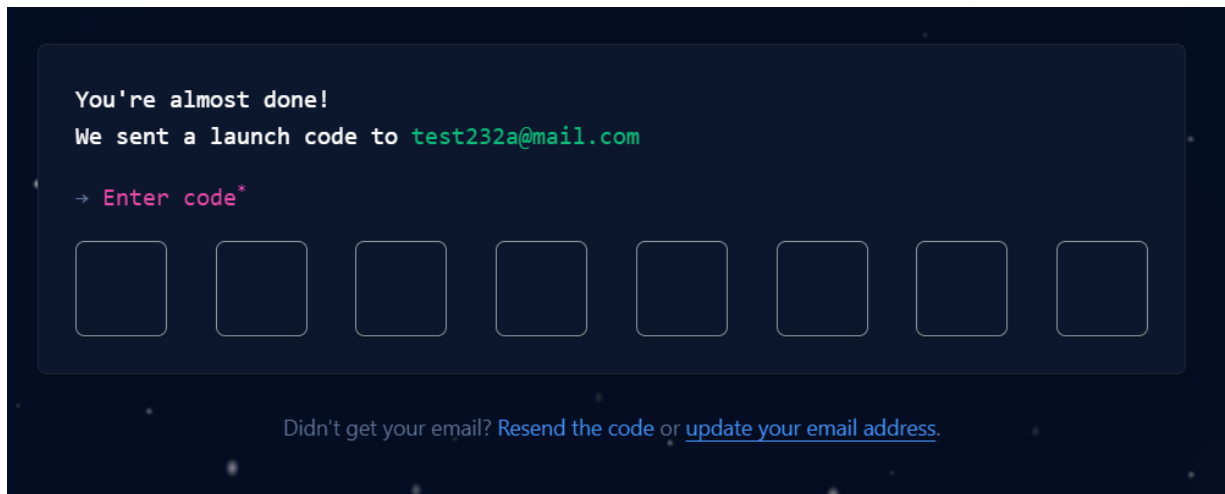
4- Debemos verificar nuestra cuenta armando el rompecabezas

Damos en submit una vez que estemos seguros



# ¿Cómo usarlo desde la web?

5- Colocamos el código que llegó a nuestro correo de registro para terminar de verificar la cuenta



You're almost done!

We sent a launch code to `test232a@mail.com`

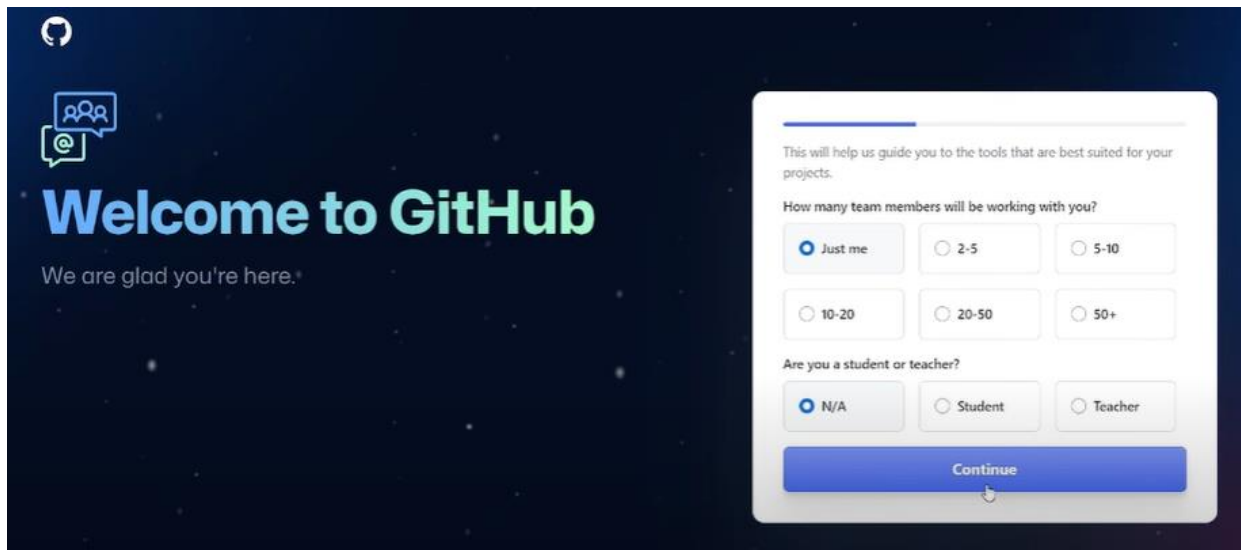
→ Enter code\*

Didn't get your email? [Resend the code](#) or [update your email address](#).



# ¿Cómo usarlo desde la web?

6- Rellenamos esta encuesta donde nos pregunta cuántos miembros hay en el equipo y que rol tiene



The image shows a screenshot of the GitHub onboarding survey. On the left, there is a dark blue background with the GitHub logo at the top left, a speech bubble icon with three people inside, and the text "Welcome to GitHub" in large, bold, light blue letters. Below this, it says "We are glad you're here." On the right, there is a white survey form with a blue border. The form contains the following text and options:

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

How many team members will be working with you?

☒ Just me    ☐ 2-5    ☐ 5-10

☐ 10-20    ☐ 20-50    ☐ 50+

Are you a student or teacher?

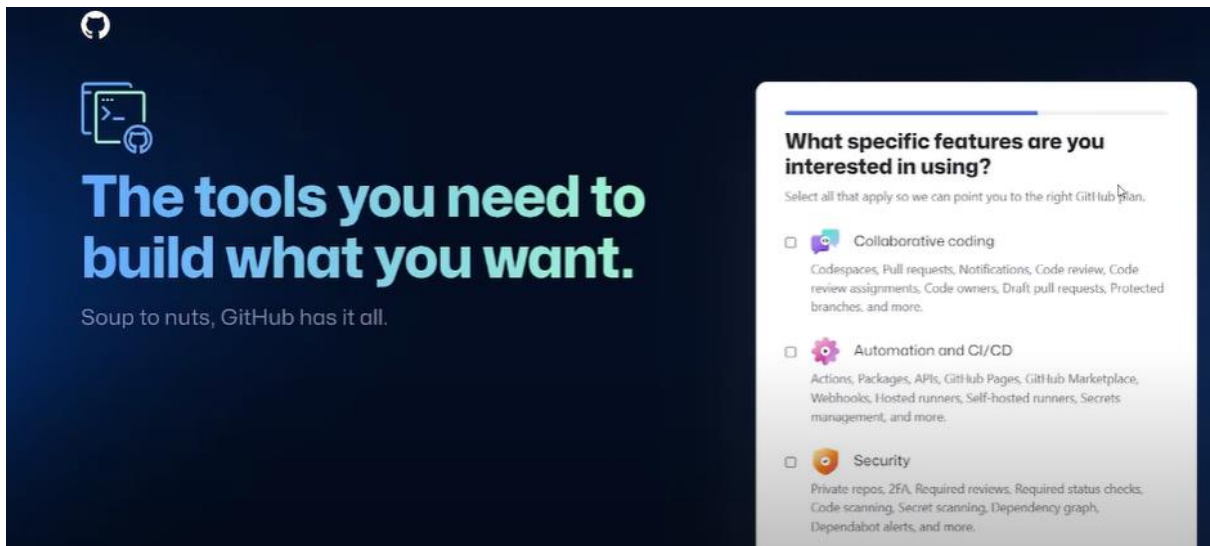
☒ N/A    ☐ Student    ☐ Teacher

At the bottom of the form is a large blue button labeled "Continue". A mouse cursor is pointing at the button.

# ¿Cómo usarlo desde la web?

7- Marcamos las características que nos interesan utilizar, podemos marcar

- Project management
- Community
- Collaborative coding



The screenshot displays the GitHub homepage. On the left, a dark blue banner features the GitHub logo, an icon of a code editor, and the text: "The tools you need to build what you want." followed by "Soup to nuts, GitHub has it all." On the right, a white sidebar titled "What specific features are you interested in using?" contains a list of features with checkboxes. The "Collaborative coding" option is checked, while "Automation and CI/CD" and "Security" are unchecked.

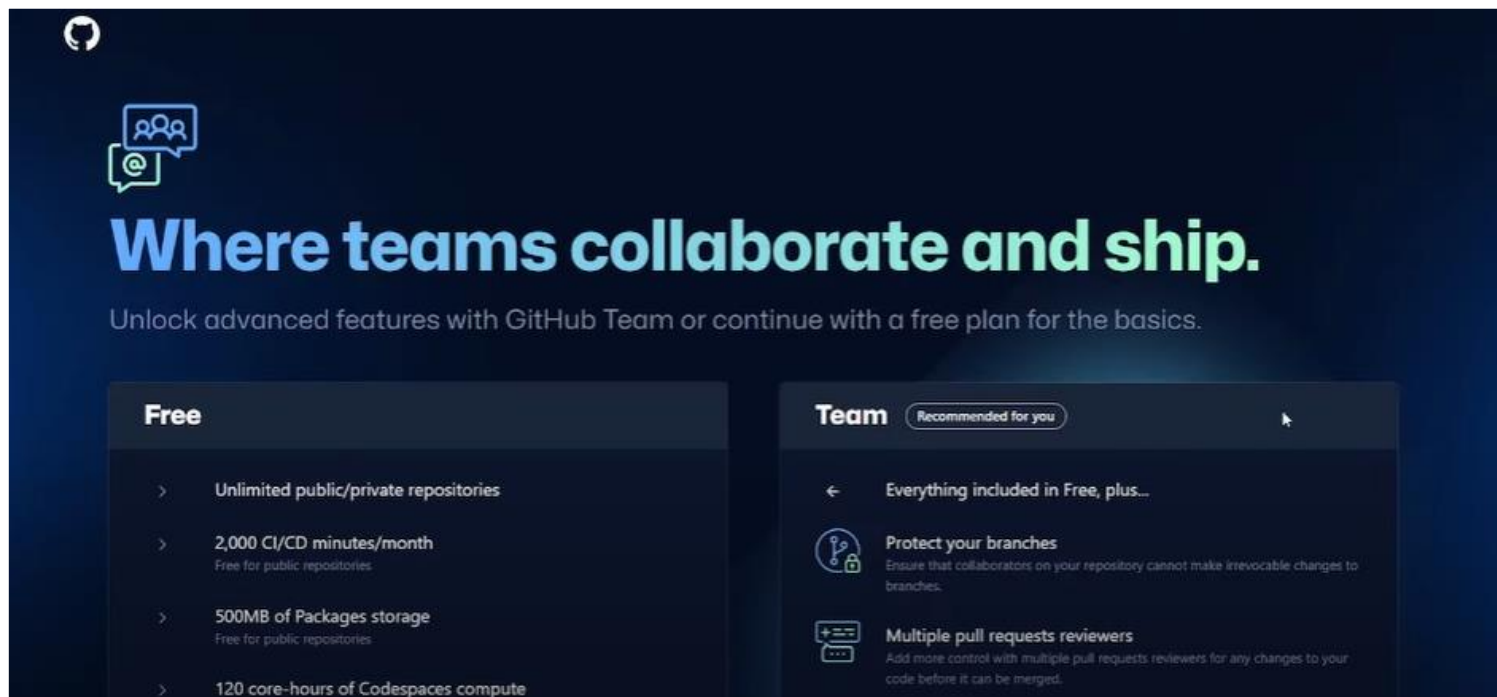
**The tools you need to build what you want.**  
Soup to nuts, GitHub has it all.

**What specific features are you interested in using?**  
Select all that apply so we can point you to the right GitHub Plan.

- ☒ **Collaborative coding**  
Codespaces, Pull requests, Notifications, Code review, Code review assignments, Code owners, Draft pull requests, Protected branches, and more.
- ☐ **Automation and CI/CD**  
Actions, Packages, APIs, GitHub Pages, GitHub Marketplace, Webhooks, Hosted runners, Self-hosted runners, Secrets management, and more.
- ☐ **Security**  
Private repos, 2FA, Required reviews, Required status checks, Code scanning, Secret scanning, Dependency graph, Dependabot alerts, and more.

# ¿Cómo usarlo desde la web?

8- Seleccionamos la versión gratuita





The screenshot shows the GitHub website's pricing page. At the top left is the GitHub logo. Below it is an icon representing collaboration (two people in a speech bubble). The main heading is "Where teams collaborate and ship." in a light blue font. Below this, a subtext says "Unlock advanced features with GitHub Team or continue with a free plan for the basics." There are two pricing plans displayed: "Free" and "Team". The "Free" plan is on the left and lists four features: "Unlimited public/private repositories", "2,000 CI/CD minutes/month" (with a note "Free for public repositories"), "500MB of Packages storage" (with a note "Free for public repositories"), and "120 core-hours of Codespaces compute". The "Team" plan is on the right, marked as "Recommended for you", and lists three features: "Everything included in Free, plus..." (with a left arrow icon), "Protect your branches" (with a lock icon and a note "Ensure that collaborators on your repository cannot make irrevocable changes to branches."), and "Multiple pull requests reviewers" (with a plus icon and a note "Add more control with multiple pull requests reviewers for any changes to your code before it can be merged.").

**Free**

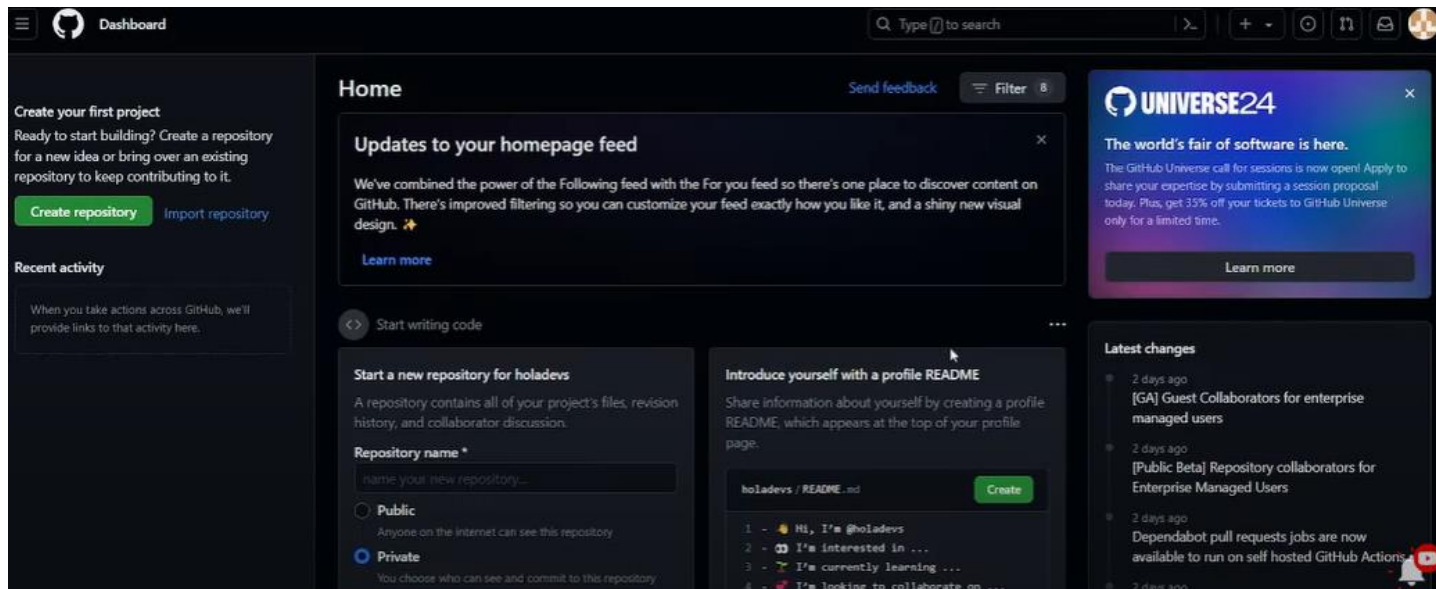
- > Unlimited public/private repositories
- > 2,000 CI/CD minutes/month  
Free for public repositories
- > 500MB of Packages storage  
Free for public repositories
- > 120 core-hours of Codespaces compute

**Team** Recommended for you

- ← Everything included in Free, plus...
-  **Protect your branches**  
Ensure that collaborators on your repository cannot make irrevocable changes to branches.
-  **Multiple pull requests reviewers**  
Add more control with multiple pull requests reviewers for any changes to your code before it can be merged.

# ¿Cómo usarlo desde la web?

9- Cuando hayamos seleccionado el plan gratuito, nos lleva al **Dashboard de Github**



The screenshot displays the GitHub Dashboard for a user. The interface is dark-themed and includes a top navigation bar with the GitHub logo, a search bar, and various utility icons. On the left sidebar, there are sections for 'Create your first project' with a 'Create repository' button, and 'Recent activity'. The main content area is titled 'Home' and features an 'Updates to your homepage feed' section. Below this, there are two primary action cards: 'Start a new repository for holadevs' and 'Introduce yourself with a profile README'. The 'Start a new repository' card includes a form for the repository name and options for visibility (Public or Private). The 'Introduce yourself' card shows a preview of a README file for the 'holadevs' repository. On the right side, there is a promotional banner for 'UNIVERSE24' and a 'Latest changes' section listing recent updates from the community.

# ¿Qué es un repositorio?

Es un espacio donde se almacena y gestiona un proyecto de desarrollo de software. Funciona como un contenedor para todo el código fuente, archivos, documentación, historial de cambios y cualquier otro recurso relacionado con un proyecto.



# ¿Puntos claves de un repositorio?

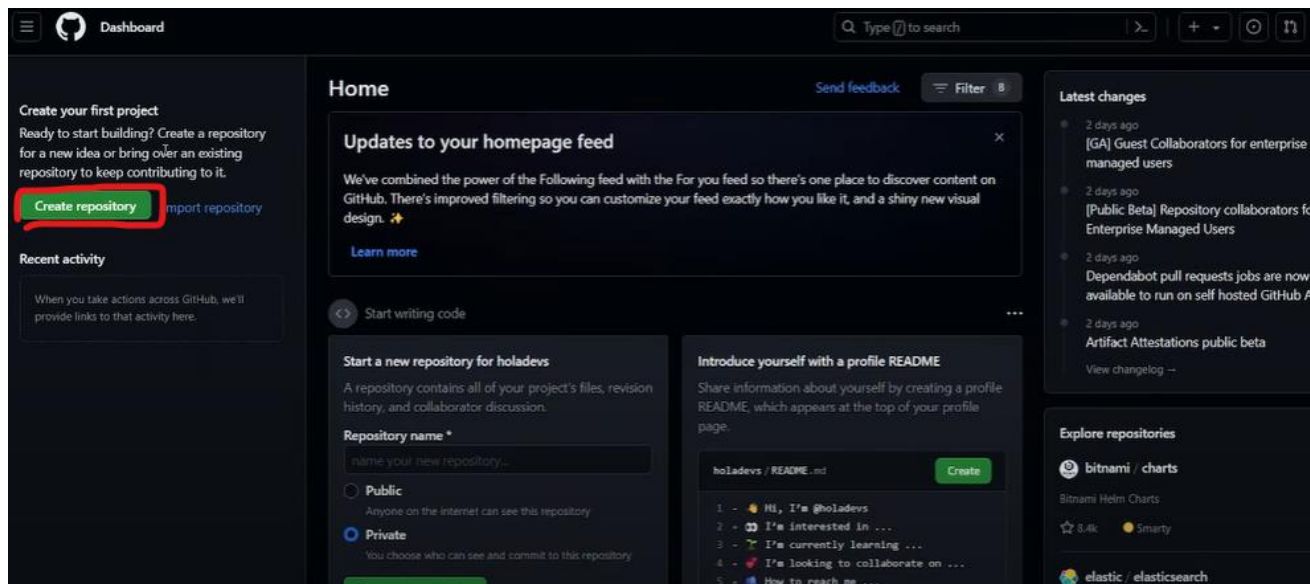
- Código fuente: El repositorio contiene el código del proyecto, organizado en archivos y carpetas
- Historial de versiones: Git, el sistema de control de versiones que GitHub utiliza, permite hacer un seguimiento de todos los cambios realizados en el código a lo largo del tiempo. Esto permite a los desarrolladores ver quién hizo qué cambios y cuándo, y revertir cambios si es necesario
- Colaboración: Los repositorios permiten que varios desarrolladores trabajen en el mismo proyecto al mismo tiempo. Pueden crear ramas (branches) para trabajar en nuevas características o correcciones sin afectar el código principal, y luego fusionar esos cambios cuando estén listos

# ¿Puntos claves de un repositorio?

- Issues y Pull Requests: Los repositorios de GitHub ofrecen herramientas para la gestión del proyecto, como issues para rastrear errores o solicitudes de características, y pull requests para revisar y discutir cambios antes de fusionarlos en la rama principal
- Documentación: Los repositorios a menudo incluyen archivos de documentación, como un archivo README.md, que describe el proyecto, cómo instalarlo y cómo contribuir
- Licencia: Un repositorio puede incluir un archivo de licencia que establece cómo se puede usar y distribuir el código

# ¿Cómo crear un repositorio?

1- En el home de Github, damos clic al botón para crear un nuevo repositorio





# ¿Cómo crear un repositorio?

2- Llenamos los campos y marcamos según la necesidad del proyecto.

En el nombre no es bueno dejar espacios

Los 3 últimos no son necesarios completarlos


## Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

*Required fields are marked with an asterisk (\*).*

Owner \*

Repository name \*

 test34521

 / 


Nombre del repositorio

⚠ New repository name must not be blank


Great repository names are short and memorable. Need inspiration? How about [literate-funicular](#) ?

Description (optional)

Descripción del proyecto

☒  **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.

Quien puede ver el proyecto?

Initialize this repository with:

☐ **Add a README file**

Sirve para añadir un archivo que tenga instrucciones para el repositorio

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

Add .gitignore

.gitignore template: **None**

Plantilla para evitar subir archivos de un proyecto generado

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

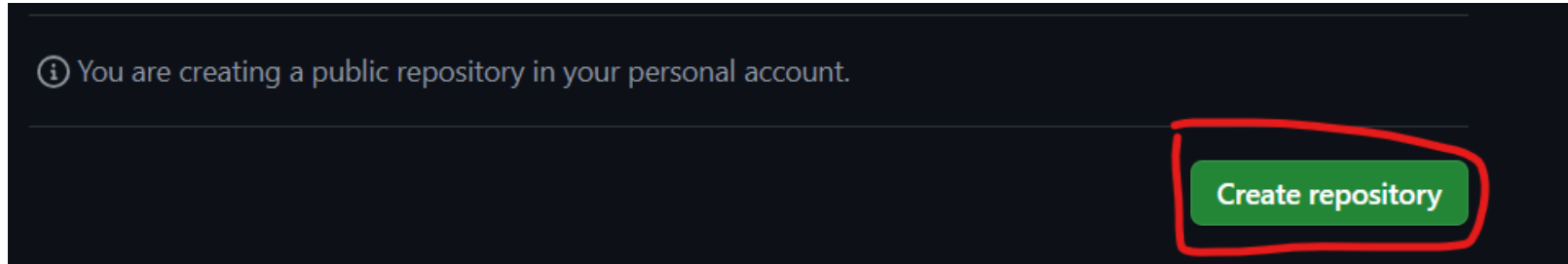
Choose a license

License: **None**

licencia de uso de nuestro proyecto

# ¿Cómo crear un repositorio?

3- Cuando completamos los campos, damos clic al botón “Crear repositorio”



A screenshot of the GitHub repository creation interface. The background is dark. At the top, there is a light gray informational message: "i You are creating a public repository in your personal account." Below this, there are several empty input fields for repository details. At the bottom right, there is a prominent green button with the text "Create repository" in white. This button is highlighted with a hand-drawn red rounded rectangle.

# ¿Cómo crear un repositorio?

4- Aparecerá esta ventana, esta ventana nos explica como hacer esto mismo pero con comandos desde terminal. En este caso dar clic en el link **Subir archivos existentes**

The screenshot shows the GitHub interface for a new repository named 'miPrimerRepositorio'. At the top, there are buttons for 'Pin', 'Unwatch' (1), 'Fork' (0), and 'Star' (0). Below this, there are two main sections: 'Set up GitHub Copilot' and 'Add collaborators to this repository'. The 'Set up GitHub Copilot' section includes a button 'Get started with GitHub Copilot'. The 'Add collaborators to this repository' section includes a button 'Invite collaborators'. Below these sections, there is a 'Quick setup' section with a dropdown menu showing 'Set up in Desktop' or 'HTTPS' or 'SSH', and a text input field containing 'https://github.com/test34522/miPrimerRepositorio.git'. Below the dropdown, there is a link 'Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).' At the bottom, there is a section titled '...or create a new repository on the command line' with a code block containing the following commands: 

```
echo "# miPrimerRepositorio" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/Dablolimenez/miPrimerRepositorio.git
```

miPrimerRepositorio Public

Pin Unwatch 1 Fork 0 Star 0

**Set up GitHub Copilot**  
Use GitHub's AI pair programmer to autocomplete suggestions as you code.  
[Get started with GitHub Copilot](#)

**Add collaborators to this repository**  
Search for people using their GitHub username or email address.  
[Invite collaborators](#)

**Quick setup — if you've done this kind of thing before**

[Set up in Desktop](#) or [HTTPS](#) or [SSH](#)

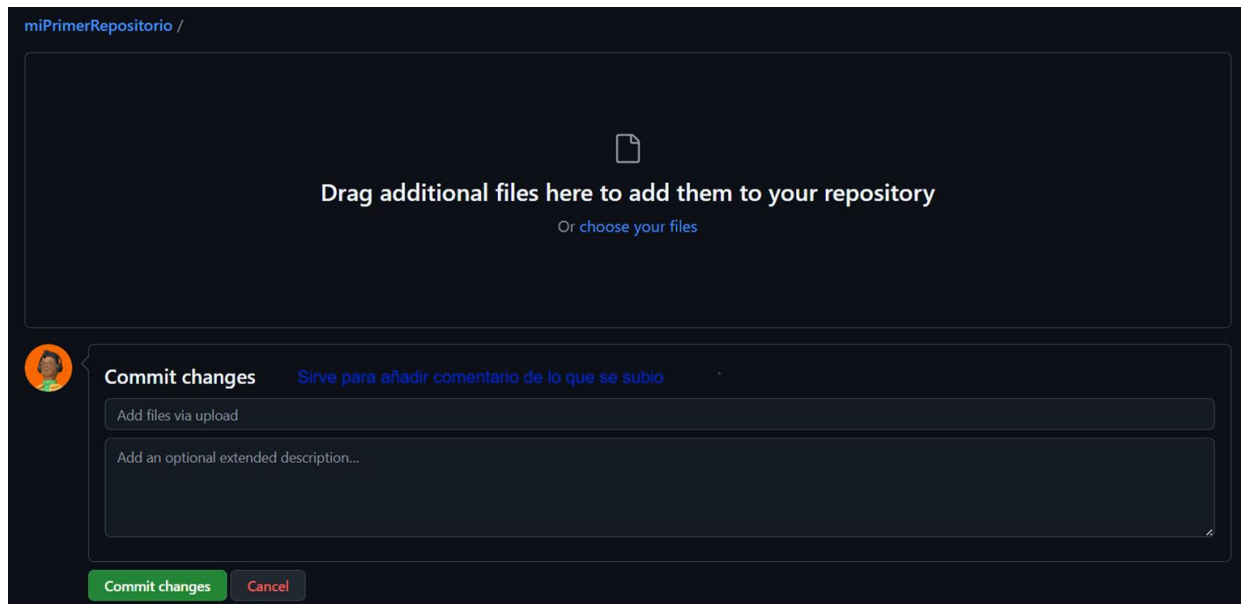
Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

**...or create a new repository on the command line**

```
echo "# miPrimerRepositorio" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/Dablolimenez/miPrimerRepositorio.git
```

# ¿Cómo crear un repositorio?

5- Se suben los archivos del proyecto, en caso contrario, si están dentro de una carpeta, se publica el folder.




The screenshot shows a dark-themed GitHub interface. At the top, the breadcrumb 'miPrimerRepositorio /' is visible. Below it is a large rectangular area with a file icon and the text 'Drag additional files here to add them to your repository' and 'Or choose your files'. At the bottom, a 'Commit changes' dialog box is open, featuring a user profile icon, the title 'Commit changes', a description field, and two buttons: 'Commit changes' and 'Cancel'.

miPrimerRepositorio /

Drag additional files here to add them to your repository

Or choose your files

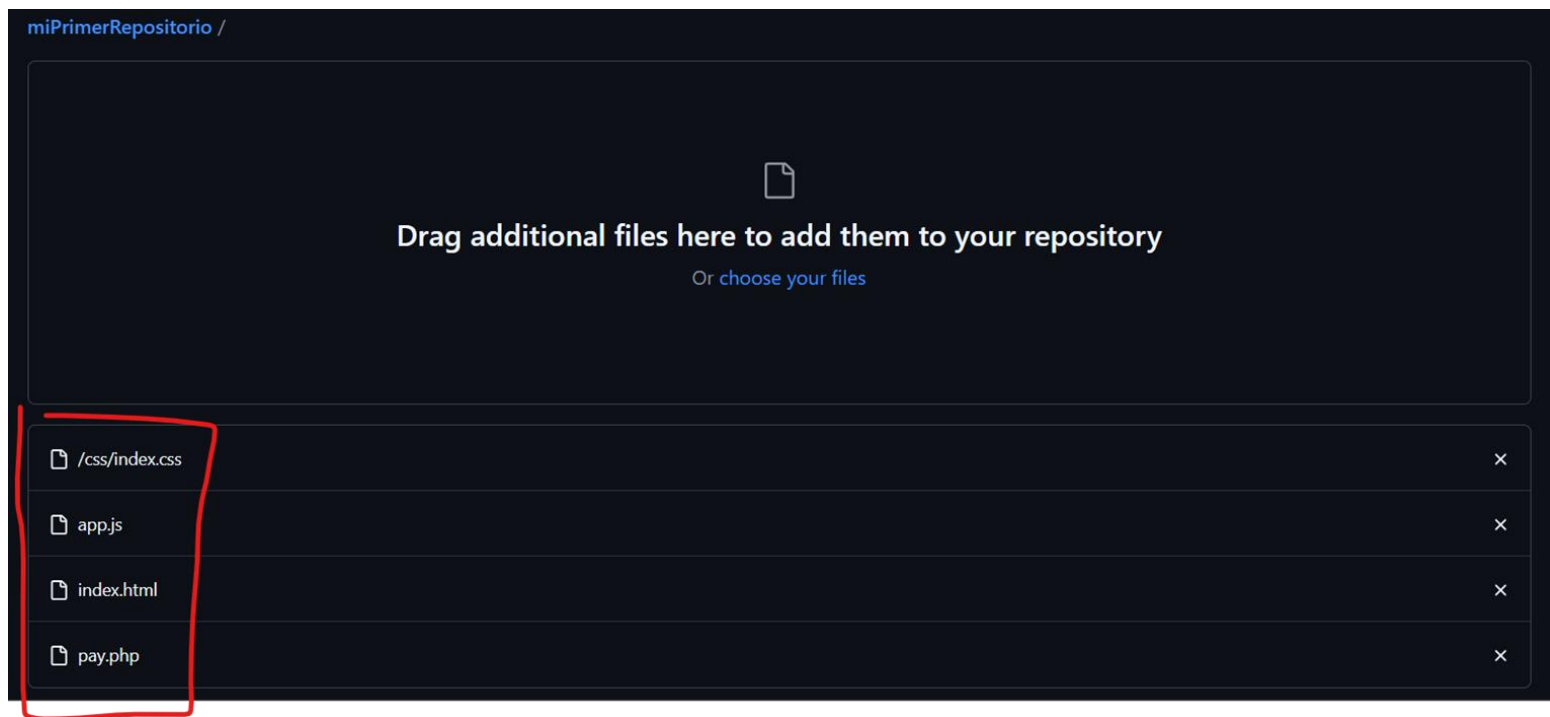
 **Commit changes** [Sirve para añadir comentario de lo que se subio](#)

Add files via upload

Add an optional extended description...

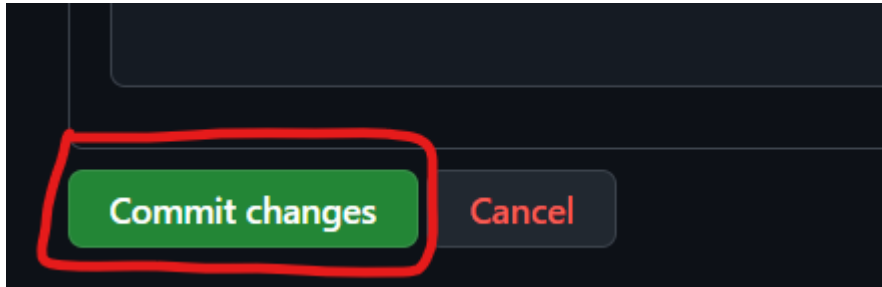
**Commit changes** **Cancel**

# ¿Cómo crear un repositorio?



# ¿Cómo crear un repositorio?

6- Una vez subidos, damos al botón **Commit changes**



# ¿Cómo crear un repositorio?

7- Una vez ejecutado el botón, nos llevará a esta página donde ya aparecerá nuestro proyecto con todos los archivos.

The screenshot shows a GitHub repository page for 'miPrimerRepositorio' (Public). The repository is currently on the 'main' branch, with 1 branch and 0 tags. The repository was created by user 'test3455343' and has 1 commit. The repository contains four files: 'css', 'app.js', 'index.html', and 'pay.php', each added via upload 1 minute ago. A 'README' file is also present, with a large 'Add a README' button and the text 'Help people interested in this repository understand your project by adding a README.' The right sidebar shows the repository's 'About' section, including activity, 0 stars, 1 watching, and 0 forks. The 'Releases' section shows no releases published, and the 'Packages' section shows no packages published.

miPrimerRepositorio Public

main 1 Branch 0 Tags

Go to file Add file <> Code

test3455343 Add files via upload 9ad9554 · 1 minute ago 1 Commit

css	Add files via upload	1 minute ago
app.js	Add files via upload	1 minute ago
index.html	Add files via upload	1 minute ago
pay.php	Add files via upload	1 minute ago

README

Add a README

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

About mi primer repositorio

Activity

0 stars

1 watching

0 forks

Releases

No releases published

[Create a new release](#)

Packages

No packages published

[Publish your first package](#)

# ¿Cómo borrar un repositorio?

1- Damos clic a nuestro perfil individual.





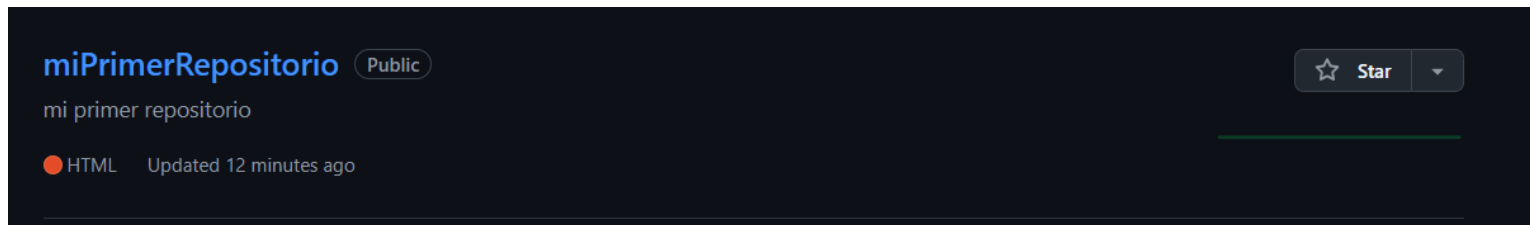
# ¿Cómo borrar un repositorio?

2- Ingresamos a **Your repositories**



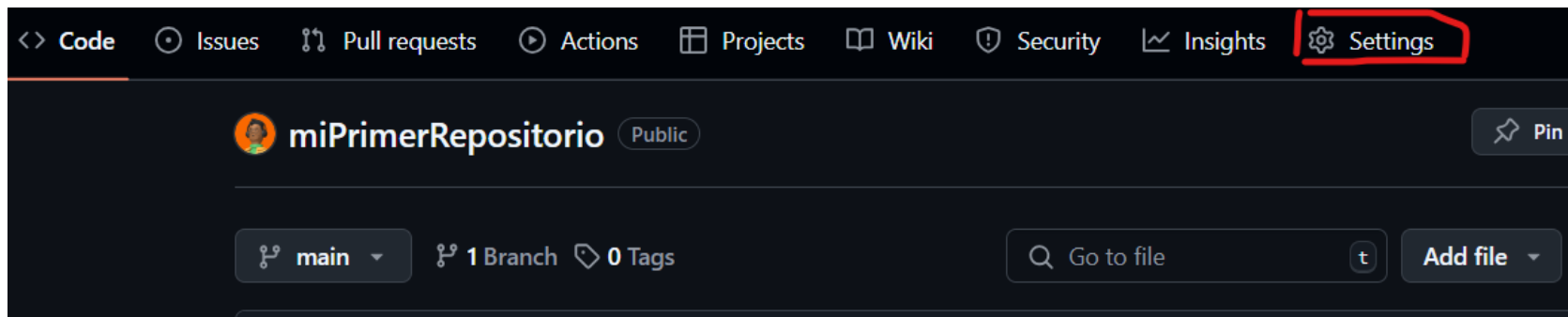
# ¿Cómo borrar un repositorio?

3- Buscamos e ingresamos al repositorio que queremos eliminar



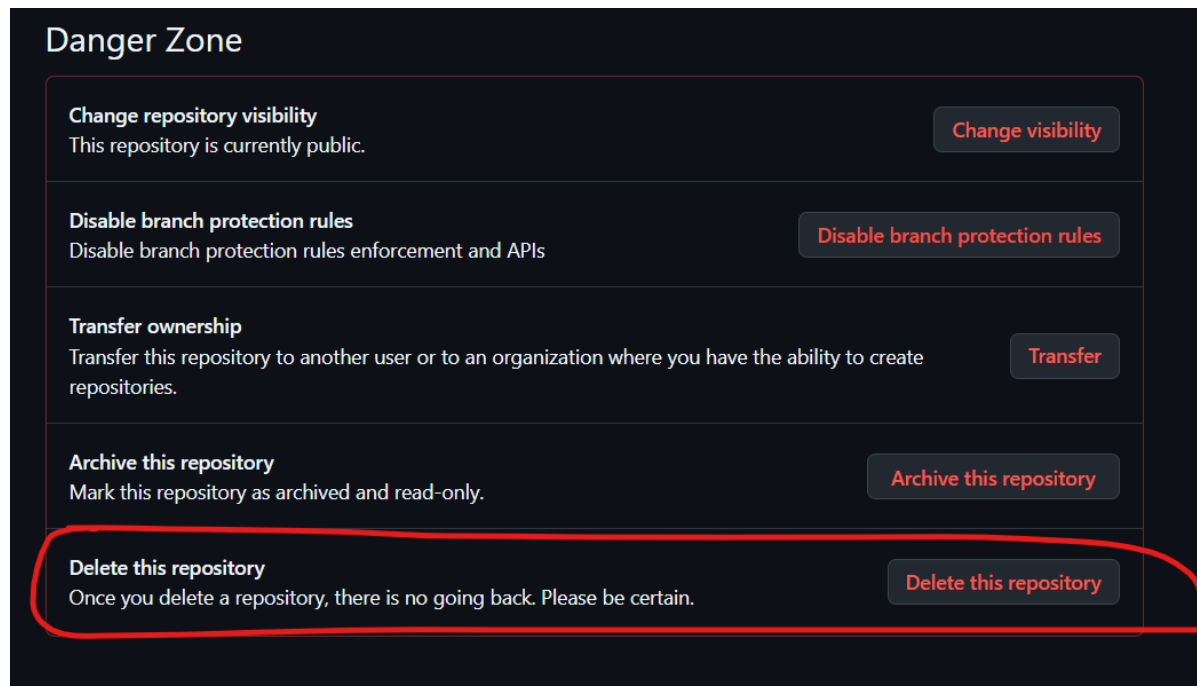
# ¿Cómo borrar un repositorio?

4- Una vez dentro, damos en **Settings**



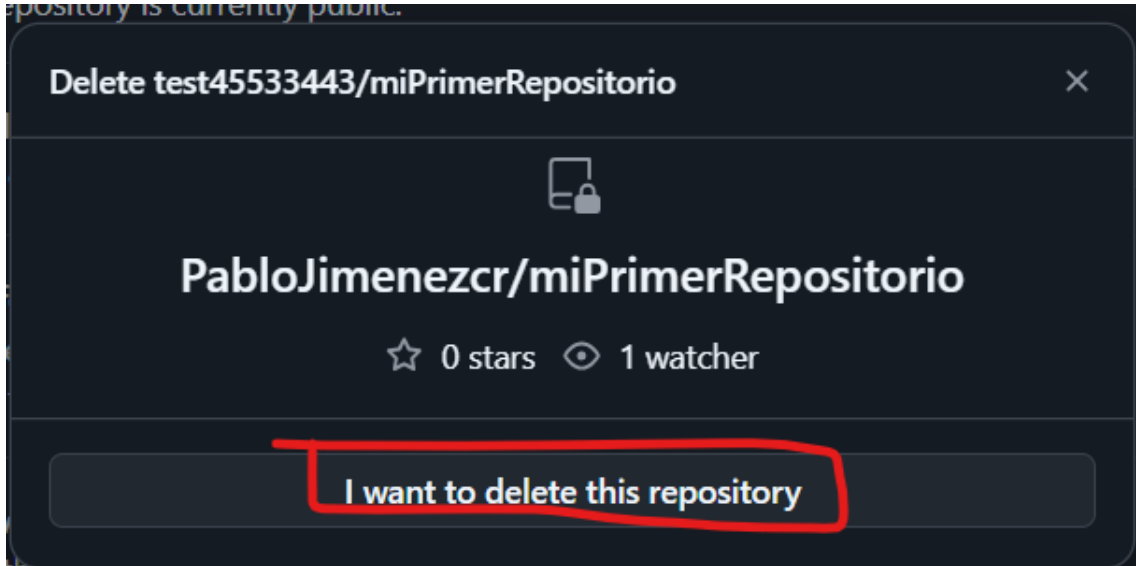
# ¿Cómo borrar un repositorio?

5- En la misma pestaña que nos mandó, en general, hacemos scroll hasta el final. Damos clic a **Delete this repository**



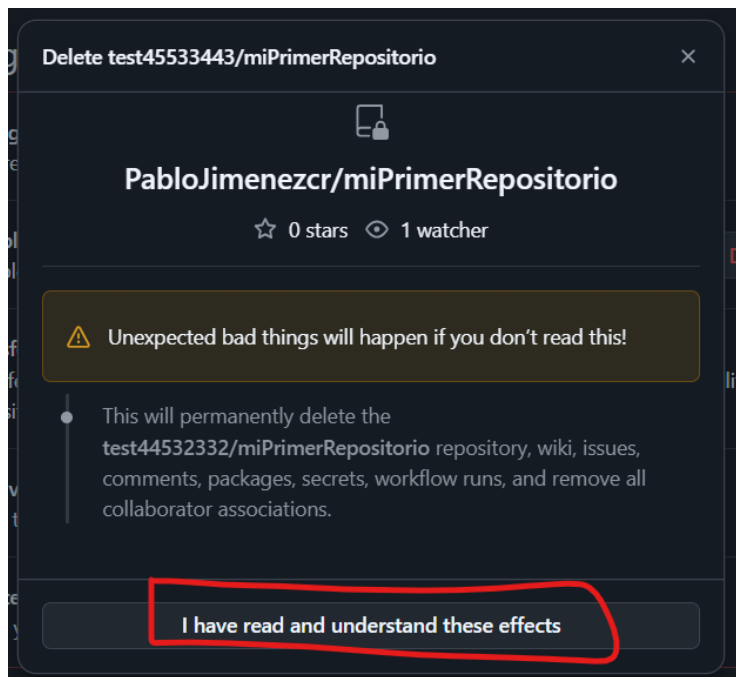
# ¿Cómo borrar un repositorio?

6- Nos aparecerá este modal, damos clic al botón **I want to delete this repository**



# ¿Cómo borrar un repositorio?

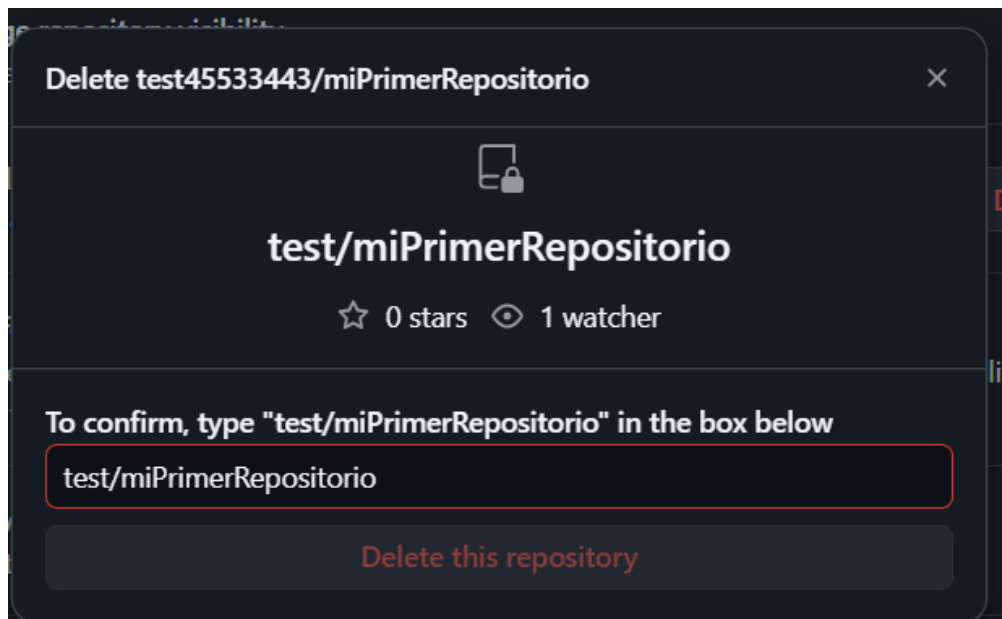
7- Aparecerá otro modal, confirmamos que si queremos eliminarlo.



# ¿Cómo borrar un repositorio?

8- Aparecerá otro modal , debemos escribir lo que nos dice en comillas dobles para confirmar la eliminación.

Una vez escrito damos al botón eliminar



# ¿Cómo borrar un repositorio?

9- Aparecerá una alerta de que se eliminó correctamente el repositorio





# ¿Qué es Github Pages?

Es un servicio proporcionado por GitHub que permite a los usuarios alojar sitios web estáticos directamente desde un repositorio de GitHub. Es una herramienta popular para crear y compartir sitios web personales, blogs, documentación de proyectos, portafolios y cualquier otro tipo de contenido que no requiere procesamiento del lado de un servidor.



# ¿Características?

- Alojamiento gratuito: GitHub Pages permite a los usuarios publicar sitios web de manera gratuita utilizando su propio dominio de GitHub (username.github.io) o un dominio personalizado.
- Sitios web estáticos: GitHub Pages está diseñado para alojar sitios web estáticos, es decir, sitios que consisten en archivos HTML, CSS, JavaScript, e imágenes. No es adecuado para aplicaciones web que requieren backend o procesamiento en el servidor.
- Integración con repositorios: GitHub Pages está directamente vinculado a los repositorios de GitHub. Los archivos que se encuentran en la rama principal del repositorio (o en una rama específica configurada para GitHub Pages) se utilizan para construir y mostrar el sitio web

# ¿Características?

- Soporte para Jekyll: GitHub Pages es compatible con Jekyll, un generador de sitios estáticos que facilita la creación de sitios web a partir de plantillas y contenido en formato Markdown. Esto es útil para crear blogs y documentación que se actualizan automáticamente al hacer commit en el repositorio.
- Implementación automática: Cada vez que se realiza un commit en el repositorio configurado para GitHub Pages, el sitio web se actualiza automáticamente. Esto hace que sea muy sencillo mantener el contenido actualizado.
- Personalización: Puede personalizar completamente tu sitio web, desde los temas y estilos hasta el uso de dominios personalizados.

# ¿Cómo implementar?

1- Creamos un repositorio nuevo y subimos los archivos del proyecto.


## Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

*Required fields are marked with an asterisk (\*).*


Owner \*

Repository name \*

 test

 / 


miPrimerWeb


 miPrimerWeb is available.

Great repository names are short and memorable. Need inspiration? How about [refactored-train](#) ?

**Description** (optional)

Web subida desde github pages

☒  **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:**

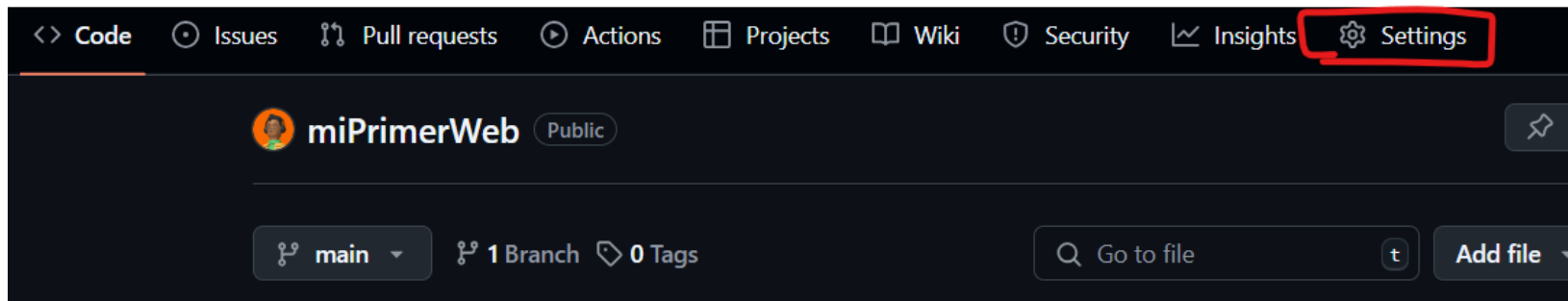
☐ **Add a README file**  
This is where you can write a long description for your project. [Learn more about READMEs.](#)

**Add .gitignore**

.gitignore template: **None**

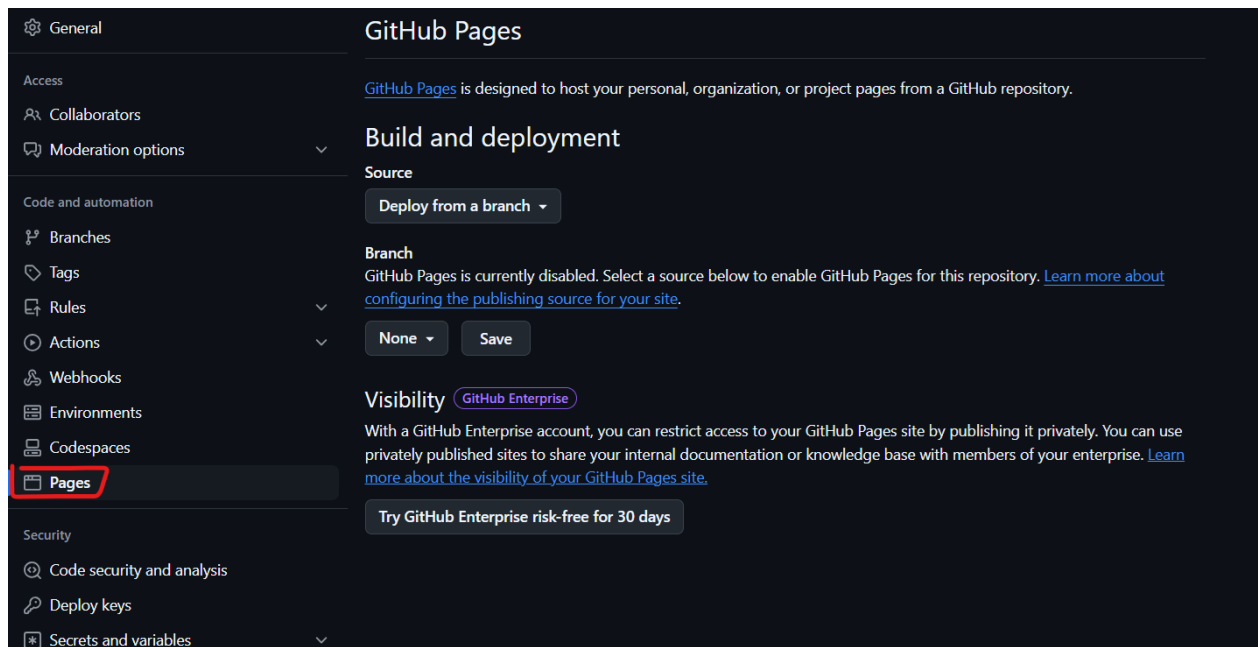
# ¿Cómo implementar?

2- Ingresamos a settings



# ¿Cómo implementar?

## 3- Ingresamos a la opción Pages



**General**

Access

- Collaborators
- Moderation options

Code and automation

- Branches
- Tags
- Rules
- Actions
- Webhooks
- Environments
- Codespaces
- Pages**

Security

- Code security and analysis
- Deploy keys
- Secrets and variables

### GitHub Pages

[GitHub Pages](#) is designed to host your personal, organization, or project pages from a GitHub repository.

### Build and deployment

**Source**

Deploy from a branch

**Branch**

GitHub Pages is currently disabled. Select a source below to enable GitHub Pages for this repository. [Learn more about configuring the publishing source for your site.](#)

None Save

**Visibility** [GitHub Enterprise](#)

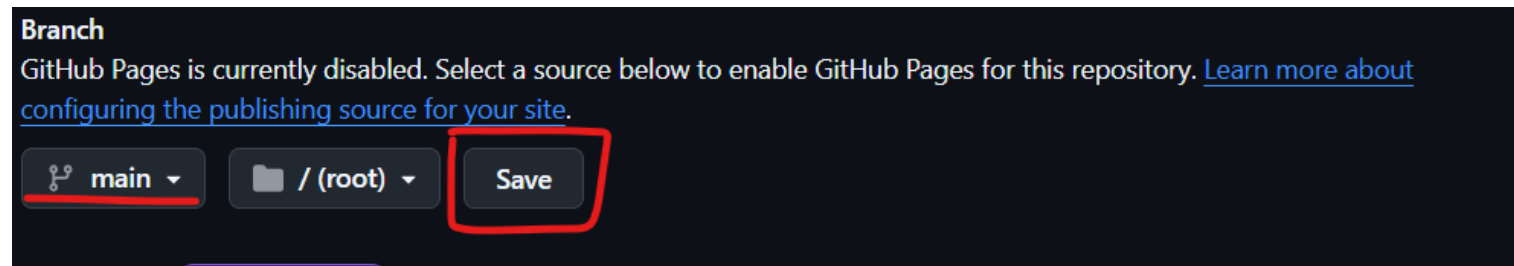
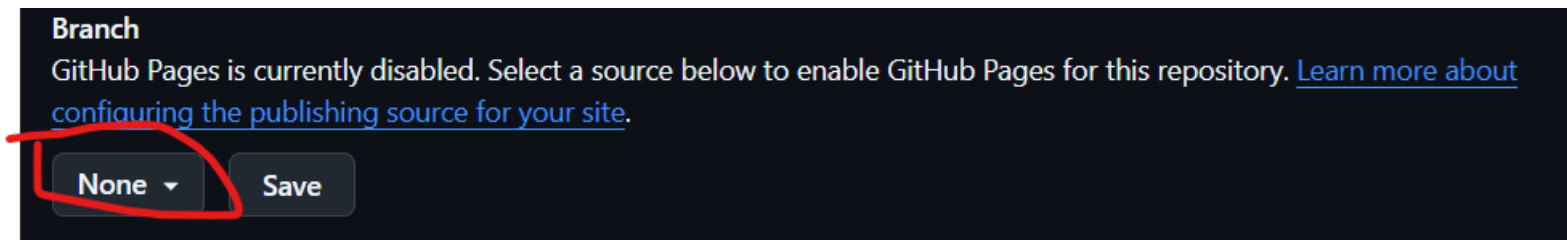
With a GitHub Enterprise account, you can restrict access to your GitHub Pages site by publishing it privately. You can use privately published sites to share your internal documentation or knowledge base with members of your enterprise. [Learn more about the visibility of your GitHub Pages site.](#)

Try GitHub Enterprise risk-free for 30 days

# ¿Cómo implementar?

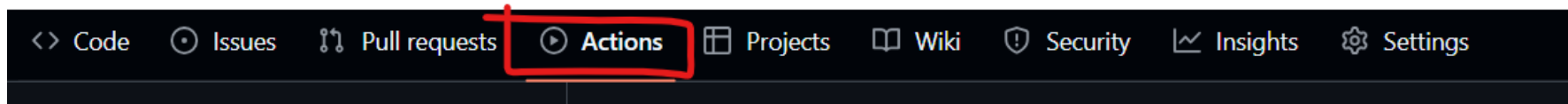
4- En Branch Damos clic en none y cambiamos por “main”.

Clic al botón Save

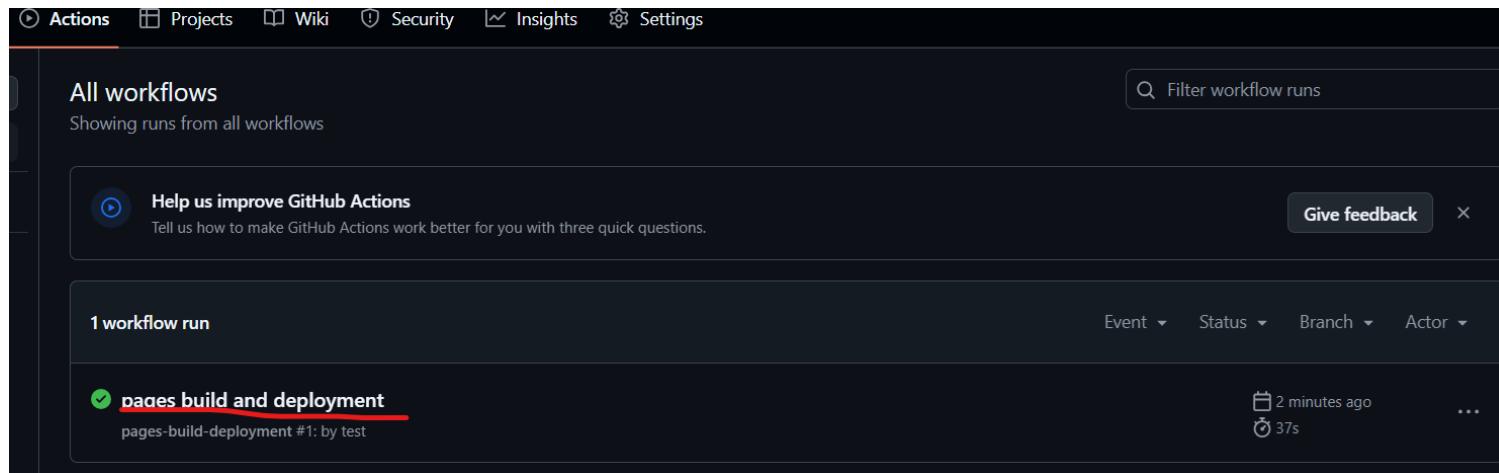


# ¿Cómo implementar?

5- Ingresamos a Action para ver el proceso, una vez se haya completado el deployment aparecerá en verde,



Ingresamos





# ¿Cómo implementar?

6- Nos aparecerá un tipo de rama, damos clic al que contiene el enlace que es el generado por Github para alojar nuestro proyecto en su nube.

The screenshot shows a GitHub Actions workflow run for the 'pages-build-deployment' job, triggered via dynamic. The workflow consists of three steps: 'build' (16s), 'report-build-status' (5s), and 'deploy' (8s). The 'deploy' step includes a URL: <https://test.github.io/miPrimerWeb/>.

Triggered via dynamic	Status	Total duration	Artifacts
test -> d99daa5	Success	37s	1

**pages-build-deployment**  
on: dynamic

- ✓ build 16s
- ✓ report-build-status 5s
- ✓ deploy 8s  
<https://test.github.io/miPrimerWeb/>

# ¿Cómo implementar?

7- Una vez ingresado se ve correctamente el proyecto y la URL contiene nuestro nombre de `usuario.github.io/nombreRepositorio`

