

# HTML

“html es un Lenguaje de Marcado de Hipertexto (HTML) es el código que se utiliza para estructurar y desplegar una página web y sus contenidos. Por ejemplo, sus contenidos podrían ser párrafos, una lista con viñetas, o imágenes y tablas de datos.”

# Etiquetas

Cuando escribes código en HTML, estás escribiendo etiquetas HTML. Todas las etiquetas HTML están hechas con un número de partes específicas, incluyendo:

- El carácter “menor que” <.
- Una palabra o carácter que determina qué etiqueta se está escribiendo.
- Cualquier número de atributos HTML que se quiera usar, escritos de la forma nombre =”valor”.
- El carácter “mayor que” >.

En la actualidad existen **142 etiquetas** HTML disponibles que permiten la creación de varios elementos. Algunos ordenadores nuevos ya no admiten algunas de estas.

Se escriben en formato `<etiqueta atributos>` contenido `</etiqueta>`

# Algunas etiquetas

- `<!DOCTYPE html>` Indica el tipo de documento
- `<html>` Contiene todo el documento
- `<head>` Contiene el encabezado
- `<body>` Contiene el cuerpo
- `<h1>` a `<h6>` Texto de encabezado, el número indica tamaño
- `<p>` Párrafo
- `<style>` Se usa para aplicar estilos CSS
- `<footer>` Contiene al pie de página

## Estilos CSS, inline

- Inline usa el atributo `style` en las etiquetas HTML.

## Estilos CSS, internal

- Internal usa la etiqueta `<style>` dentro de `<head>` para definir el estilo de la página
- Inline tiene “prioridad” sobre internal.

# Ejemplo Inline

```
<h1 style="color:yellow;">Un titulo  
amarillo</h1>
```

```
<h2>Un texto negro</h2>
```

```
<p style="color:red;">Un texto rojo</p>
```

**Un titulo amarillo**

**Un texto negro**

Un texto rojo

# Ejemplo Internal

```
<head>
<style>
body {background-color: #f15414;}
h1 {color: green;}
h2 {color: blue;}
</style>
</head>
<body>
<h1>Un titulo verde</h1>
<h2
style="color:white;background-color:black;">
Un texto blanco</h2>
</body>
```



<pre>&lt;!DOCTYPE html&gt; &lt;html&gt;  &lt;head&gt; &lt;title&gt;El titulo de la pagina&lt;/title&gt;  &lt;style&gt; body {background-color: black;} h1  {color: yellow;} h2  {color: white;} p   {color: #f15414;} &lt;/style&gt;  &lt;/head&gt;...</pre>	<pre>&lt;body&gt;  &lt;h1&gt;Un titulo amarillo&lt;/h1&gt;  &lt;h2&gt;Un texto blanco&lt;/h2&gt;  &lt;p&gt;Un texto rojo&lt;/p&gt;  &lt;footer&gt; &lt;p style="color:white;"&gt; El fondo&lt;/p&gt; &lt;/footer&gt; ...</pre>	<div>El titulo de la pagina</div> <div>Un titulo amarillo</div> <div>Un texto blanco</div> <div>Un texto rojo</div> <div>El fondo</div> <div>...</div> <div>&lt;/body&gt;</div> <div>&lt;/html&gt;</div>
--	--	--

# GIT

“Git es un sistema de control de versiones distribuido, lo que significa que un clon local del proyecto es un repositorio de control de versiones completo. Estos repositorios locales plenamente funcionales permiten trabajar sin conexión o de forma remota con facilidad.”



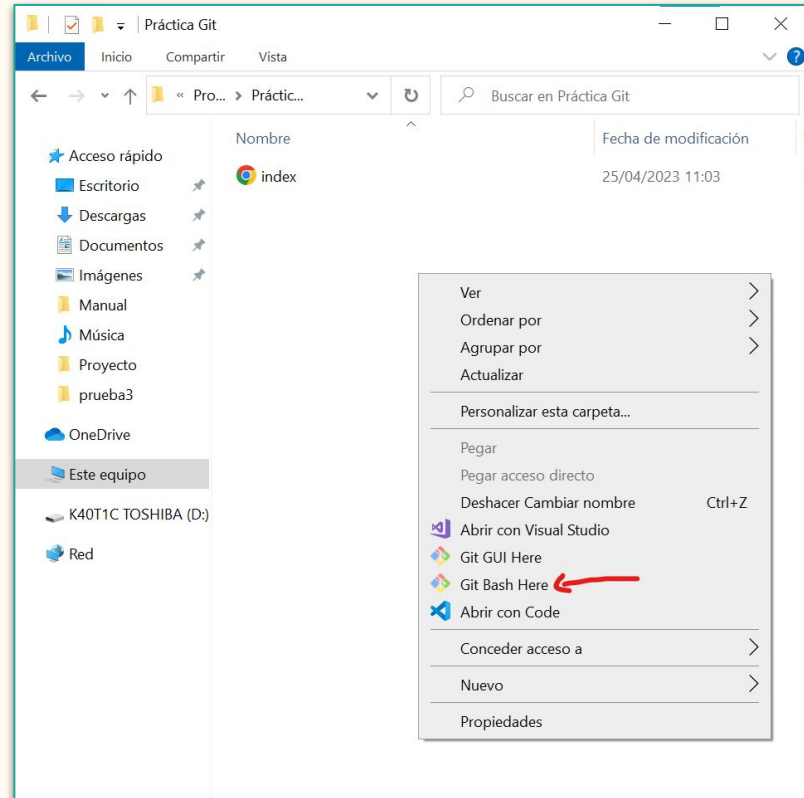
# ETIQUETAS EN GIT

El **etiquetado** es un **mecanismo adicional** que sirve para crear una **instantánea** de un **repositorio de Git**. Tradicionalmente, se utiliza el etiquetado para crear etiquetas identificativas con números de versión semánticos que se corresponden con ciclos de publicación de software. El comando **git tag** es el impulsor principal de una etiqueta: **creación, modificación y eliminación**.

Hay dos tipos de etiquetas: **anotadas** y **ligeras**. Por lo general, las etiquetas anotadas son la práctica recomendada, ya que almacenan metadatos valiosos adicionales sobre la etiqueta

# **COMO CREAR UN REPOSITORIO EN GIT**

**Paso 1.** Haremos click derecho en el directorio donde está ubicado index.html y seleccionaremos **Git Bash Here**




**Paso 2.** Se abrirá git y pondremos el comando **git init**

```
Gabriel FM@DESKTOP-2073SMD MINGW64 /c/Proyectos/Práctica Git  
$ git init  
Initialized empty Git repository in C:/Proyectos/Práctica Git/.git/  
  
Gabriel FM@DESKTOP-2073SMD MINGW64 /c/Proyectos/Práctica Git (main)  
$ |
```

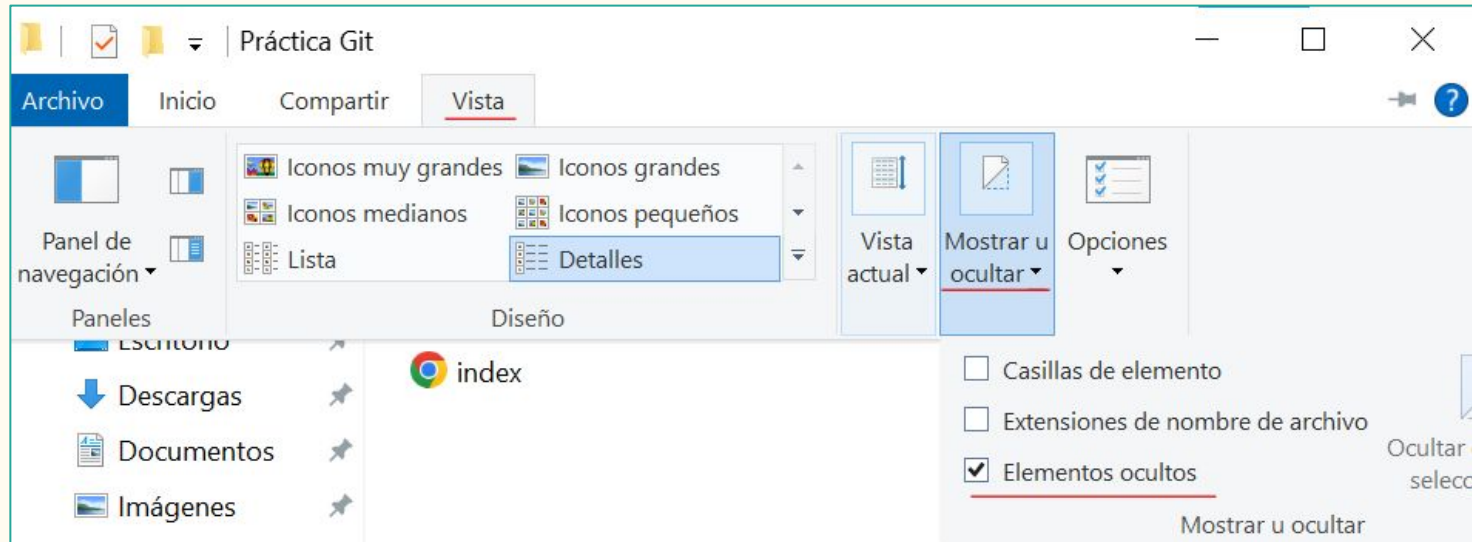
### Paso 3. Ejecutaremos el comando **git add index.html**

```
Gabriel FM@DESKTOP-2073SMD MINGW64 /c/Proyectos/Práctica Git (main)  
$ git add index.html
```



```
Gabriel FM@DESKTOP-2073SMD MINGW64 /c/Proyectos/Práctica Git (main)  
$
```

## Opcional. Habilitar carpetas ocultas para ver el proyecto git en **standing area**



# Aquí subiremos los dos cambios al repertorio

**Paso 1.** Ejecutaremos el comando `git commit -m "Primer commit"`

```
Gabriel FM@DESKTOP-2073SMD MINGW64 /c/Proyectos/Práctica Git (main)
$ git commit -m "Primer commit"
[main (root-commit) 420c496] Primer commit
1 file changed, 15 insertions(+)
create mode 100644 index.html

Gabriel FM@DESKTOP-2073SMD MINGW64 /c/Proyectos/Práctica Git (main)
$
```

**Paso 2.** Ejecutaremos el siguiente comando **git commit -m "Second commit"**

```
Gabriel FM@DESKTOP-2073SMD MINGW64 /c/Proyectos/Práctica Git (main)
$ git commit -m "Second commit"
[main 096065c] Second commit
1 file changed, 1 insertion(+), 1 deletion(-)
```



# Ahora obtendremos el historial de commits realizados

**Paso 1.** Aplicamos el comando **git log** y obtendremos el historial de commits realizados.

```
Gabriel FM@DESKTOP-2073SMD MINGW64 /c/Proyectos/Práctica Git (main)
$ git log
commit 096065c90b61635db45301e0a684f612425b00fa (HEAD -> main)
Author: Gabriel <gabrielfradesmoras@gmail.com>
Date:   Wed Apr 26 09:25:24 2023 +0200

    Second commit

commit 561a2fa7c2bbfa2703f5a527a58345ca17a8cd98
Author: Gabriel <gabrielfradesmoras@gmail.com>
Date:   Tue Apr 25 14:41:16 2023 +0200

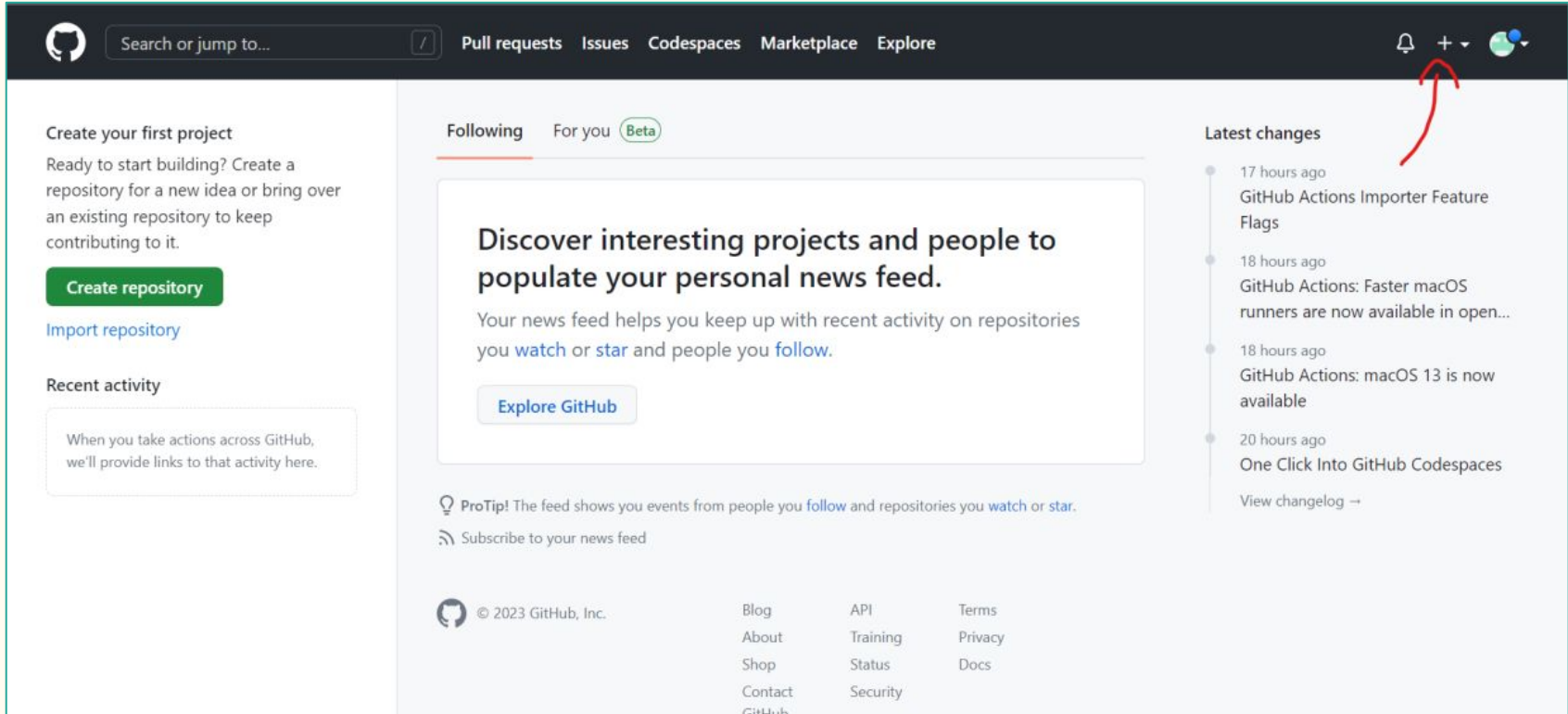
    Second commit

commit 420c496d28f241ab90b7d304eab5c192e76800b3 (origin/main)
Author: Gabriel <gabrielfradesmoras@gmail.com>
Date:   Tue Apr 25 12:05:43 2023 +0200

    Primer commit
```

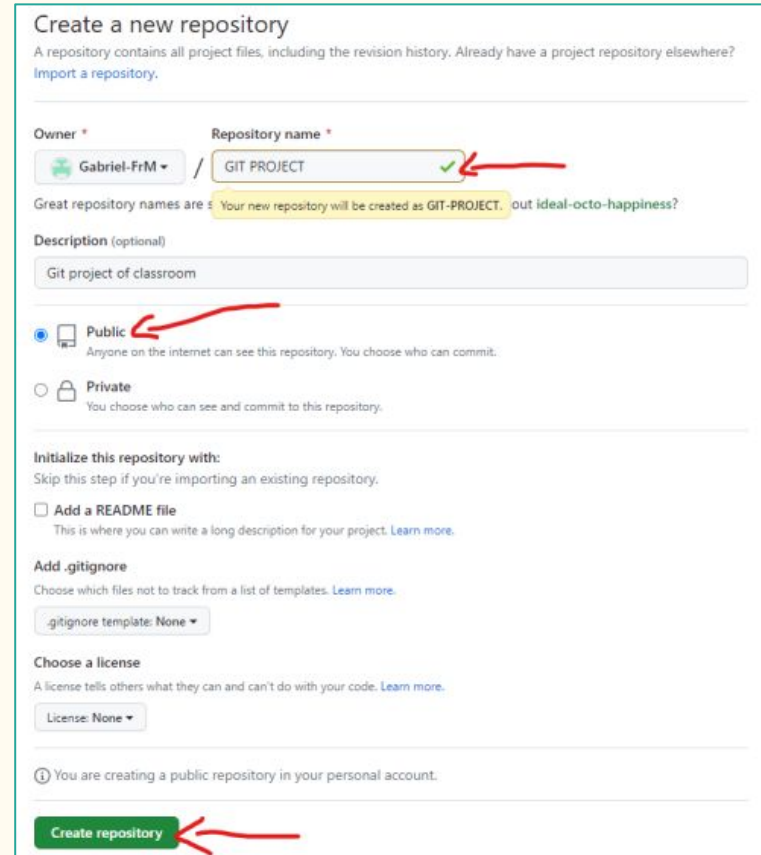
**SUBIR REPOSITORIO A GITHUB**

**Paso 1.** Entraremos a github y seleccionaremos el signo + en la parte superior.



The screenshot shows the GitHub homepage. At the top, there is a dark navigation bar. On the left, it features the GitHub logo and a search bar labeled "Search or jump to...". To the right of the search bar are links for "Pull requests", "Issues", "Codespaces", "Marketplace", and "Explore". On the far right of the navigation bar are icons for notifications (bell), a plus sign (+) with a dropdown arrow, and a user profile icon. A red arrow points to the plus sign icon. Below the navigation bar, the main content area is divided into three columns. The left column has a section "Create your first project" with a green "Create repository" button and a link "Import repository". Below that is "Recent activity" with a note about activity links. The middle column has a "Following" tab (active) and a "For you" tab with a "Beta" badge. It features a large card titled "Discover interesting projects and people to populate your personal news feed." with a button "Explore GitHub". Below the card is a "ProTip!" and a "Subscribe to your news feed" link. The right column is titled "Latest changes" and lists recent updates from GitHub Actions, such as "GitHub Actions Importer Feature Flags" and "GitHub Actions: Faster macOS runners". At the bottom, there is a footer with the GitHub logo, copyright information "© 2023 GitHub, Inc.", and a grid of links including "Blog", "About", "Shop", "Contact", "API", "Training", "Status", "Security", "Terms", "Privacy", and "Docs".

**Paso 2.** Una vez dentro pondremos un nombre al repositorio, lo dejaremos en público y crearemos el repositorio.



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

Owner \* Gabriel-FrM / Repository name \* GIT PROJECT ✓

Great repository names are [short, lowercase, and contain only hyphens](#). Your new repository will be created as **GIT-PROJECT**. Out ideal-octo-happiness?

Description (optional)  
Git project of classroom

☒ **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:**  
Skip this step if you're importing an existing repository.

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

**Add .gitignore**  
Choose which files not to track from a list of templates. [Learn more.](#)  
.gitignore template: None


**Choose a license**  
A license tells others what they can and can't do with your code. [Learn more.](#)  
License: None

ⓘ You are creating a public repository in your personal account.

**Create repository**

## Paso 3. Copiaremos el URI y pondremos la linea de comando en el propio GIT.

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

 Set up in Desktop or **HTTPS** **SSH** <https://github.com/Gabriel-FrM/GIT-PROJECT.git> 

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 "# GIT-PROJECT" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/Gabriel-FrM/GIT-PROJECT.git
git push -u origin main
```

...or push an existing repository from the command line

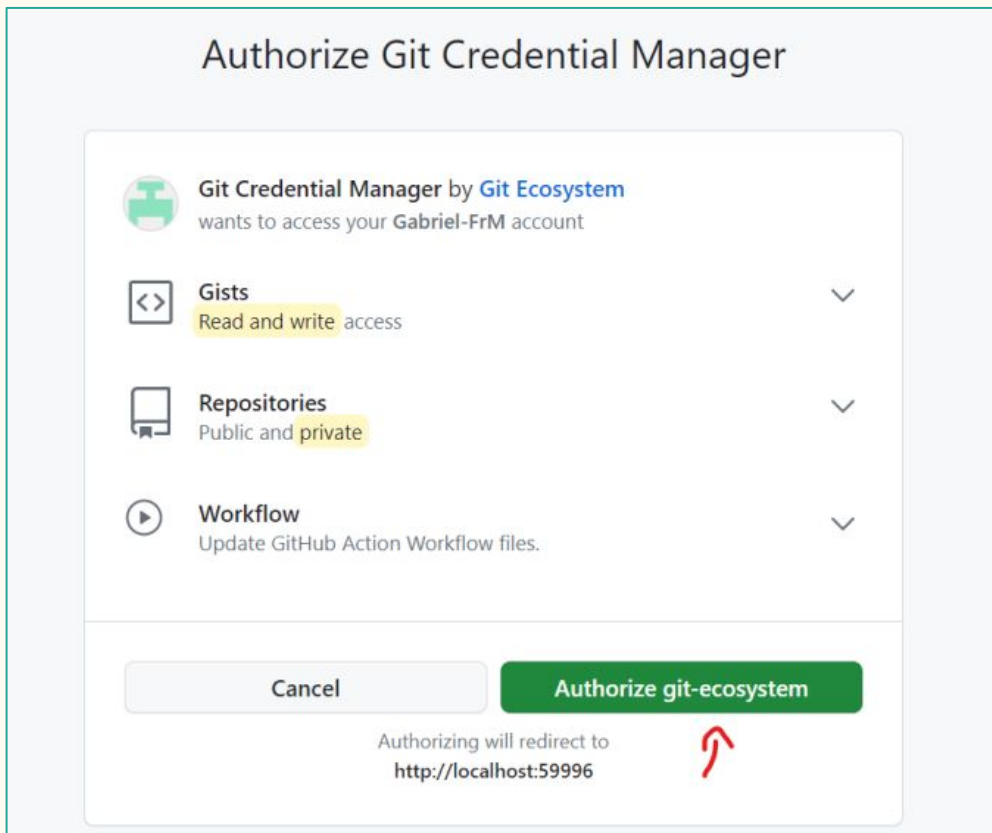
```
git remote add origin https://github.com/Gabriel-FrM/GIT-PROJECT.git
git branch -M main
git push -u origin main
```

```
Gabriel FM@DESKTOP-2073SMD MINGW64 /c/Proyectos/Práctica Git (main)
$ git remote add origin https://github.com/Gabriel-FrM/GIT-PROJECT.git
```

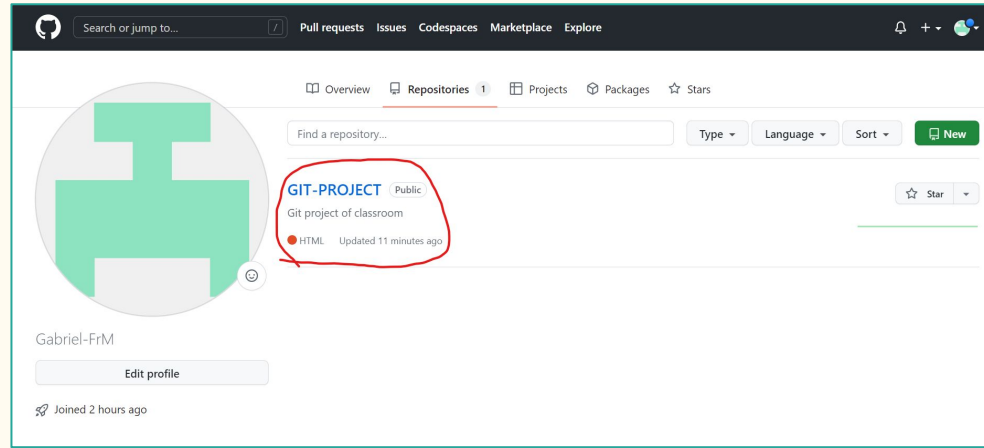
```
Gabriel FM@DESKTOP-2073SMD MINGW64 /c/Proyectos/Práctica Git (main)
$ git branch -M main
```

```
Gabriel FM@DESKTOP-2073SMD MINGW64 /c/Proyectos/Práctica Git (main)
$ git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 479 bytes | 479.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Gabriel-FrM/GIT-PROJECT.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

**Paso 4.** Te redireccionará a una página web donde deberás logearte y se subirá a tu perfil de **Github**.



**Paso 5.** Una vez en tu perfil podrás ver que el repositorio se ha subido correctamente.



**Gabriel-FrM** first commit

d72d7a8 1 minute ago 4 commits



index.html

first commit

1 minute ago