

Git



Introducción

Control de versiones que registra los cambios realizados en nuestros archivos para recuperar versiones específicas.

La historia de Git

Git comenzó un poco de la destrucción creativa y una gran polémica.

En el 2005 la comunidad encargada del desarrollo del Kernel Linux usaba la herramienta DVCS de Bitkeeper pero esta relación dejó de funcionar al dejar de estar disponible esta herramienta de forma gratuita. Esto impulsó a la comunidad y en particular a Linux Torvalds (video: Torvalds explica por qué hizo git), el creador de Linux a desarrollar su propia herramienta con los siguientes objetivos:

- Velocidad
- Diseño sencillo
- Gran Soporte para desarrollo no lineal (Ramas)
- Completamente distribuido
- Capaz de manejar grandes proyectos

Fundamentos de Git

La principal diferencia entre Git y cualquier otro VCS es la forma en la que manejan sus datos. Conceptualmente, la mayoría de los otros sistema almacenan la información como una lista de cambios en los archivos y las modificaciones hechas a cada una de ellos a través del tiempo.



Objetivos

- Crear proyecto de front en repositorio de git.
- Enviar repositorio en GitHub.
- Publicar sitio en GitHub Pages.

Antes de empezar

Instalaciones

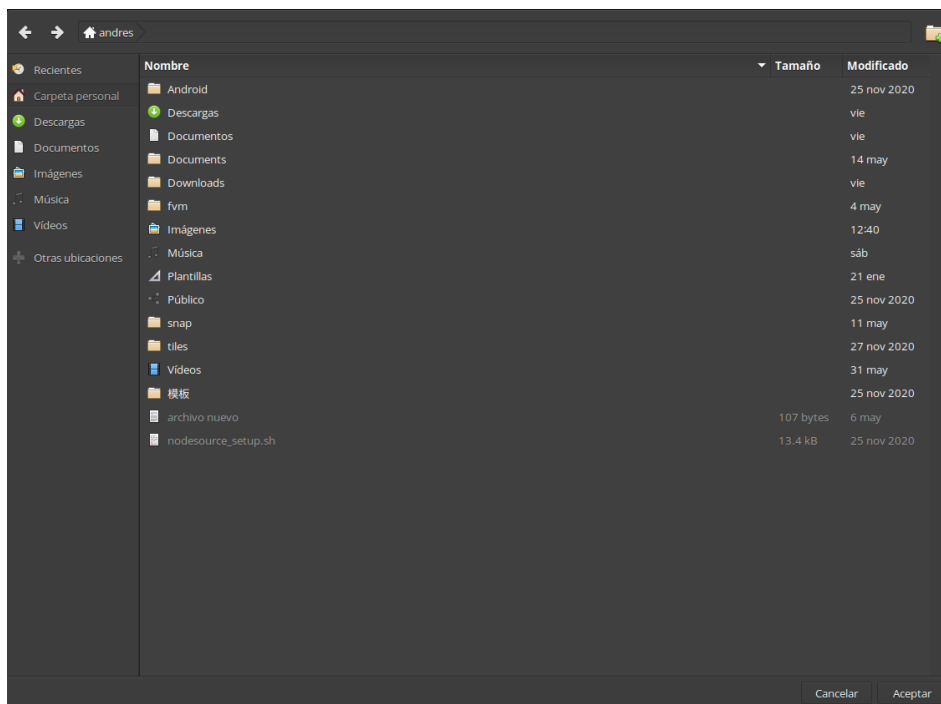
A continuación te mostramos las instalaciones necesarias para tu módulo:

- **Git**
 - Puedes descargar visual studio code del siguiente [link](#)
 - En el siguiente [link](#) pueden ver un video con los pasos de instalación
- **Visual Studio Code**
 - Puedes descargar visual studio code del siguiente [link](#)
 - En el siguiente [link](#) pueden ver un video con los pasos de instalación

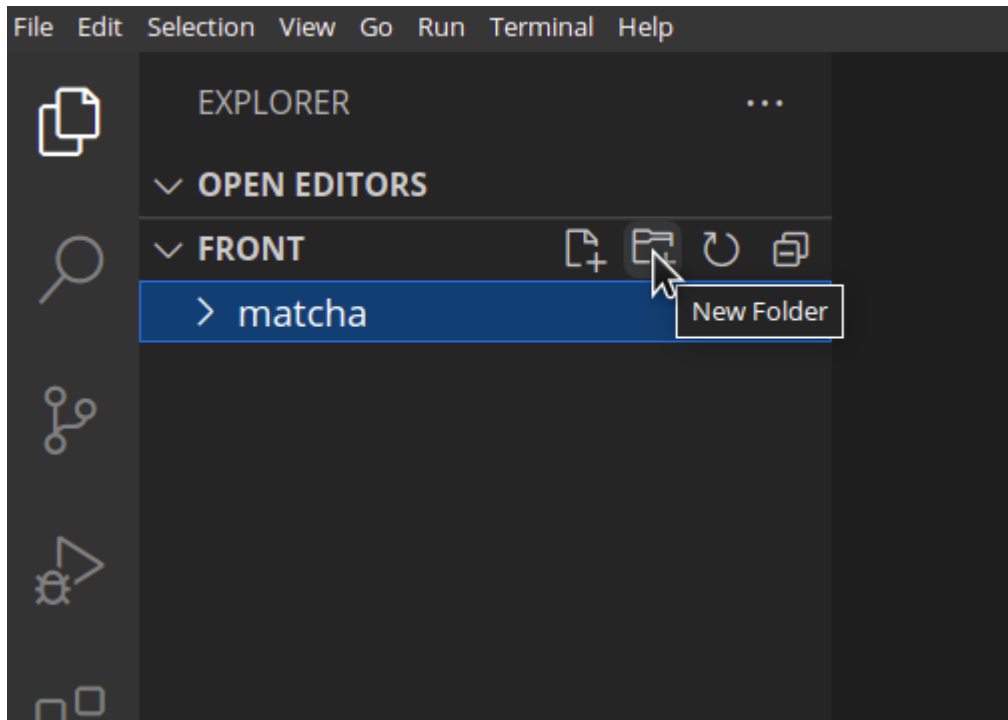
¡Comenzamos! 😊

Tema 01. Crear y visualizar proyecto en Visual Studio Code

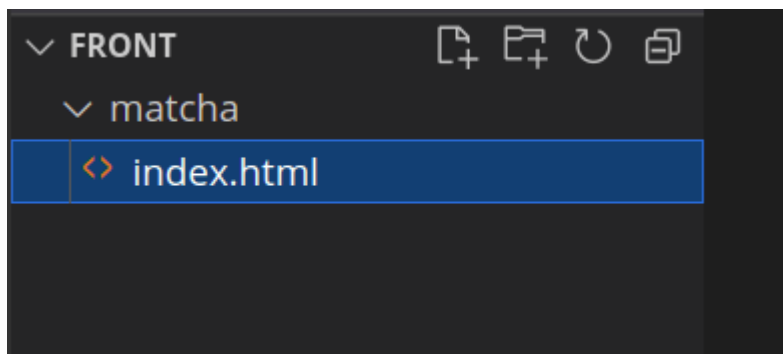
1. Abrimos visual y damos clic en File -> Open Folder, veremos algo similar a lo siguiente:



2. Creamos una nueva carpeta y le asignamos el nombre de FRONT
3. Ahora creamos una carpeta con el nombre de **matcha**



4. Dentro de la carpeta creamos el **index.html**



5. Dentro agregamos el siguiente html

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>Document</title>
  <link rel="stylesheet" href="./styles.css">
</head>
```

```
<body>
  <h1> Mi primer etiqueta </h1>
  <h2> Mi etiqueta </h2>
  <h3>Mi etiqueta</h3>
  <h4>Mi etiqueta</h4>
  <h5>Mi etiqueta</h5>
  <h6>Mi etiqueta</h6>

  <p>
    <b>Mi etiqueta</b> parrafo
  </p>

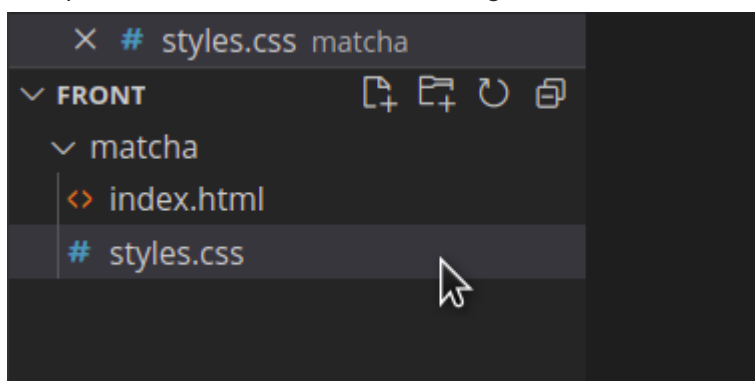
  <!-- <input type="text"><br> -->
  <form>
    <label for="name">Nombre</label>
    <input type="text" id="name" name="fname" required><br><br>
    <input type="text" id="lastname" name="lastname"
placeholder="Apellido"><br><br>
    <input type="number" id="age" name="age" placeholder="Edad"
min="18" max="35"><br><br>
    <input type="email" id="email" name="email"
placeholder="Correo"><br><br>
    <input type="password" id="password" name="password"
placeholder="Contraseña"><br><br>
    <button type="submit">Enviar</button>
  </form>

</body>

</html>
```

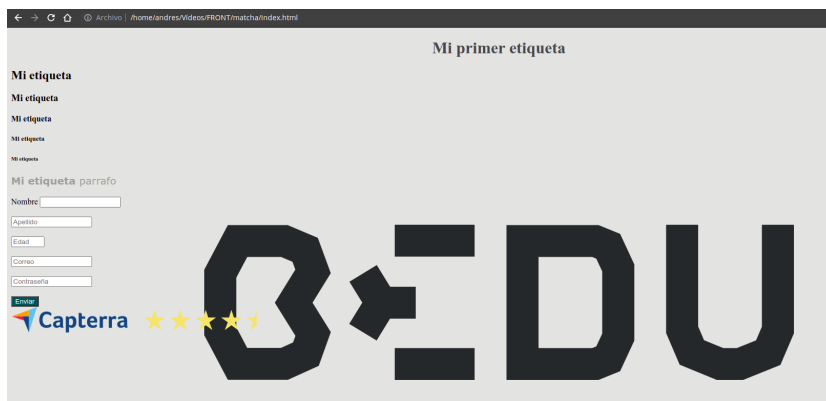
6. Después creamos el archivo **styles.css**



7. A continuación agregamos el siguiente style al archivo que acabamos de crear

```
body {  
    background-color: #f49e47;  
    background-image:  
url('https://pro2-bar-s3-cdn-cf.myportfolio.com/67cf45ca-338d-4fd0-  
943b-32d96162976a/b7b52d14-b827-4c3a-906f-3b94fa13f35f_rw_192  
0.png');  
}  
  
h1 {  
    color: #46484c;  
    text-align: center;  
}  
  
p {  
    color: #8b8b8bcc;  
    font-family: verdana;  
    font-size: 20px;  
}  
  
button {  
    color: #fff;  
    background-color: #025157;  
}
```

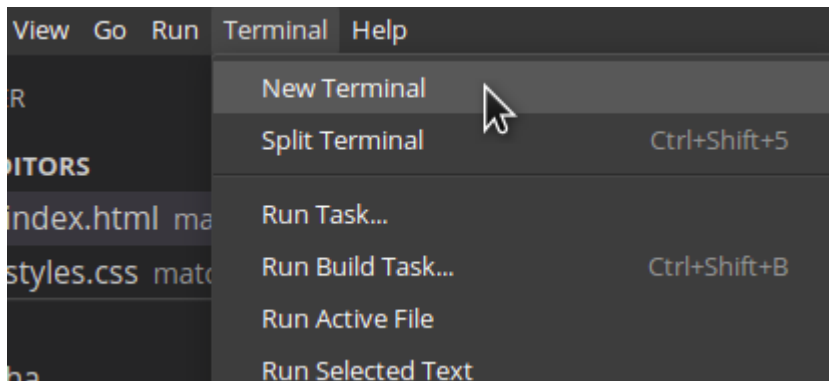
8. Es momento de abrir nuestro **html** en el **navegador**, hay varias opciones:
- Clic derecho en el nombre del archivo y clic en **copy path**, nos dirigimos al navegador y pegamos la ruta.
 - Buscar el archivo en el administrador de archivos “Explorer, Finder...” doble clic en el index.html o clic derecho abrir con y seleccionamos el navegador que queramos.
9. Listo ya podemos ver nuestro html en el navegador



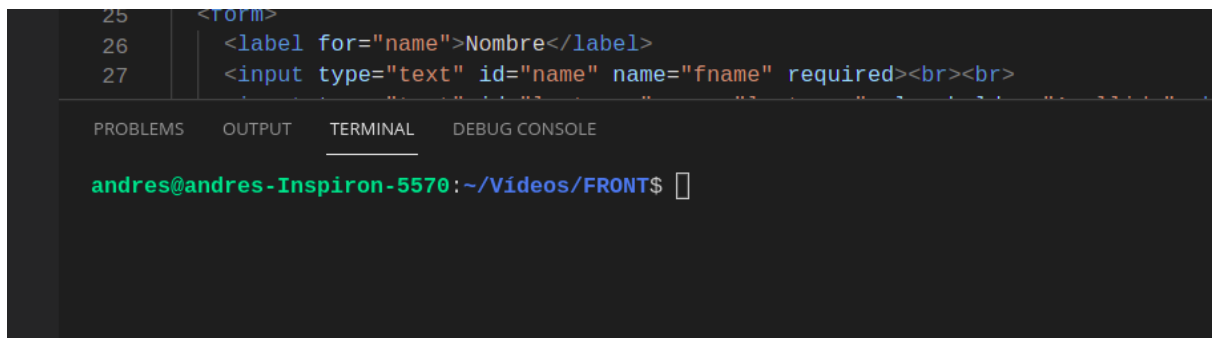
Tema 02. Crear repositorio

Ya tenemos nuestro proyecto de front, ahora vamos crear un repositorio, para ello seguimos los siguiente pasos:

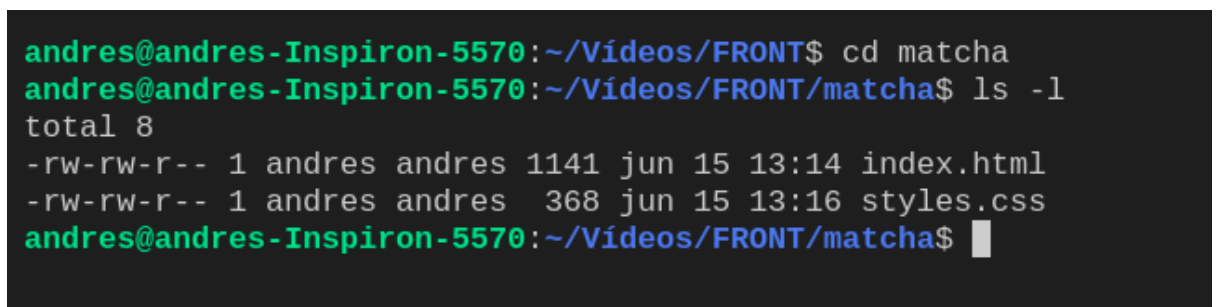
1. Abrimos la terminal de visual studio code, para ello damos clic en Terminal -> New terminal



2. Esto abre la terminal en la parte inferior de Code



3. Tenemos que entrar a la carpeta de **matcha**, ejecutamos el siguiente comando **cd matcha**
4. Ahora ejecutamos **ls -l** deberíamos de ver algo similar a lo siguiente



5. Después agregamos el siguiente comando para iniciar el proyecto de git **git init** nos mostrará un mensaje similar a **Inicializado repositorio Git vacío en /home/andres/Videos/FRONT/matcha/.git/**
6. A continuación preparamos los archivos que serán enviados al commit con **git add**.

7. Ahora agregamos el commit con su respectivo mensaje **git commit -m "Mi primer commit"**

```
andres@andres-Inspiron-5570:~/Videos/FRONT/matcha$ git commit -m "Mi primer commit"
[master (commit-raíz) 348c7d1] Mi primer commit
2 files changed, 60 insertions(+)
create mode 100644 index.html
create mode 100644 styles.css
andres@andres-Inspiron-5570:~/Videos/FRONT/matcha$
```

Si el paso anterior les muestra un error similar al siguiente, necesitan hacer unos pasos extra:

```
*** Please tell me who you are

Run

git config --global user.email 'you@email.com'
git config --global user.name 'Your name'

...
```

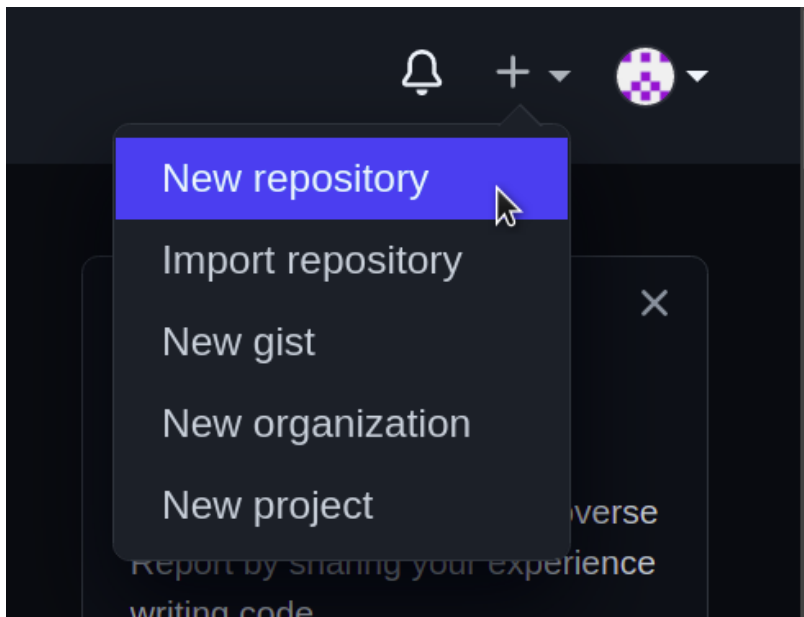
Lo primero que debemos realizar es configurar la identidad, debemos establecer el nombre de usuario y dirección de correo electrónico.

Esto es importante porque cada que realicemos un “commit” (Registro de cambio) git usa esta información para registrar quien realizó este cambio.

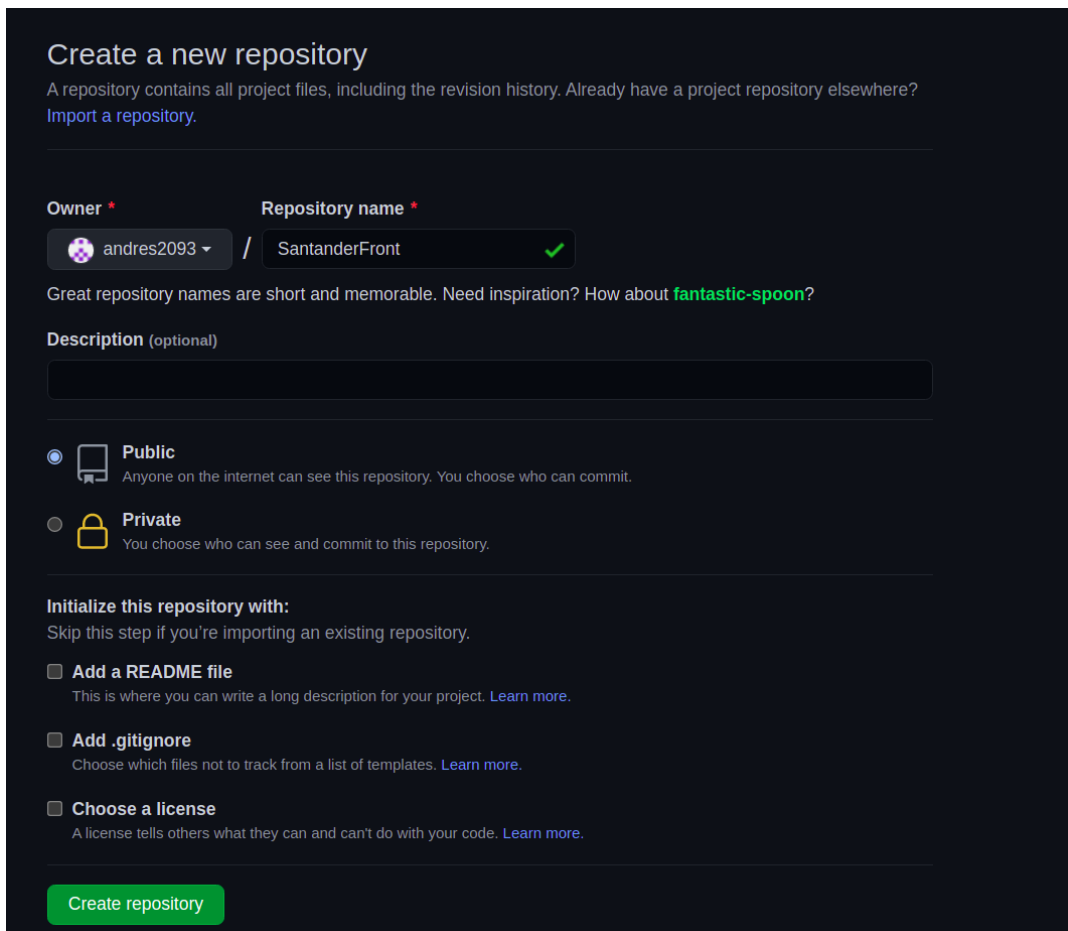
1. Para realizar esta acción debemos escribir la siguiente línea de código para establecer nuestro nombre de usuario:
 - a. `git config --global user.name "Nombre de Usuario"`
 2. A continuación para establecer el correo electrónico que usaremos ingresamos la siguiente línea de código:
 - a. `git config --global user.email "micorreo@example.com"`
 3. Ahora podremos realizar el commit
8. Listo ya creamos el repositorio

Tema 03. Enviar cambios a GitHub

1. Nos dirigimos a GitHub y creamos un nuevo repositorio



2. Agregamos el siguiente nombre **SantanderFront** y damos clic en **Create repository**


 A screenshot of the 'Create a new repository' form on GitHub. The form has a title 'Create a new repository' and a subtitle 'A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)'. Below this, there are two input fields: 'Owner' with a dropdown menu showing 'andres2093' and 'Repository name' with a text input showing 'SantanderFront' and a green checkmark. A message below says 'Great repository names are short and memorable. Need inspiration? How about [fantastic-spoon?](#)'. There is a 'Description (optional)' text area. Below that, there are two radio button options: 'Public' (selected) and 'Private'. The 'Public' option has a description: 'Anyone on the internet can see this repository. You choose who can commit.' The 'Private' option has a description: 'You choose who can see and commit to this repository.' Below these, there is a section 'Initialize this repository with:' with a subtitle 'Skip this step if you're importing an existing repository.' and three checkboxes: 'Add a README file' (with a description 'This is where you can write a long description for your project. [Learn more.](#)'), 'Add .gitignore' (with a description 'Choose which files not to track from a list of templates. [Learn more.](#)'), and 'Choose a license' (with a description 'A license tells others what they can and can't do with your code. [Learn more.](#)'). At the bottom, there is a green 'Create repository' button.

- Nos muestra la siguiente pantalla, si se dan cuenta ya realizamos algunos de los pasos, ahora vamos a agregar el remoto con el comando que nos muestra en el paso 6 “El comando lo ejecutamos en Visual Studio Code”



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

or HTTPS SSH <https://github.com/andres2093/SantanderFront.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

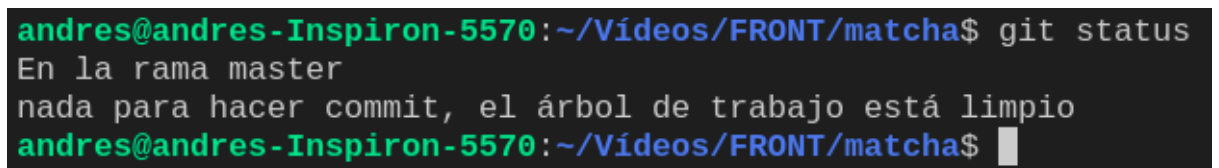
```
echo "# SantanderFront" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/andres2093/SantanderFront.git
git push -u origin main
```

```
/FRONT/matcha$ git remote add origin https://github.com/andres2093/SantanderFront.git
/FRONT/matcha$
```

- En el siguiente paso varios tuvieron problemas, tenemos dos alternativas para enviar la info a GitHub, lo vamos a separar en dos temas (4 y 5)

Tema 04. Primer método de subir a GitHub

- Ejecutamos la siguiente línea para enviar los cambios en tu rama, si seguiste los pasos anteriores tu rama debería de llamarse **master**, para comprobar que tiene ese nombre podemos ejecutar **git status** este nos muestra el branch que tenemos seleccionado

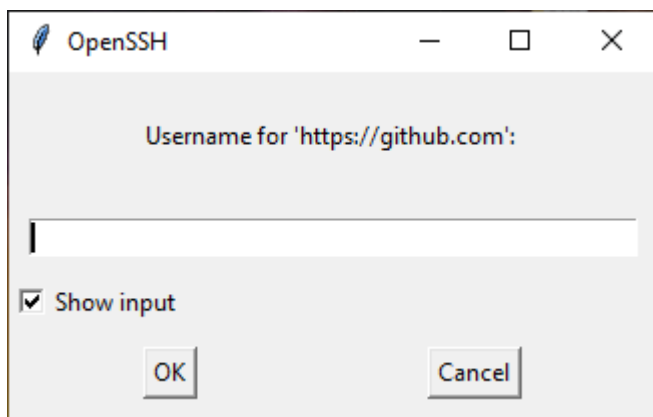


```
andres@andres-Inspiron-5570:~/Videos/FRONT/matcha$ git status
En la rama master
nada para hacer commit, el árbol de trabajo está limpio
andres@andres-Inspiron-5570:~/Videos/FRONT/matcha$
```

- Ahora que sabemos el branch en el que estamos, ejecutamos el siguiente comando **git push -u origin master** si tu branch se llama diferente que master cambialo por el tuyo, por ejemplo **main**

Lo siguiente cambia dependiendo nuestro sistema operativo y las versiones que tenga instalado, pero nos debe preguntar las credenciales de nuestra cuenta de GitHub, a continuación se muestran algunos ejemplos de interfaces que puede mostrar, en todos los casos debemos agregar nuestro correo o nombre de usuario y la contraseña

La primera vez tarda, solo hay que tener un poco de paciencia.

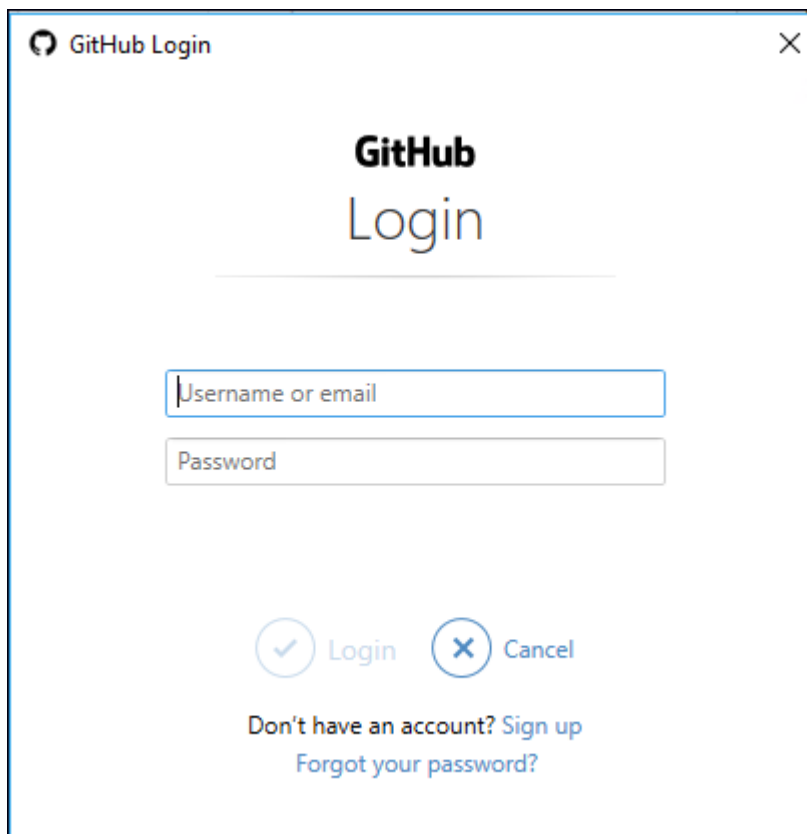


OpenSSH

Username for 'https://github.com':

☒ Show input

OK Cancel



GitHub Login

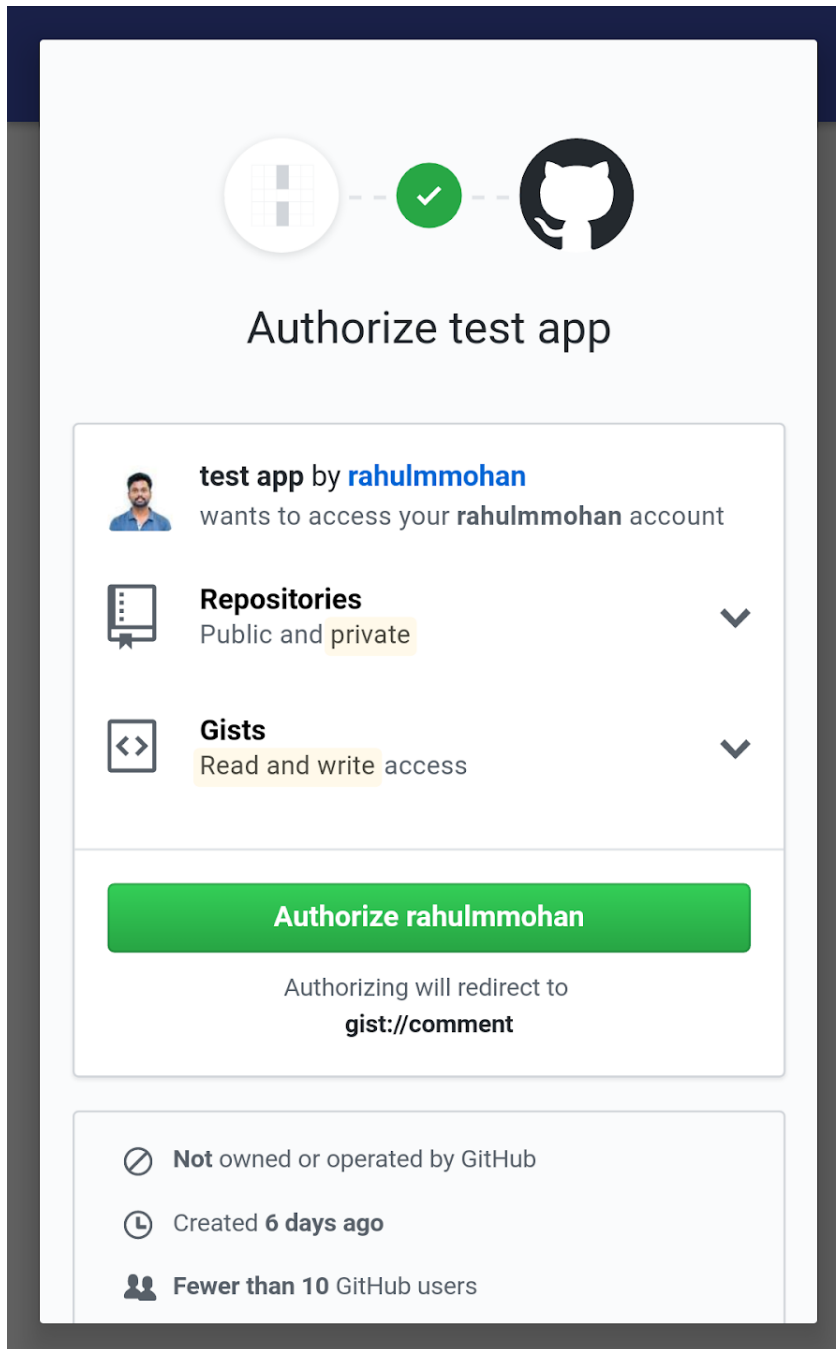
GitHub
Login

Username or email

Password

☒ Login ☐ Cancel

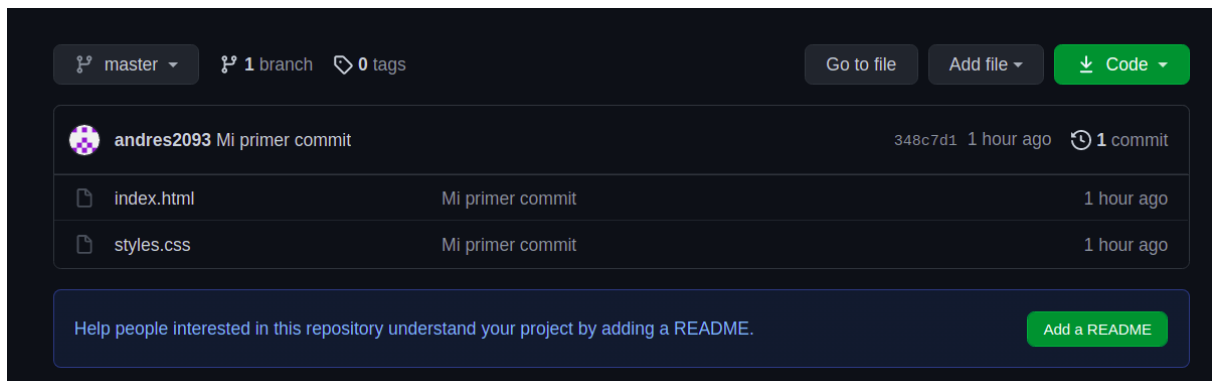
Don't have an account? [Sign up](#)
[Forgot your password?](#)



3. Si los cambios se enviaron correctamente veríamos la siguiente pantalla

```
andres@andres-Inspiron-5570:~/Videos/FRONT/matcha$ git push -u origin master
Enumerando objetos: 4, listo.
Contando objetos: 100% (4/4), listo.
Compresión delta usando hasta 8 hilos
Comprimiendo objetos: 100% (4/4), listo.
Escribiendo objetos: 100% (4/4), 1.02 KiB | 1.02 MiB/s, listo.
Total 4 (delta 0), reusado 0 (delta 0), pack-reusado 0
To https://github.com/andres2093/SantanderFront.git
 * [new branch]      master -> master
Rama 'master' configurada para hacer seguimiento a la rama remota 'master' de 'origin'.
andres@andres-Inspiron-5570:~/Videos/FRONT/matcha$
```

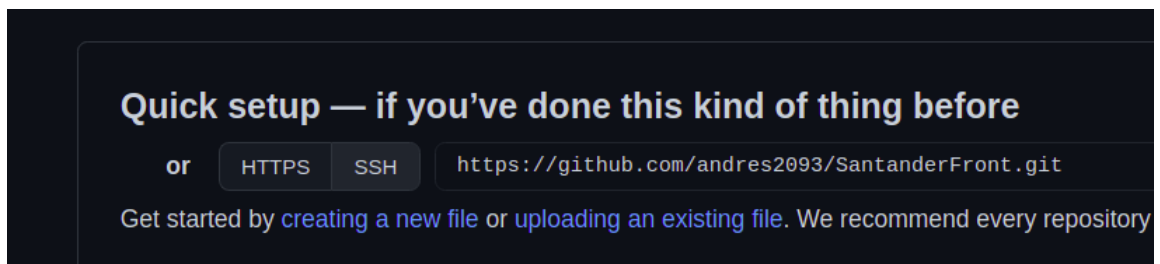
4. Nos dirigimos a GitHub y actualizamos la pagina, deberíamos de ver nuestros cambios



Tema 05. Segundo método de subir a GitHub

Si el método anterior falló, probamos con la siguiente alternativa:

1. Eliminamos el remoto https con el siguiente comando **git remote remove origin**
2. En la página de git debemos buscar la conexión con ssh

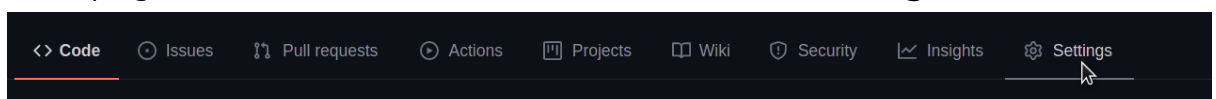


Seleccionamos la opción **ssh** el remoto debe tener la siguiente estructura **git@github.com:andres2093/SantanderFront.git**

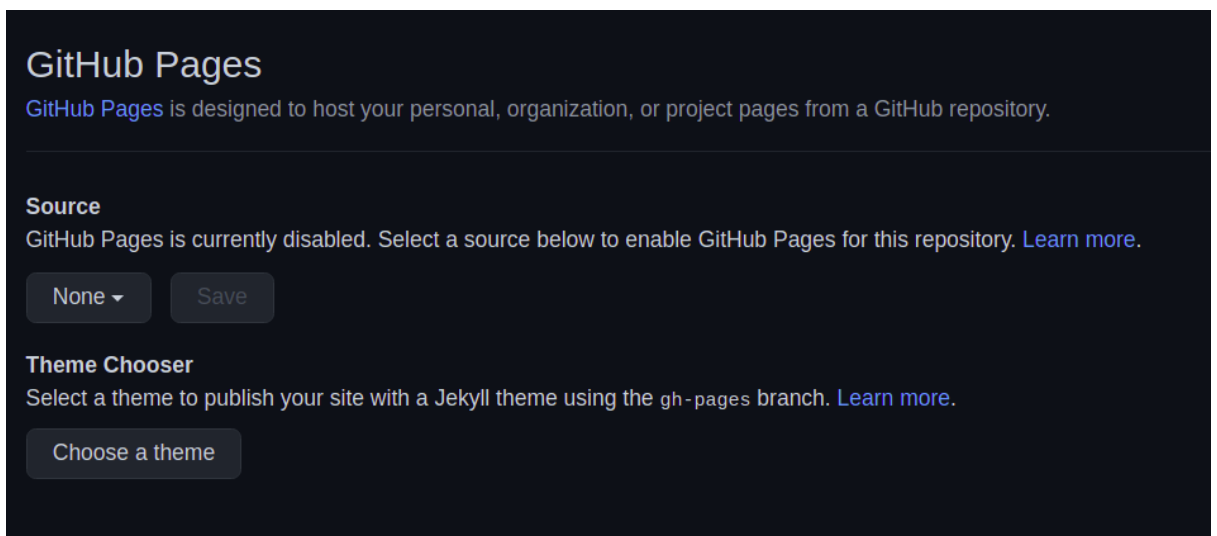
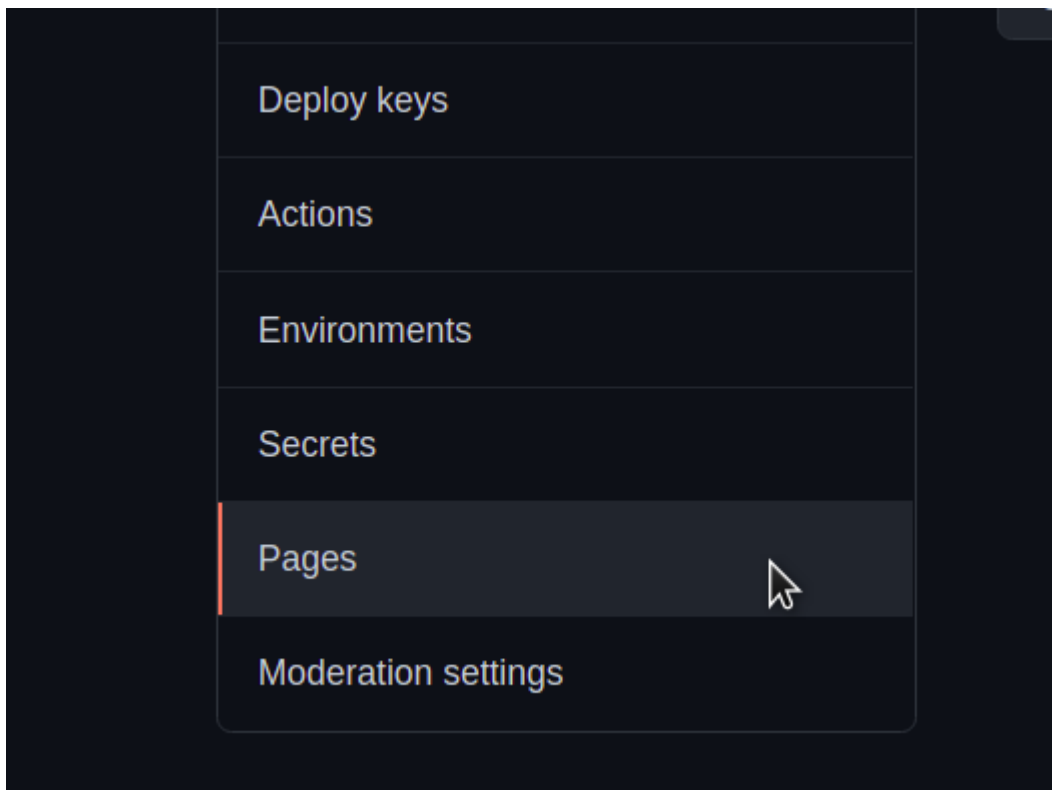
3. Tomamos el remoto y agregamos los comandos para indicar que agregaremos el remoto, quedando de la siguiente manera:
git remote add origin
git@github.com:andres2093/SantanderFront.git
4. Los siguientes pasos los realizaremos con ayuda de la documentación de git, donde debemos crear la llave ssh y vincularla con GitHub. Entramos al siguiente [link](#)

Tema 06. Publicar repositorio en GitHub Pages

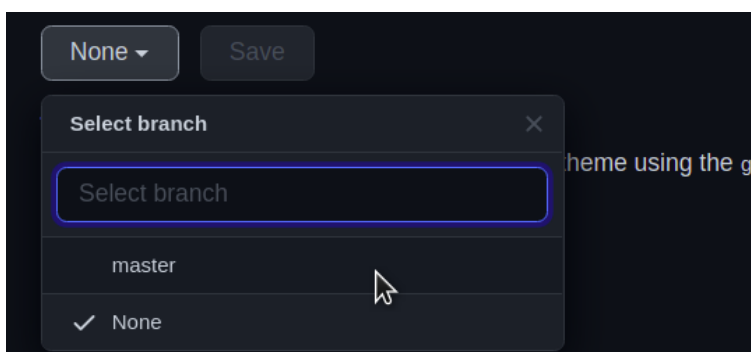
1. En la página de GitHub damos clic en el botón de **Settings**



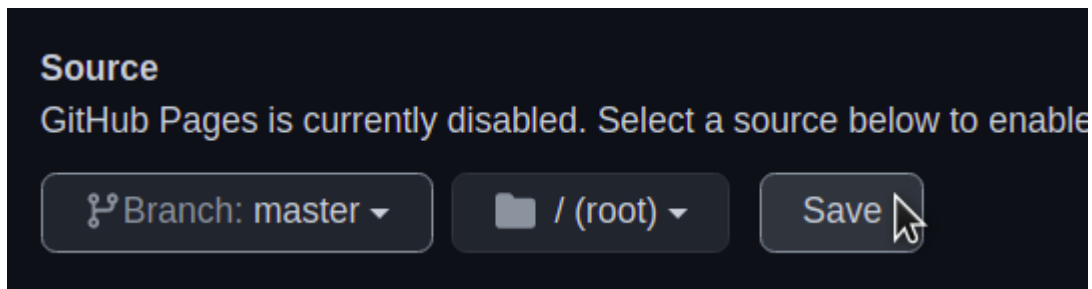
2. Después en **Pages**



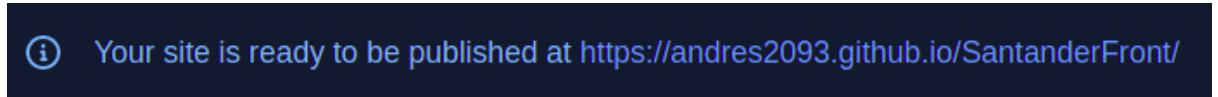
3. Luego damos clic en **None** y seleccionamos nuestro branch, en mi caso es master



4. A continuación damos clic en **Save**



5. Esperamos unos minutos a que se realice la publicación



6. Después de unos minutos, damos clic en el link y veríamos nuestra página publicada en Pages

