¿Qué es GitHub?

GitHub es una plataforma basada en la nube donde puedes almacenar, compartir y trabajar junto con otros usuarios para escribir código. [1]

Almacenar tu código en un "repositorio" en GitHub te permite lo siguiente:

- **Presentar o compartir** el trabajo.
- **Seguir y administrar** los cambios en el código a lo largo del tiempo.
- Dejar que otros usuarios **revisen** el código y realicen sugerencias para mejorarlo.
- **Colaborar** en un proyecto compartido, sin preocuparse de que los cambios afectarán al trabajo de los colaboradores antes de que esté listo para integrarlos.

El trabajo colaborativo, una de las características fundamentales de GitHub, es posible gracias al software de código abierto Git, en el que se basa GitHub.

¿Cómo crear un repositorio en GitHub?

Paso 1: Crear una cuenta de GitHub

Ir a: https://github.com/ crear una cuenta si no tienes una, en caso contrario iniciar sesión.

Paso 2: Crear nuevo repositorio

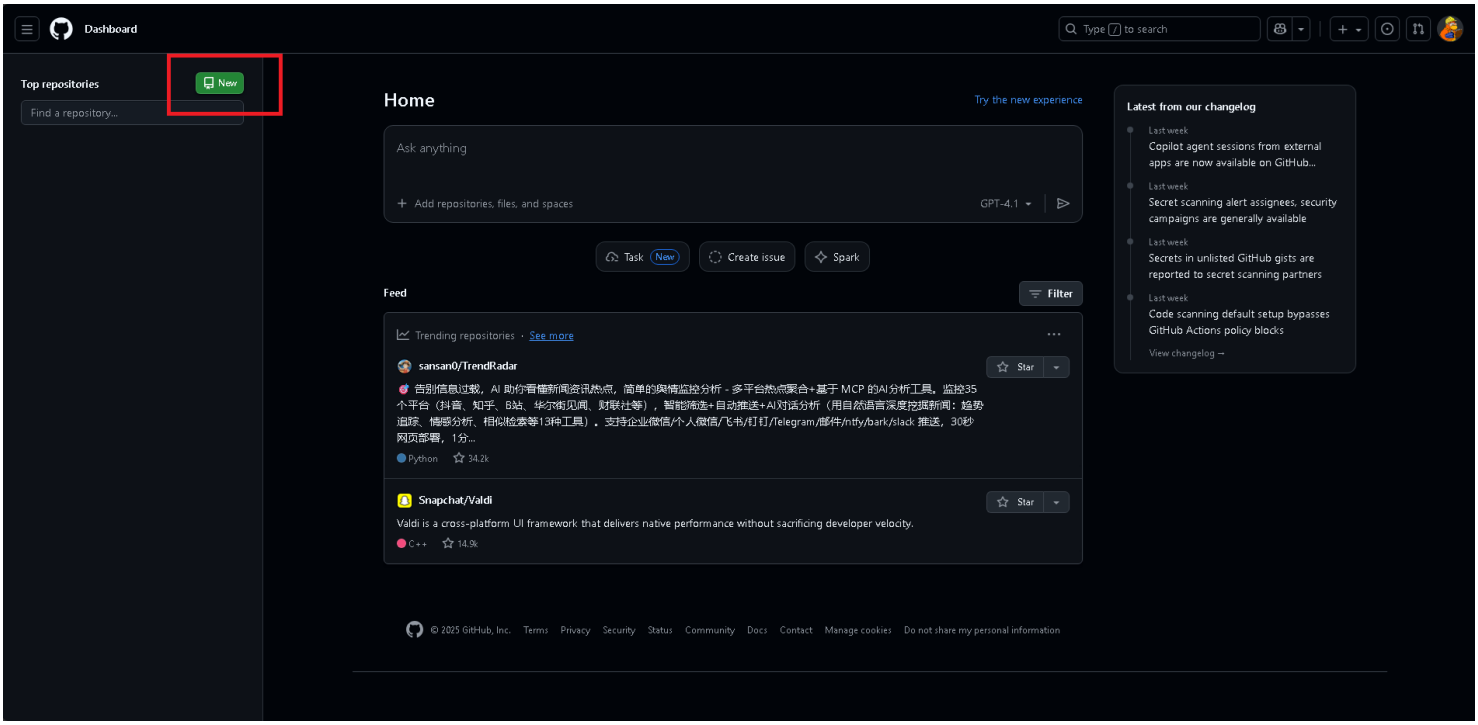


Figura 1: página principal sesión de GitHub.

Tras iniciar sesión se deben encontrar en una pagina similar a la de la figura 1, en esta pagina vamos a clickear en el botón verde resaltado por el recuadro rojo.

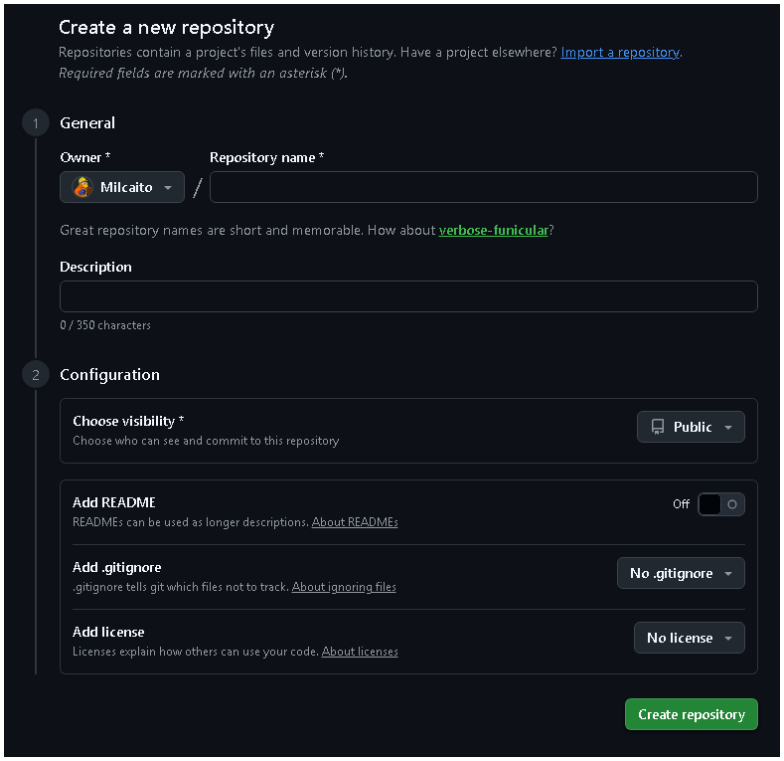


Figura 2: creación de repositorio.

Al darle click nos debería enviar a una pagina a rellenar para la creación del repositorio, similar a la Figura 2. Aquí debemos completar los siguientes campos:

- **Repository name:** Nombre claro y corto, idealmente sin espacios. Ej.: simulación_fenomenos_transporte.
- **Description (opcional):** Breve descripción del proyecto (1–2 líneas).
- **Choose visibility:**
    - **Public:** Recomendado para uso académico, ya que permite que docentes, ayudantes y compañeros revisen el avance del proyecto.
    - **Private:** Útil cuando el trabajo contiene información sensible o se requiere restringir el acceso.
- **Add README:** Si se selecciona, GitHub creará automáticamente un archivo README.md, que servirá como carta de presentación del proyecto. Es recomendable activarlo, ya que facilita la documentación inicial.
- **Add .**gitignore: Esta opción permite seleccionar una plantilla para ignorar archivos que no deben ser versionados (por ejemplo: archivos temporales, cachés, salidas de simulaciones, archivos pesados, etc.). GitHub ofrece plantillas predefinidas (por ejemplo: Python, Visual Studio).
- **Add license:** Permite asignar una licencia al proyecto. Esto define legalmente cómo otros usuarios pueden usar, modificar o distribuir tu trabajo. Por simplicidad pueden dejarlo como *No license*.

Una vez elegida la configuración, pueden darle al botón de Create repository, ya tienen creado su repositorio.

Paso 3: hacer cambios directos repositorio remoto.

Para subir archivos hay diversas formas, la más sencilla es arrastrando los archivos directamente y se suben al repositorio. Esto se puede hacer clickeando en *Upload files* en el repositorio remoto.

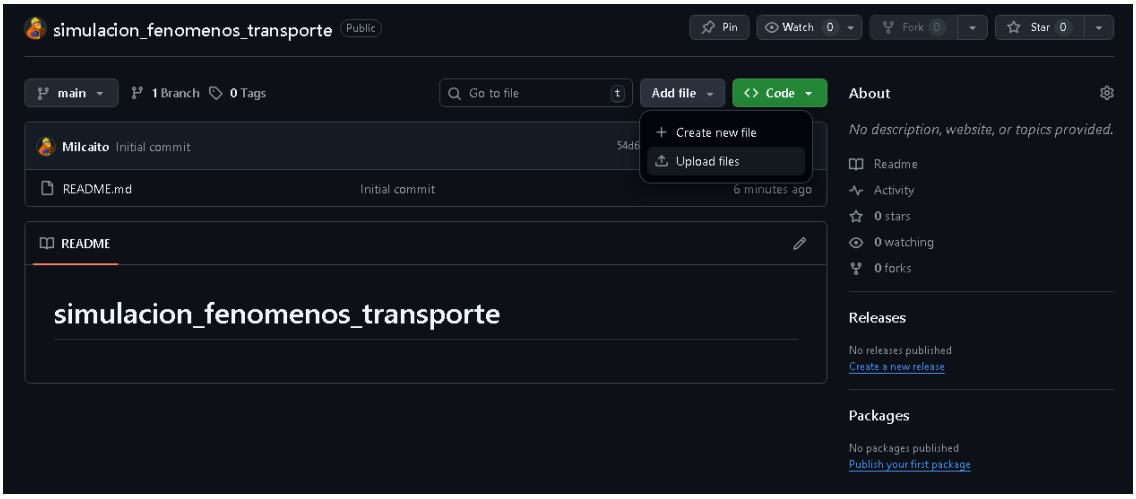


Figura 3: repositorio remoto.

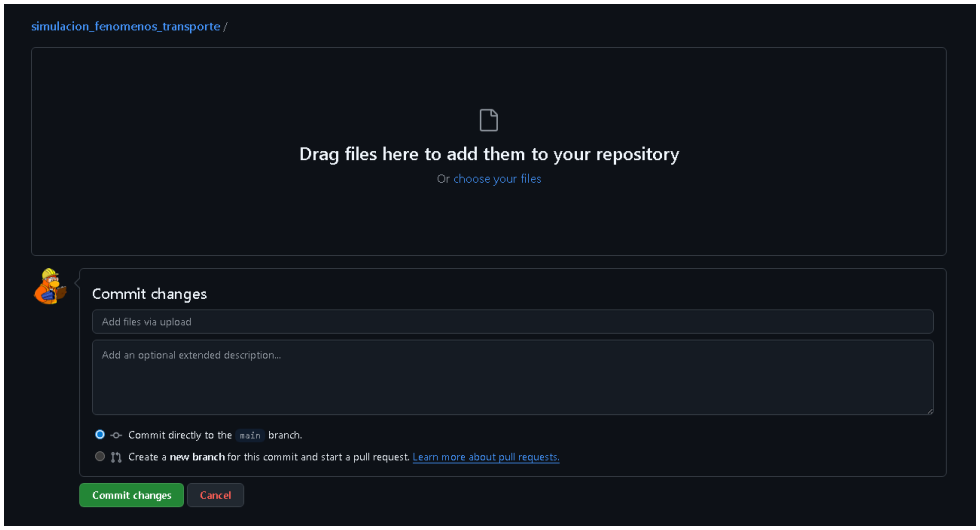


Figura 4: Pestaña subida de archivos método directo.

Como se observa en la figura 4, se pueden arrastrar los archivos de manera directa para subirlos a la carpeta principal de nuestro repositorio.

También es importante saber como eliminar archivos en nuestro repositorio, clickeamos en el archivo que queremos editar/eliminar en la sección resaltada en la figura 5. Luego dando click sobre los 3 puntos resaltados en la figura 6, se despliega una ventana de opciones, aquí podemos seleccionar eliminar el archivo que estemos inspeccionando.

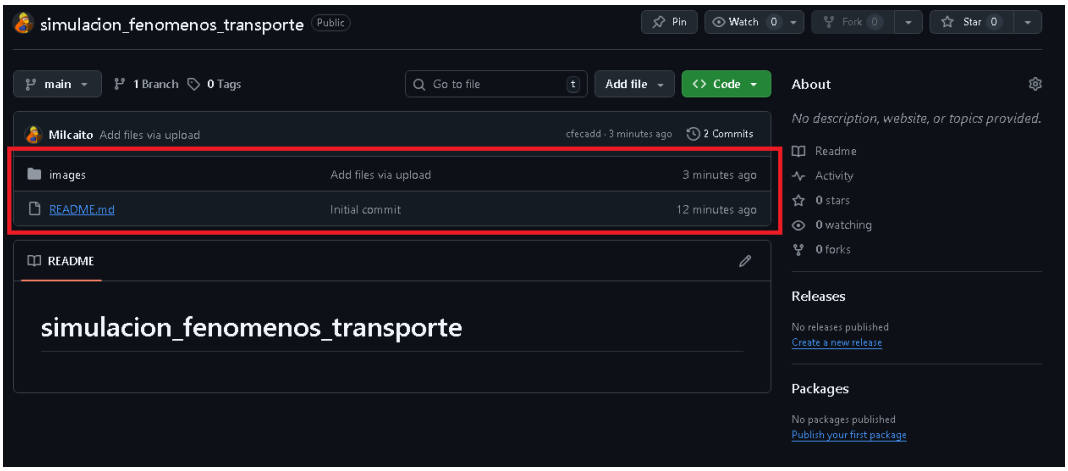


Figura 5: sección de archivos repositorio.

Figura 6: eliminación de archivos repositorio.

Paso 4 (opcional): Clonar repositorio localmente y sincronizar.

Primero debemos de instalar Git, esto lo podemos hacer directamente desde la consola de comando de Windows con el siguiente comando: “winget install Git.Git” y verificamos la instalación con “git –version”.

```
C:\Users\sotoj\Desktop\Proyecto Semetral IIQ2003\Material de apoyo\Repositorio_GitHub>winget install Git.Git
El origen 'msstore' requiere que vea los siguientes contratos antes de usarlo.
Terms of Transaction: https://aka.ms/microsoft-store-terms-of-transaction
El origen requiere que la región geográfica de dos letras de la máquina actual se envíe al servicio back-end para que funcione correctamente (por ejemplo, "EE. UU.").

¿Está de acuerdo con todos los términos de los contratos de origen?
[Y] Sí  [N] No: git --version
[Y] Sí  [N] No: Y
Encontrado Git [Git.Git] Versión 2.52.0
El propietario de esta aplicación le concede una licencia.
Microsoft no es responsable, ni tampoco concede ninguna licencia de paquetes de terceros.
Descargando https://github.com/git-for-windows/git/releases/download/v2.52.0.windows.1/Git-2.52.0-64-bit.exe
  ████████████████████████████████  63.2 MB / 63.2 MB
El hash del instalador se verificó correctamente
Iniciando instalación de paquete...
El instalador solicitará que se ejecute como administrador. Espere una indicación.
Instalado correctamente

C:\Users\sotoj\Desktop\Proyecto Semetral IIQ2003\Material de apoyo\Repositorio_GitHub>git --version
git version 2.52.0.windows.1
```

Figura 7: instalación git.

El siguiente paso debemos abrir una consola de comando de Windows en la carpeta donde queremos ubicar nuestro repositorio, esto lo podemos hacer yendo a nuestra carpeta y escribiendo “cmd” donde se ubica el PATH, ver la figura 8 como referencia.

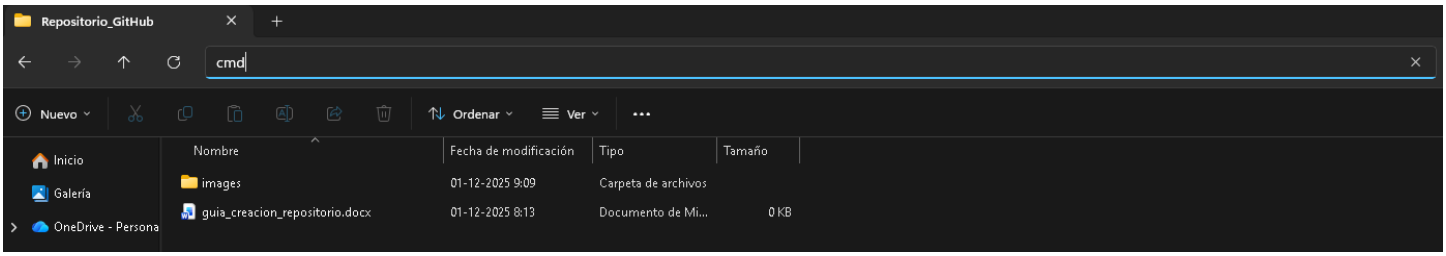


Figura 8: iniciar consola de comandos.

Una vez tengamos la consola de comando abierta ingresaremos los siguientes comandos, es importante hacer la copia por HTTPS:

- git clone https://github.com/usuario/nombre-repositorio.git, el enlace lo podemos copiar en la pagina principal del repositorio haciendo click en *code*, y luego en *HTTPS*, como se observa en laFigura 8.
- cd nombre-repositorio
- git status

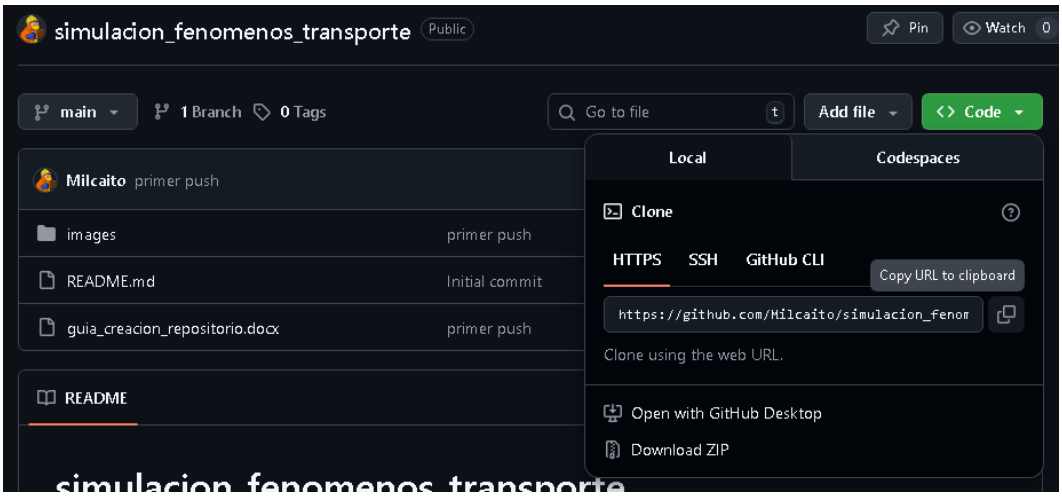


Figura 9: selección clonar repositorio HTTPS.

en este punto ya tenemos listo una copia de nuestro repositorio remoto en nuestro computador, el siguiente paso consiste en iniciar sesión localmente para poder realizar los cambios mediante Git directamente a GitHub. Para esto clickeamos en nuestra foto de perfil y vamos a configuración como se observa en la Figura 10.

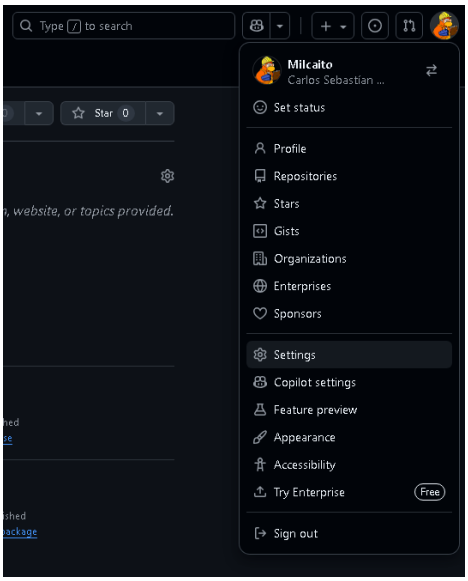
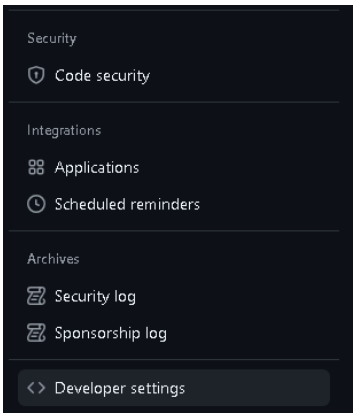
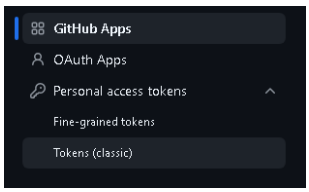


Figura 10: como ir a configuración.

Ya en la configuración de nuestro perfil, vamos a la ultima opción *Developer settings*, luego escogemos Personal Access tokens / tokens (classic), y aquí generamos nuestro token. IMPORTANTE deben rellenar la casilla de repo, para así poder hacer cambios en su repositorio. Una vez obtenido el token copienlo.





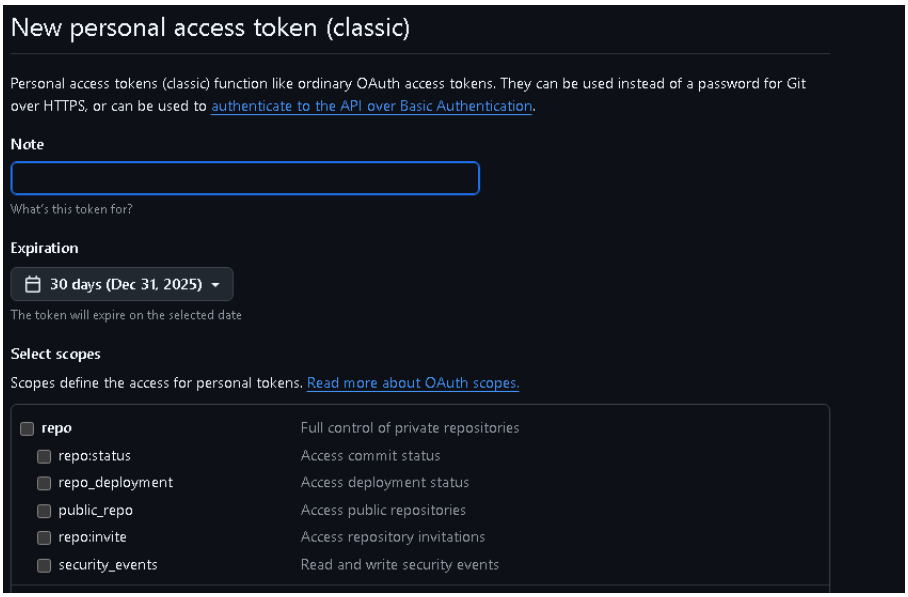


Figura 11: creación de token.

Para configurar su Git localmente deben ingresar los siguientes comandos en la consola:

- git --version
- git config --global user.name "Tu Nombre"
- git config --global user.email tu-correo@ejemplo.cl
- git push

una vez que hacen git push se debe abrir una ventana emergente aquí puede ingresar el token creado. Ya tienen configurado y sincronizado su repositorio local y remoto. Para el control de versiones en su repositorio local puede utilizar los siguientes comandos cada vez que quieran subir su avance al repositorio remoto:

- git add . (añade todos los archivos, sin considerar el .gitignore) o en su defecto pueden ocupar git add nombre-archivo.extension-archivo
- git commit -m “mensaje breve de su commit” (cada commit es una nueva versión de nuestro código que queda guardada en el repositorio remoto)
- git push (Sube el commit al repositorio remoto)
- git pull (descarga los archivos del repositorio remoto y actualiza los archivos del repositorio local para estar al día con el remoto.)

Referencias

[1] GitHub. (2025). *Acerca de GitHub y Git*. GitHub Docs. https://docs.github.com/es/get-started/start-your-journey/about-github-and-git