



**CICLO: DESARROLLO DE APLICACIONES WEB**  
**MÓDULO DE DAW**

# **Tarea N°6**

**Alumno:**  
**Sonia Enjuto Gil**  
**51489452L**

*Los documentos, elementos gráficos, vídeos, transparencias y otros recursos didácticos incluidos en este contenido pueden contener imprecisiones técnicas o errores tipográficos. Periódicamente se realizan cambios en el contenido. Fomento Ocupacional FOC SL puede realizar en cualquier momento, sin previo aviso, mejoras y/o cambios en el contenido.*

*Es responsabilidad del usuario el cumplimiento de todas las leyes de derechos de autor aplicables. Ningún elemento de este contenido (documentos, elementos gráficos, vídeos, transparencias y otros recursos didácticos asociados), ni parte de este contenido puede ser reproducida, almacenada o introducida en un sistema de recuperación, ni transmitida de ninguna forma ni por ningún medio (ya sea electrónico, mecánico, por fotocopia, grabación o de otra manera), ni con ningún propósito, sin la previa autorización por escrito de Fomento Ocupacional FOC SL.*

*Este contenido está protegido por la ley de propiedad intelectual e industrial. Pertenecen a Fomento Ocupacional FOC SL los derechos de autor y los demás derechos de propiedad intelectual e industrial sobre este contenido.*

*Sin perjuicio de los casos en que la ley aplicable prohíbe la exclusión de la responsabilidad por daños, Fomento Ocupacional FOC SL no se responsabiliza en ningún caso de daños indirectos, sean cuales fueren su naturaleza u origen, que se deriven o de otro modo estén relacionados con el uso de este contenido.*

## Contenido

1. Documentos que se adjuntan a este informe.	2
2. CONTEXTO DE LA TAREA	2
3. ENUNCIADO DE LA TAREA	2
4. (RA6_d) Se han utilizado herramientas colaborativas para la elaboración y mantenimiento de la documentación.	2
5. (RA6_e) Se ha instalado, configurado y utilizado un sistema de control de versiones.	7
6. (RA6_f) Se ha garantizado la accesibilidad y seguridad de la documentación almacenada por el sistema de control de versiones.	13
ACTUALIZACIÓN 1	14
ACTUALIZACIÓN 2	23
7. (RA6_g) Se ha documentado la instalación, configuración y uso del sistema de control de versiones utilizado.	24
8. CONSEJOS Y RECOMENDACIONES PARA REALIZAR LA TAREA.	24

## 1. Documentos que se adjuntan a este informe.

A continuación se detallan los documentos que componen la presente entrega de la tarea:

1. Informe de elaboración de la tarea

## 2. CONTEXTO DE LA TAREA

El proceso y software de documentación así como las soluciones adoptadas en el control de versiones en los proyectos informáticos es un problema que debe ser resuelto habitualmente en el entorno de trabajo. Asimismo, el uso de determinadas técnicas de trabajo, pueden condicionar el uso de las herramientas en esta etapa del proceso de desarrollo e implantación de soluciones. En la tarea que deberá realizar, simulará una situación habitual en su entorno de trabajo: documentará un código fuente y generará la documentación automáticamente del mismo y trabajará con un sistema de control de versiones.

## 3. ENUNCIADO DE LA TAREA

Para realizar esta tarea es necesario que cumpla estos requisitos mínimos:

- El alumno/a debe entregar un informe perfectamente detallado, con capturas descriptivas del proceso. En dicho informe, debe incluir las capturas de todos los comandos utilizados para la ejecución de la tarea, no se debe dar ninguno por supuesto.
- Se debe utilizar el DNI o NombreApellido del alumno/a para describir la máquina virtual en VirtualBox o el nombre del servidor Ubuntu

Esta tarea consiste en realizar las siguientes sub-tareas:

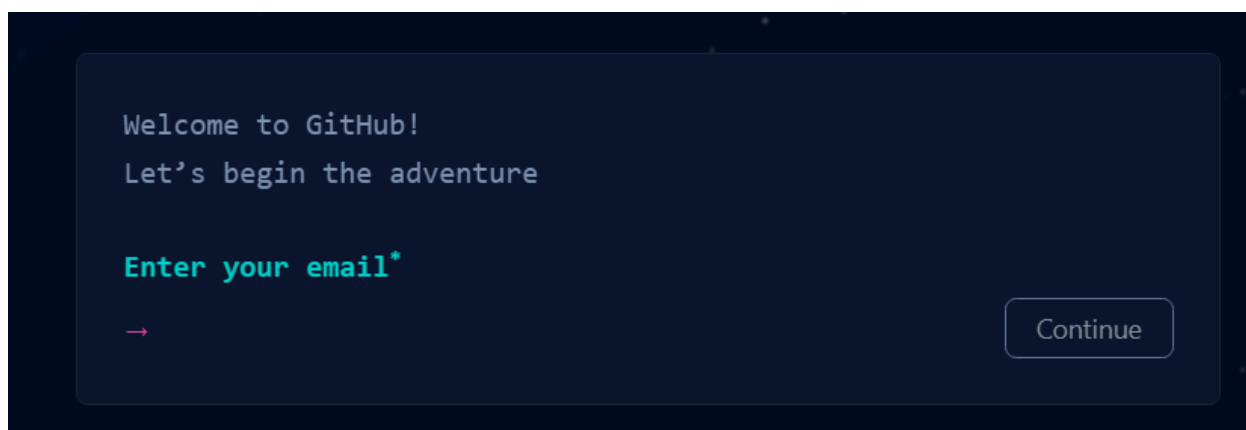
## 4. (RA6\_d) Se han utilizado herramientas colaborativas para la elaboración y mantenimiento de la documentación.

- El alumno/a se registrará en GitHub, creará un repositorio para la documentación.

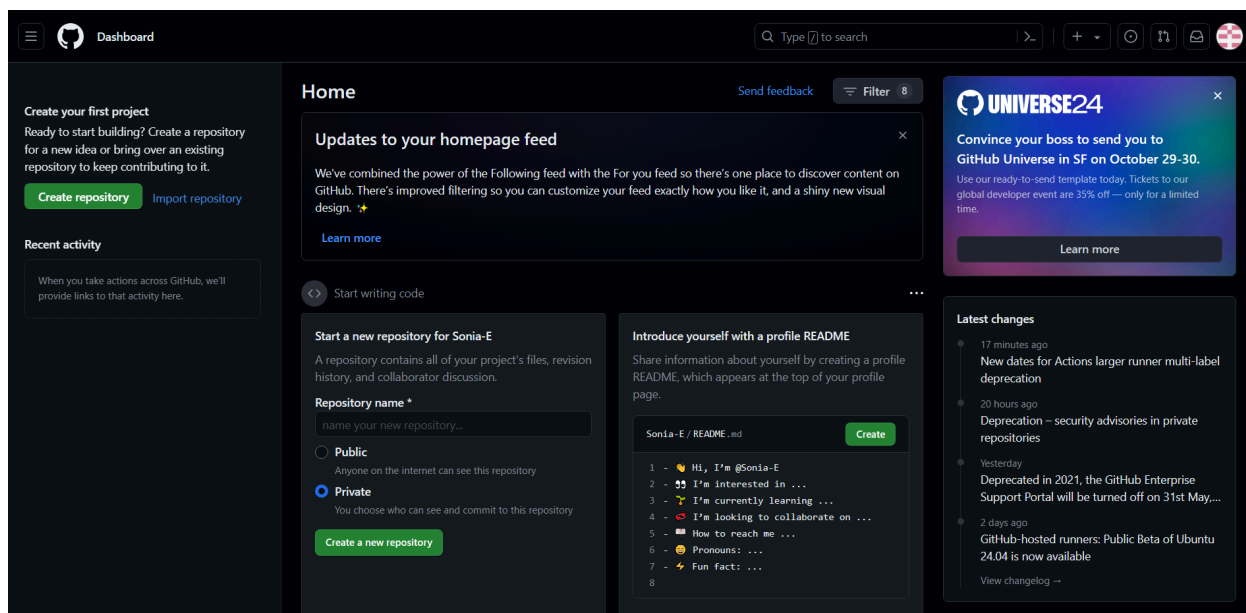
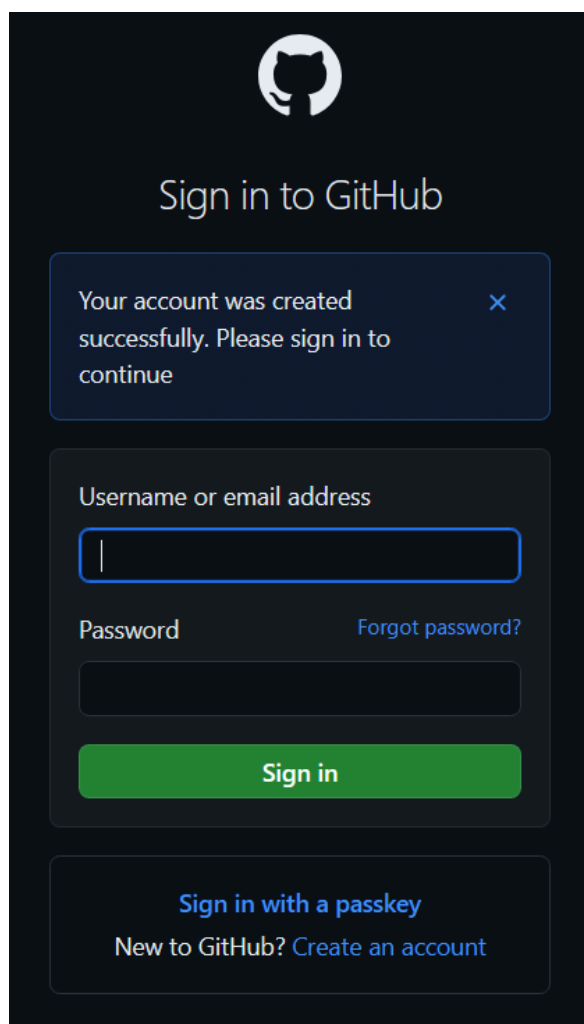
Ya me tuve que registrar en GitHub cuando realicé la tarea 6 de la asignatura ED del

año pasado, por lo que ya usé mi correo principal para ello y no tengo otro. Por ello, no me volveré a registrar aquí sino que compartiré las capturas que en su entonces hice para registrarme:

Para poder crear un repositorio en github, primero tenemos que estar registrados. Así que nos registraremos:



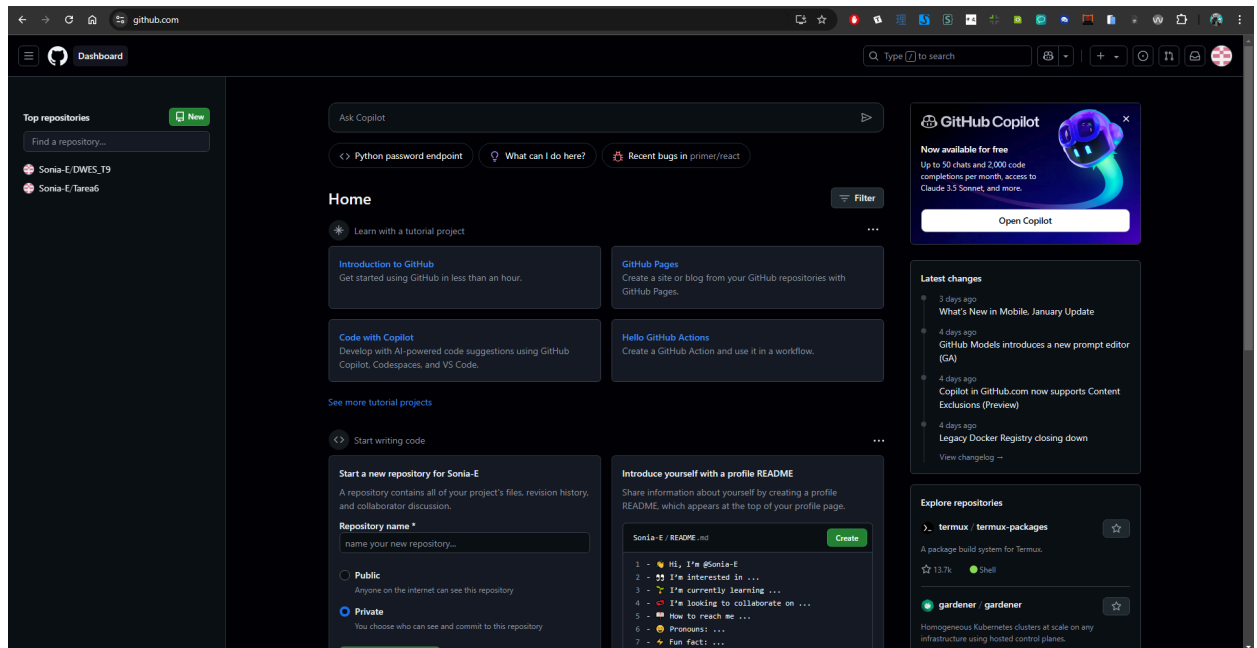
Y al darle a “Continue” deberemos elegir una contraseña y un nombre de usuario. Finalizado el proceso ya ingresaremos con nuestra cuenta creada en el siguiente menú:



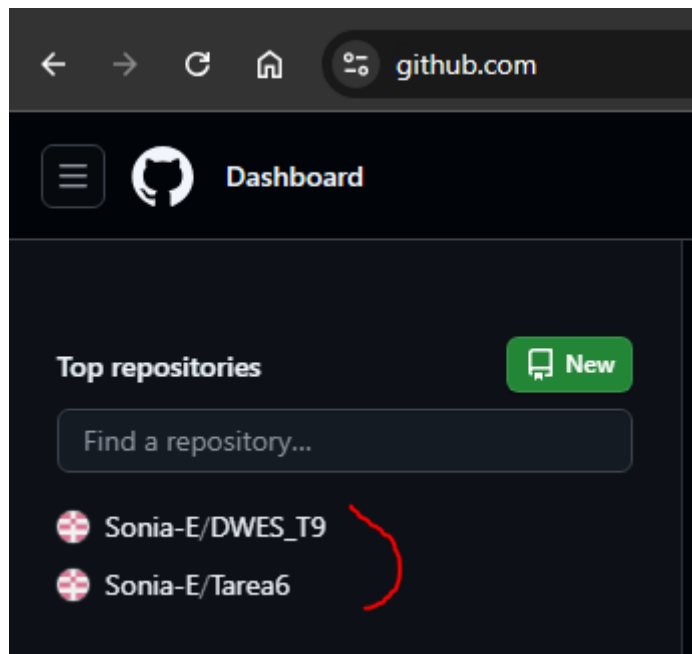
Como podemos ver, como era en su entonces una nueva cuenta ya directamente nos aparecía a la izquierda del todo que creásemos un repositorio a través del botón verde

“Create repository”.

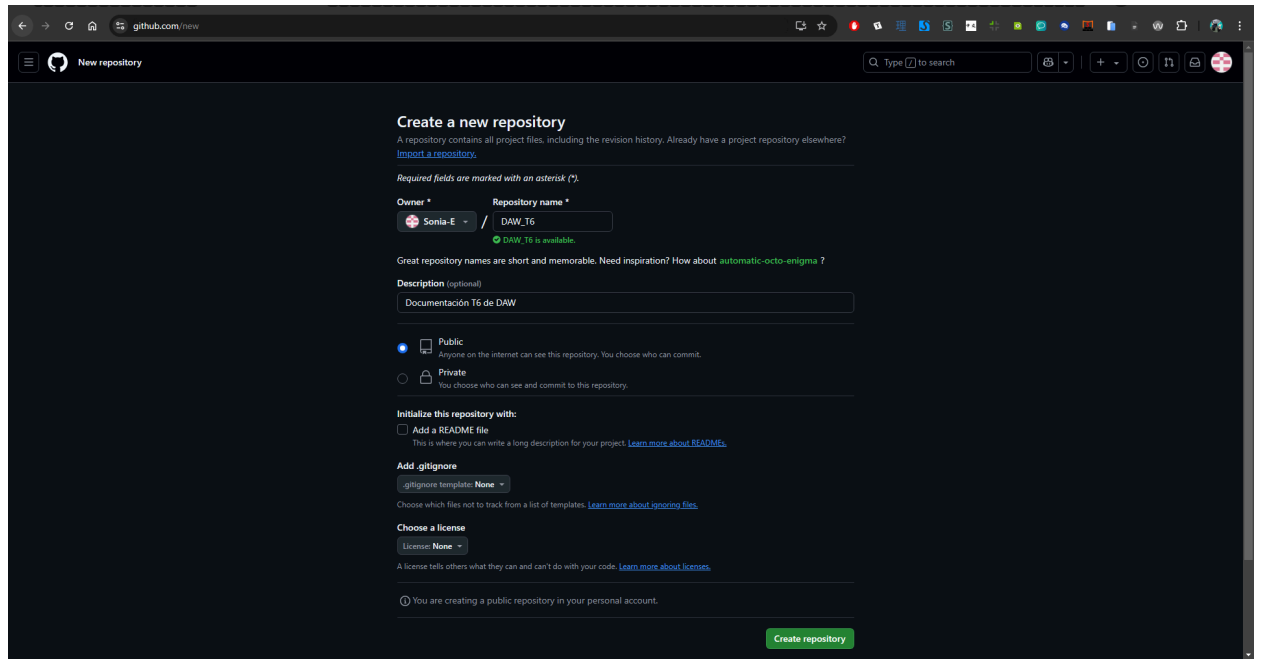
Ahora, en cambio, al iniciar sesión nos aparece lo siguiente:



Se puede ver a la izquierda como tengo ya dos repositorios creados:



Para crear un nuevo repositorio le daremos al botón verde “New” que se ve en la captura anterior:



github.com/new

### 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 \* Sonia-E / Repository name \* DAW\_T6  
DAW\_T6 is available.

Great repository names are short and memorable. Need inspiration? How about [automatic-octo-enigma](#)?

Description (optional)  
Documentación T6 de DAW

☒ 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.](#)

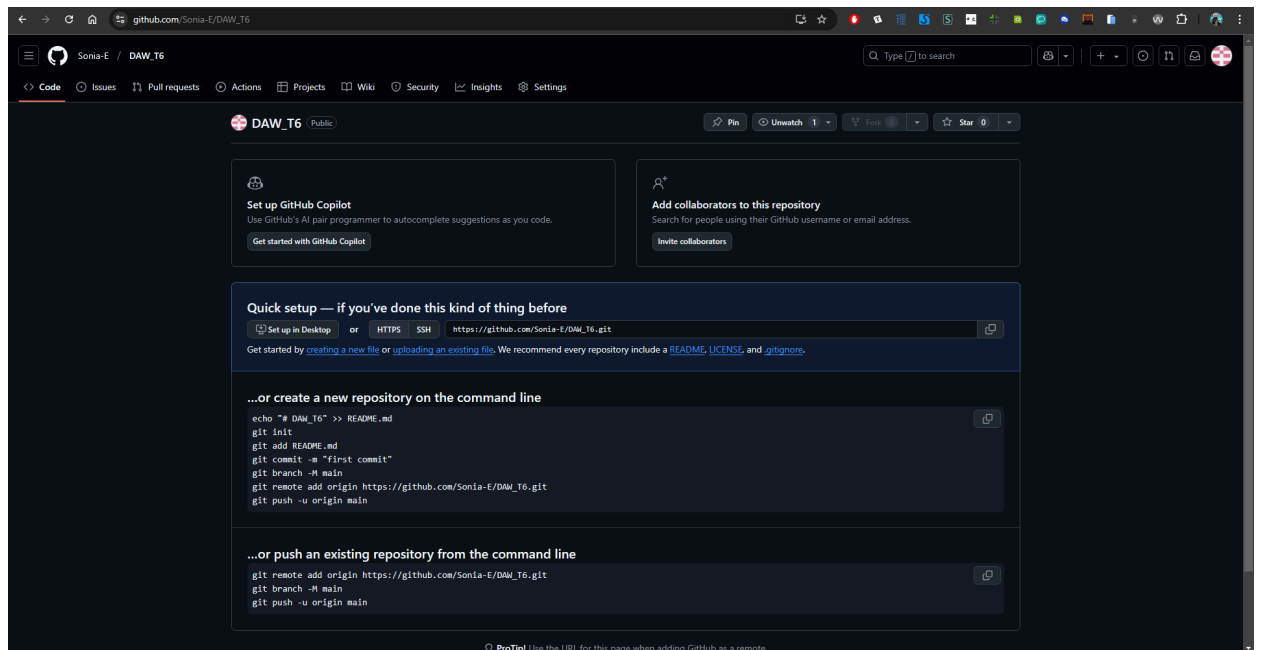
Add .gitignore  
gitignore template: None  
Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

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

① You are creating a public repository in your personal account.

[Create repository](#)

Llamamos al repositorio DAW\_T6 con descripción “Documentación T6 de DAW” y lo hacemos público. Por último, le damos a “Create repository”:



github.com/Sonia-E/DAW\_T6

DAW\_T6 Public

[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](#) [SSH](#) [https://github.com/Sonia-E/DAW\\_T6.git](https://github.com/Sonia-E/DAW_T6.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 "# DAW_T6" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/Sonia-E/DAW_T6.git
git push -u origin main
```

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

```
git remote add origin https://github.com/Sonia-E/DAW_T6.git
git branch -M main
git push -u origin main
```

ProTip! Use the URL for this page when adding GitHub as a remote.

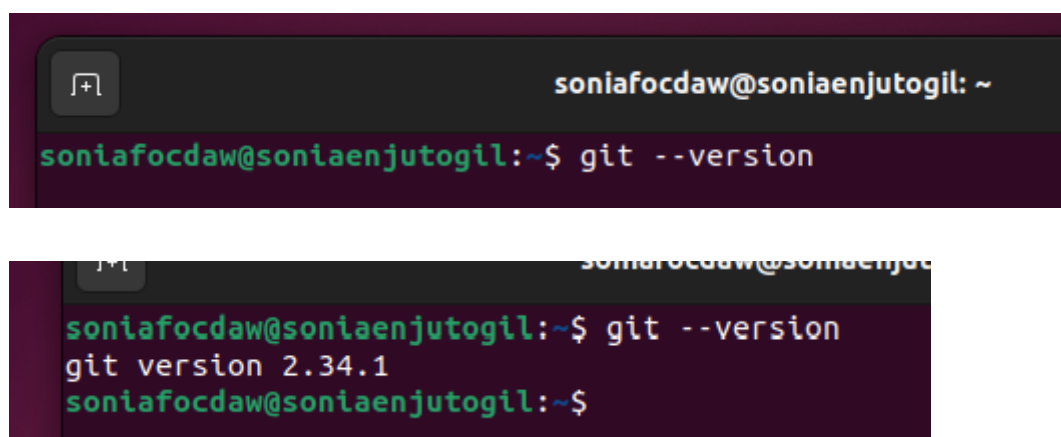
Y ahí lo tenemos.



## 5. (RA6\_e) Se ha instalado, configurado y utilizado un sistema de control de versiones.

- El alumno/a instalará la aplicación git en su terminal y la sincronizará con su repositorio.

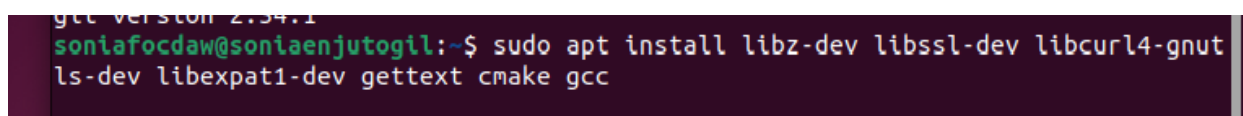
En Ubuntu ya está instalado por defecto Git. Podemos comprobarlo con el siguiente comando:



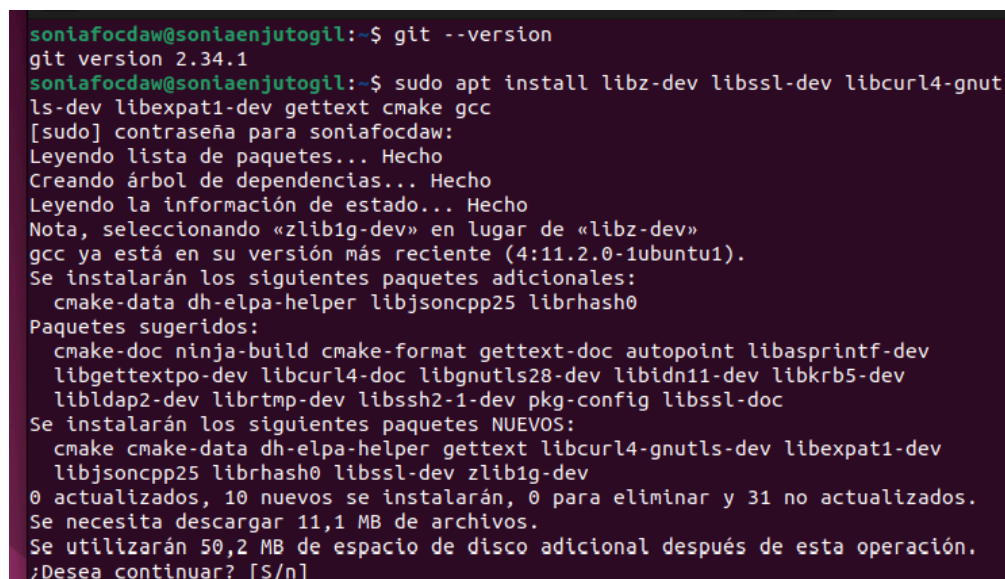
```
soniafocdaw@soniaenjutogil: ~
soniafocdaw@soniaenjutogil:~$ git --version

soniafocdaw@soniaenjutogil:~$ git --version
git version 2.34.1
soniafocdaw@soniaenjutogil:~$
```

No obstante, es recomendable instalar una versión más actualizada. Para ello, primero debemos instalar unas librerías de las que git depende. Lo haremos con el siguiente comando:



```
git version 2.34.1
soniafocdaw@soniaenjutogil:~$ sudo apt install libz-dev libssl-dev libcurl4-gnutls-dev libexpat1-dev gettext cmake gcc
```



```
soniafocdaw@soniaenjutogil:~$ git --version
git version 2.34.1
soniafocdaw@soniaenjutogil:~$ sudo apt install libz-dev libssl-dev libcurl4-gnutls-dev libexpat1-dev gettext cmake gcc
[sudo] contraseña para soniafocdaw:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Nota, seleccionando «zlib1g-dev» en lugar de «libz-dev»
gcc ya está en su versión más reciente (4:11.2.0-1ubuntu1).
Se instalarán los siguientes paquetes adicionales:
  cmake-data dh-elpa-helper libjsoncpp25 librhash0
Paquetes sugeridos:
  cmake-doc ninja-build cmake-format gettext-doc autopoint libasprintf-dev libgettextpo-dev libcurl4-doc libgnutls28-dev libidn11-dev libkrb5-dev libldap2-dev librtmp-dev libssh2-1-dev pkg-config libssl-doc
Se instalarán los siguientes paquetes NUEVOS:
  cmake cmake-data dh-elpa-helper gettext libcurl4-gnutls-dev libexpat1-dev libjsoncpp25 librhash0 libssl-dev zlib1g-dev
0 actualizados, 10 nuevos se instalarán, 0 para eliminar y 31 no actualizados.
Se necesita descargar 11,1 MB de archivos.
Se utilizarán 50,2 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n]
```

Le damos a sí con “S”:

```
soniafocdaw@soniaenjutogil: ~
d64 3.0.2-0ubuntu1.18 [2.375 kB]
Des:10 http://es.archive.ubuntu.com/ubuntu jammy-updates/main amd64 zlib1g-dev a
md64 1:1.2.11.dfsg-2ubuntu9.2 [164 kB]
Descargados 11,1 MB en 3s (4.005 kB/s)
Seleccionando el paquete libjsoncpp25:amd64 previamente no seleccionado.
(Leyendo la base de datos ... 214687 ficheros o directorios instalados actualmen
te.)
Preparando para desempaquetar .../0-libjsoncpp25_1.9.5-3_amd64.deb ...
Desempaquetando libjsoncpp25:amd64 (1.9.5-3) ...
Seleccionando el paquete librhash0:amd64 previamente no seleccionado.
Preparando para desempaquetar .../1-librhash0_1.4.2-1ubuntu1_amd64.deb ...
Desempaquetando librhash0:amd64 (1.4.2-1ubuntu1) ...
Seleccionando el paquete dh-elpa-helper previamente no seleccionado.
Preparando para desempaquetar .../2-dh-elpa-helper_2.0.9ubuntu1_all.deb ...
Desempaquetando dh-elpa-helper (2.0.9ubuntu1) ...
Seleccionando el paquete cmake-data previamente no seleccionado.
Preparando para desempaquetar .../3-cmake-data_3.22.1-1ubuntu1.22.04.2_all.deb .
..
Desempaquetando cmake-data (3.22.1-1ubuntu1.22.04.2) ...
Seleccionando el paquete cmake previamente no seleccionado.
Preparando para desempaquetar .../4-cmake_3.22.1-1ubuntu1.22.04.2_amd64.deb ...
Desempaquetando cmake (3.22.1-1ubuntu1.22.04.2) ...
Progreso: [ 22%] [#####.....]
```

```
soniafocdaw@soniaenjutogil: ~
Configurando libexpat1-dev:amd64 (2.4.7-1ubuntu0.5) ...
Configurando dh-elpa-helper (2.0.9ubuntu1) ...
Configurando libssl-dev:amd64 (3.0.2-0ubuntu1.18) ...
Configurando libjsoncpp25:amd64 (1.9.5-3) ...
Configurando zlib1g-dev:amd64 (1:1.2.11.dfsg-2ubuntu9.2) ...
Configurando librhash0:amd64 (1.4.2-1ubuntu1) ...
Configurando cmake-data (3.22.1-1ubuntu1.22.04.2) ...
Configurando cmake (3.22.1-1ubuntu1.22.04.2) ...
Procesando disparadores para libc-bin (2.35-0ubuntu3.8) ...
Procesando disparadores para man-db (2.10.2-1) ...
Procesando disparadores para install-info (6.8-4build1) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
soniafocdaw@soniaenjutogil:~$
```

Ahora ya podemos instalar git. Para ello realizamos el siguiente comando para descargarnos la versión 2.9.5 de la página oficial:

```
soniafocdaw@soniaenjutogil:~$ sudo curl -o git.tar.gz https://mirrors.edge.kerne
l.org/pub/software/scm/git/git-2.9.5.tar.gz && mv -f git.tar.gz /tmp
```

```
soniafocdaw@soniaenjutogil: ~
soniafocdaw@soniaenjutogil:~$ sudo curl -o git.tar.gz https://mirrors.edge.kernel.org/pub/software/scm/git/git-2.9.5.tar.gz && mv -f git.tar.gz /tmp
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left   Speed
100 5789k  100 5789k    0     0  6807k      0  --:--:-- --:--:-- --:--:-- 6803k
soniafocdaw@soniaenjutogil:~$
```

Ya descargado, ahora tenemos que descomprimirlo ejecutando la siguiente secuencia de comandos:

```
100 5789k  100 5789k    0     0  6807k
soniafocdaw@soniaenjutogil:~$ cd /tmp
```

```
soniafocdaw@soniaenjutogil:~$ cd /tmp
soniafocdaw@soniaenjutogil:/tmp$
```

```
soniafocdaw@soniaenjutogil:/tmp$ tar -zxf git.tar.gz
```

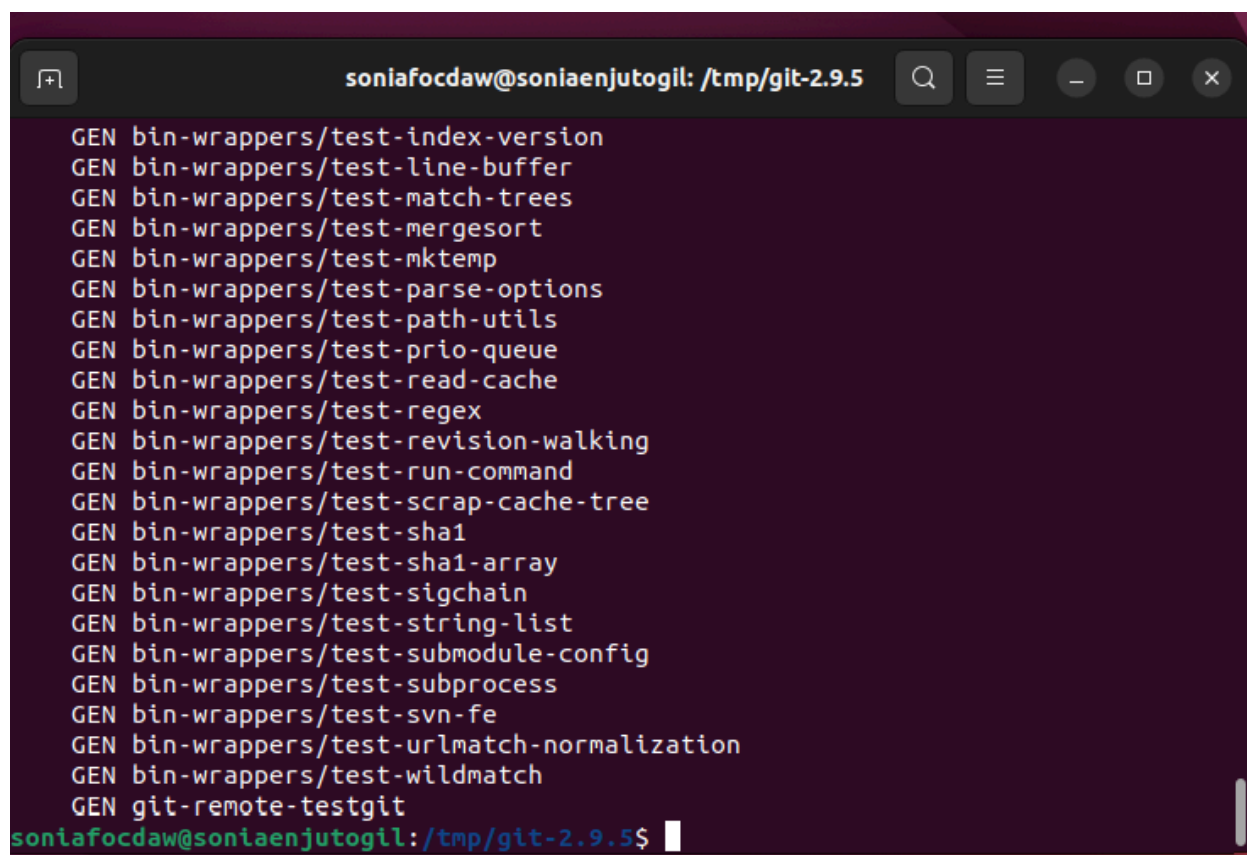
```
soniafocdaw@soniaenjutogil:/tmp$ tar -zxf git.tar.gz
soniafocdaw@soniaenjutogil:/tmp$ cd git-*
```

```
soniafocdaw@soniaenjutogil:/tmp$ tar -zxf git.tar.gz
soniafocdaw@soniaenjutogil:/tmp$ cd git-*
soniafocdaw@soniaenjutogil:/tmp/git-2.9.5$
```

Ya dentro del directorio con la fuente de git, compilaremos la aplicación localmente ejecutando el siguiente comando:

```
soniafocdaw@soniaenjutogil:/tmp$ cd git-*
soniafocdaw@soniaenjutogil:/tmp/git-2.9.5$ sudo make prefix=/usr/local all
```

```
soniafocdaw@soniaenjutogil:/tmp$ cd git-*
soniafocdaw@soniaenjutogil:/tmp/git-2.9.5$ sudo make prefix=/usr/local all
GIT_VERSION = 2.9.5
* new build flags
CC credential-store.o
* new link flags
CC common-main.o
CC abspath.o
CC advice.o
CC alias.o
CC alloc.o
CC archive.o
CC archive-tar.o
CC archive-zip.o
CC argv-array.o
```



```
soniafocdaw@soniaenjutogil: /tmp/git-2.9.5
GEN bin-wrappers/test-index-version
GEN bin-wrappers/test-line-buffer
GEN bin-wrappers/test-match-trees
GEN bin-wrappers/test-mergesort
GEN bin-wrappers/test-mktemp
GEN bin-wrappers/test-parse-options
GEN bin-wrappers/test-path-utils
GEN bin-wrappers/test-prio-queue
GEN bin-wrappers/test-read-cache
GEN bin-wrappers/test-regex
GEN bin-wrappers/test-revision-walking
GEN bin-wrappers/test-run-command
GEN bin-wrappers/test-scrap-cache-tree
GEN bin-wrappers/test-sha1
GEN bin-wrappers/test-sha1-array
GEN bin-wrappers/test-sigchain
GEN bin-wrappers/test-string-list
GEN bin-wrappers/test-submodule-config
GEN bin-wrappers/test-subprocess
GEN bin-wrappers/test-svn-fe
GEN bin-wrappers/test-urlmatch-normalization
GEN bin-wrappers/test-wildmatch
GEN git-remote-testgit
soniafocdaw@soniaenjutogil: /tmp/git-2.9.5$
```

Ahora instalaremos la aplicación con el siguiente comando:

```
soniafocdaw@soniaenjutogil: /tmp/git-2.9.5$ sudo make prefix=/usr/local install
```

```

soniafocdaw@soniaenjutogil: /tmp/git-2.9.5
install -d -m 755 '/usr/local/libexec/git-core/mergetools'
install -m 644 mergetools/* '/usr/local/libexec/git-core/mergetools'
install -d -m 755 '/usr/local/share/locale'
(cd po/build/locale && tar cf - .) | \
(cd '/usr/local/share/locale' && umask 022 && tar xof -)
make -C perl prefix='/usr/local' DESTDIR='' install
make[1]: se entra en el directorio '/tmp/git-2.9.5/perl'
make[2]: se entra en el directorio '/tmp/git-2.9.5/perl'
Manifying 8 pod documents
Installing /usr/local/share/perl/5.34.0/Git.pm
Installing /usr/local/share/perl/5.34.0/Git/SVN.pm
Installing /usr/local/share/perl/5.34.0/Git/I18N.pm
Installing /usr/local/share/perl/5.34.0/Git/IndexInfo.pm
Installing /usr/local/share/perl/5.34.0/Git/SVN/Migration.pm
Installing /usr/local/share/perl/5.34.0/Git/SVN/Ra.pm
Installing /usr/local/share/perl/5.34.0/Git/SVN/GlobSpec.pm
Installing /usr/local/share/perl/5.34.0/Git/SVN/Utils.pm
Installing /usr/local/share/perl/5.34.0/Git/SVN/Editor.pm
Installing /usr/local/share/perl/5.34.0/Git/SVN/Fetcher.pm
Installing /usr/local/share/perl/5.34.0/Git/SVN/Log.pm
Installing /usr/local/share/perl/5.34.0/Git/SVN/Prompt.pm
Installing /usr/local/share/perl/5.34.0/Git/SVN/Memoize/YAML.pm
Installing /usr/local/share/man/man3/Git::SVN::Prompt.3pm

```

```

soniafocdaw@soniaenjutogil: /tmp/git-2.9.5
k git-reflog git-remote git-remote-ext git-remote-fd git-repack git-replace git-
rerere git-reset git-rev-list git-rev-parse git-revert git-rm git-send-pack git-
shortlog git-show-branch git-show-ref git-stripspace git-submodule--helper git-s
ymbolic-ref git-tag git-unpack-file git-unpack-objects git-update-index git-upda
te-ref git-update-server-info git-upload-archive git-var git-verify-commit git-v
erify-pack git-verify-tag git-worktree git-write-tree git-cherry git-cherry-pick
git-format-patch git-fsck-objects git-init git-merge-subtree git-show git-stage
git-status git-whatchanged; do \
    rm -f "$execdir/$p" && \
    test -z "" && \
    ln "$execdir/git" "$execdir/$p" 2>/dev/null || \
    ln -s "git" "$execdir/$p" 2>/dev/null || \
    cp "$execdir/git" "$execdir/$p" || exit; \
done && \
remote_curl_aliases="git-remote-https git-remote-ftp git-remote-ftps" && \
for p in $remote_curl_aliases; do \
    rm -f "$execdir/$p" && \
    test -z "" && \
    ln "$execdir/git-remote-http" "$execdir/$p" 2>/dev/null || \
    ln -s "git-remote-http" "$execdir/$p" 2>/dev/null || \
    cp "$execdir/git-remote-http" "$execdir/$p" || exit; \
done && \
./check_bindir "z$bindir" "z$execdir" "$bindir/git-add"
soniafocdaw@soniaenjutogil: /tmp/git-2.9.5$

```

Por último, actualizaremos las variables del sistema con el comando:



```
soniafocdaw@soniaenjutogil: /tmp/git-2.9.5
soniafocdaw@soniaenjutogil:/tmp/git-2.9.5$ exec bash
```

Comprobamos que se ha instalado correctamente:

```
soniafocdaw@soniaenjutogil:~$ git --version
```

```
soniafocdaw@soniaenjutogil:~$ git --version
git version 2.9.5
soniafocdaw@soniaenjutogil:~$
```

Y efectivamente ahora tenemos la versión 2.9.5.

A continuación, tenemos que configurar git por medio de la herramienta git config. Crearemos nuestro usuario git. Primero estableceremos el nombre:

```
git version 2.9.5
soniafocdaw@soniaenjutogil:~$ git config --global user.name "SoniaEnjuto"
```

Ahora el correo:

```
soniafocdaw@soniaenjutogil:~$ git config --global user.name "SoniaEnjuto"
soniafocdaw@soniaenjutogil:~$ git config --global user.email soniaenjutom94@gmail.com
```

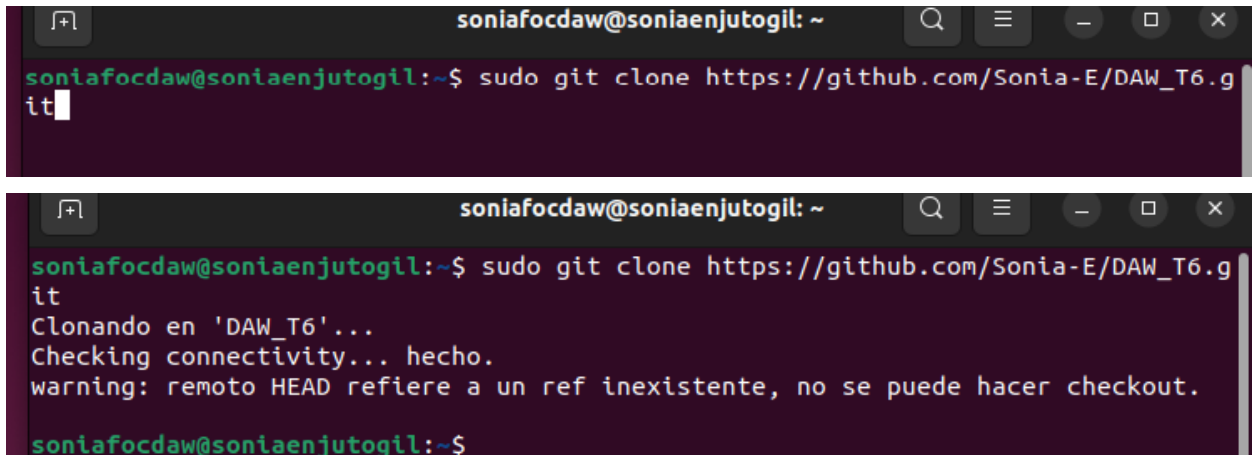
Comprobamos que se han añadido ambos:

```
soniafocdaw@soniaenjutogil:~$ git config --list
```

```
soniafocdaw@soniaenjutogil:~$ git config --list
user.name=SoniaEnjuto
user.email=soniaenjutom94@gmail.com
soniafocdaw@soniaenjutogil:~$
```

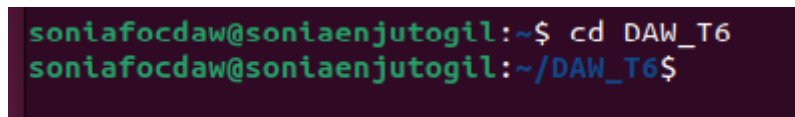
CREACIÓN REPOSITORIO LOCAL CLONANDO

Vamos entonces a clonar nuestro repositorio creado en Github para hacerlo local y sincronizar ambos así. Para ello usaremos `sudo git clone` y la URL de nuestro repositorio en Github:



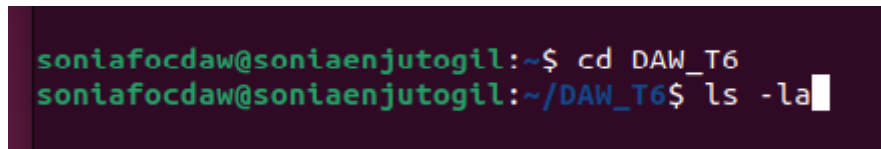
The first screenshot shows the command `sudo git clone https://github.com/Sonia-E/DAW_T6.git` being entered in a terminal window titled `soniafocdaw@soniaenjutogil: ~`. The second screenshot shows the output of the command: `Clonando en 'DAW_T6'...`, `Checking connectivity... hecho.`, and a warning: `warning: remoto HEAD refiere a un ref inexistente, no se puede hacer checkout.`. The prompt returns to `soniafocdaw@soniaenjutogil:~$`.

Ahora nos movemos dentro de la copia local:



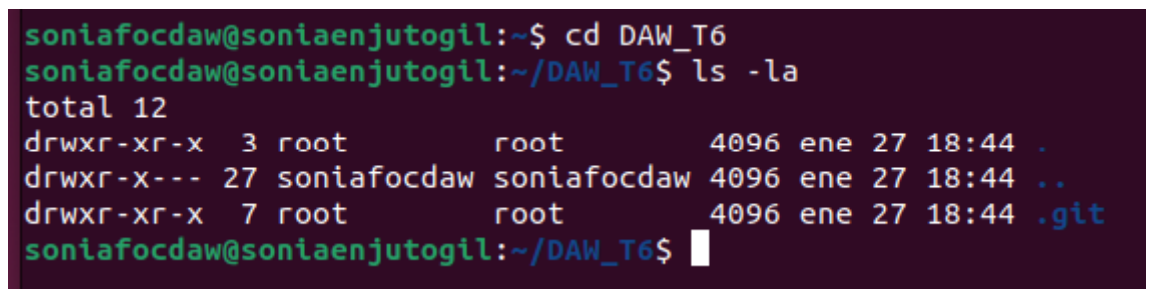
The terminal shows the command `cd DAW_T6` being executed, changing the current directory to `~/DAW_T6`. The prompt is now `soniafocdaw@soniaenjutogil:~/DAW_T6$`.

Listamos contenido:



The terminal shows the command `ls -la` being executed in the `DAW_T6` directory. The output shows the directory listing with permissions, owner, group, size, date, and file names.

Como podemos ver nuestro repositorio **ya está inicializado**, por eso tiene el subdirectorio oculto `“.git”`; si no no estaría:



The terminal shows the command `ls -la` being executed in the `DAW_T6` directory. The output shows the directory listing with permissions, owner, group, size, date, and file names. The `.git` directory is listed with permissions `drwxr-xr-x`, owner `root`, group `root`, size `4096`, date `ene 27 18:44`, and name `.git`.

6. (RA6\_f) Se ha garantizado la accesibilidad y seguridad de la documentación almacenada por el sistema de control de

versiones.

- El alumno/a realizará varias actualizaciones del repositorio, por ejemplo subiendo en primer lugar los documentos HTML y posteriormente el PDF.

## ACTUALIZACIÓN 1

Ahora vamos a coger el index.html del código fuente facilitado para esta tarea; lo copiaremos a nuestro repositorio con el siguiente comando:

```
soniafocdaw@soniaenjutogil:~/DAW_T6$ sudo cp ~/Descargas/código\ fuente/index.html .
```

El punto al final marca nuestra posición actual, para que se copie donde estamos, en nuestro repositorio.

Ahora sí añadimos el documento con git add:

```
soniafocdaw@soniaenjutogil:~/DAW_T6$ sudo git add index.html
```

Comprobamos que se ha añadido listando:

```
soniafocdaw@soniaenjutogil:~/DAW_T6$ ls -la
total 16
drwxr-xr-x  3 root      root      4096 ene 27 18:58 .
drwxr-x--- 27 soniafocdaw soniafocdaw 4096 ene 27 18:44 ..
drwxr-xr-x  7 root      root      4096 ene 27 18:59 .git
-rw-r--r--  1 root      root        379 ene 27 18:58 index.html
soniafocdaw@soniaenjutogil:~/DAW_T6$
```

Hacemos git commit y añadimos una descripción a este punto de control:

```
soniafocdaw@soniaenjutogil:~/DAW_T6$ sudo git commit -m "Agregué el archivo index.html"
```



```
soniafocdaw@soniaenjutogil:~/DAW_T6$ sudo git commit -m "Agregué el archivo index.html"
[master (commit-raíz) 64542d6] Agregué el archivo index.html
Committer: root <root@soniaenjutogil.foc.local>
Tu nombre y correo fueron configurados automáticamente basados
en tu usuario y nombre de host. Por favor verifica que sean correctos.
Tú puedes suprimir este mensaje configurándolos de forma explícita. Ejecuta el
siguiente comando y sigue las instrucciones de tu editor
para modificar tu archivo de configuración:

    git config --global --edit

Tras hacer esto, puedes arreglar la identidad usada para este commit con:

    git commit --amend --reset-author

1 file changed, 15 insertions(+)
create mode 100644 index.html
soniafocdaw@soniaenjutogil:~/DAW_T6$
```

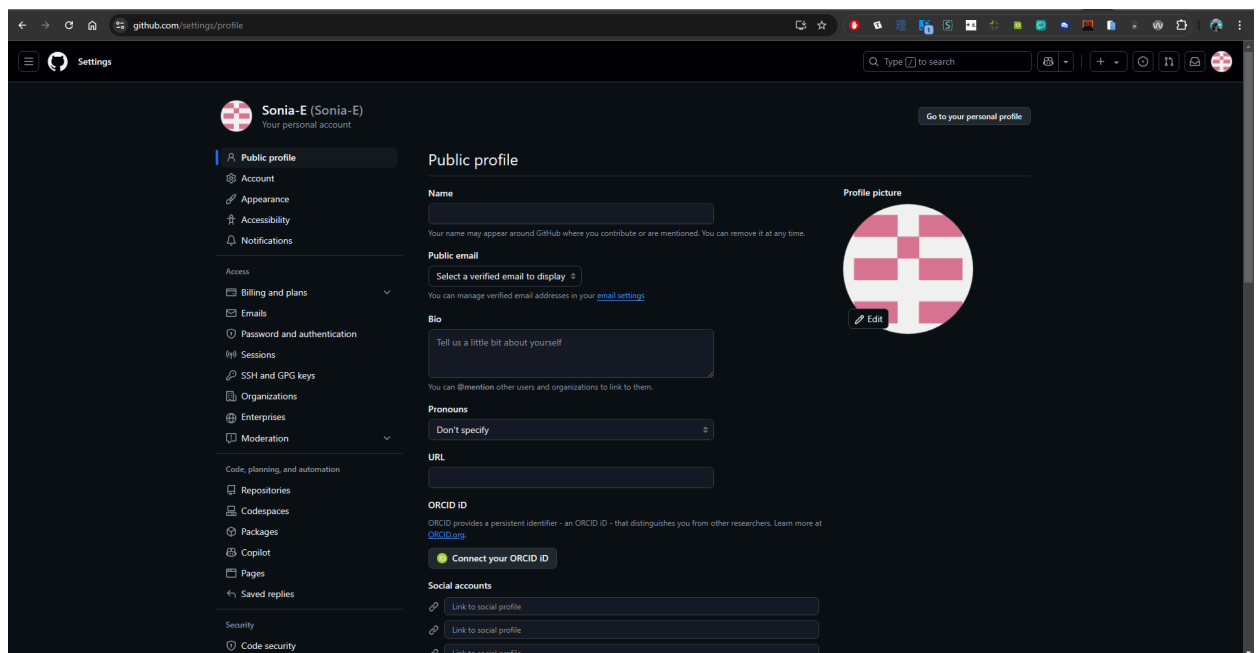
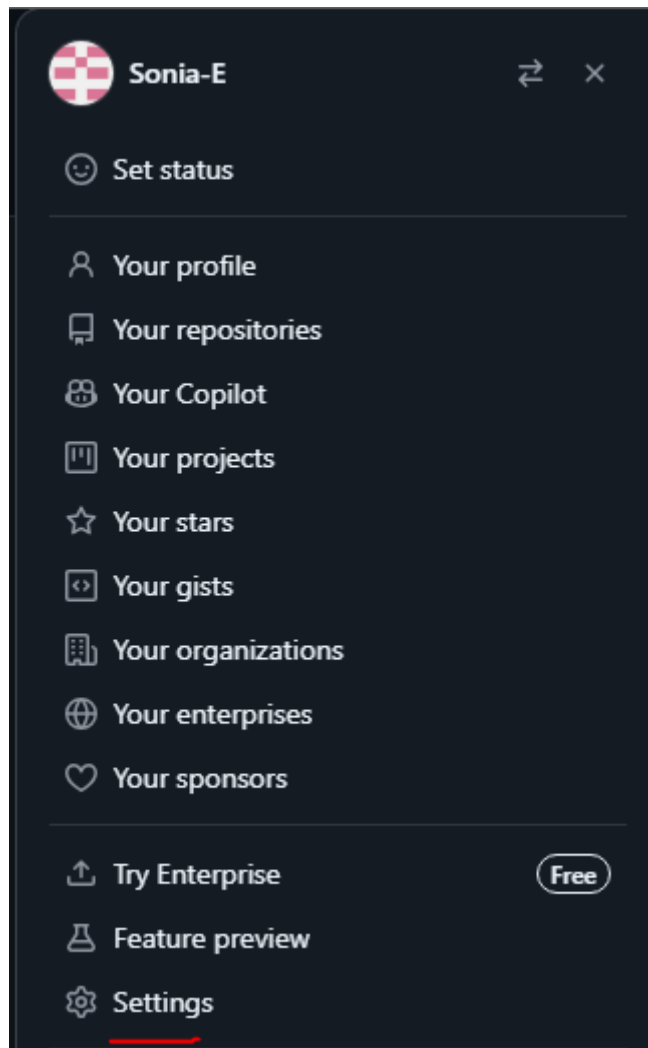
Hacemos git status para ver dónde nos encontramos:

```
soniafocdaw@soniaenjutogil:~/DAW_T6$ git status
En la rama master
nada para hacer commit, el árbol de trabajo está limpio
soniafocdaw@soniaenjutogil:~/DAW_T6$
```

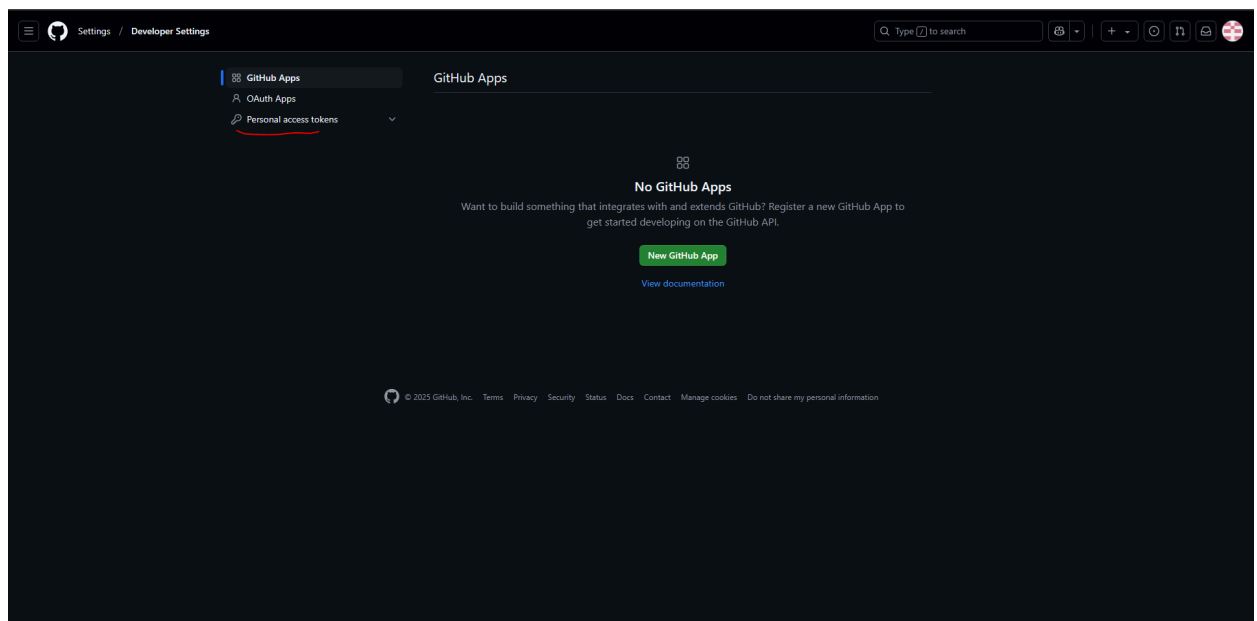
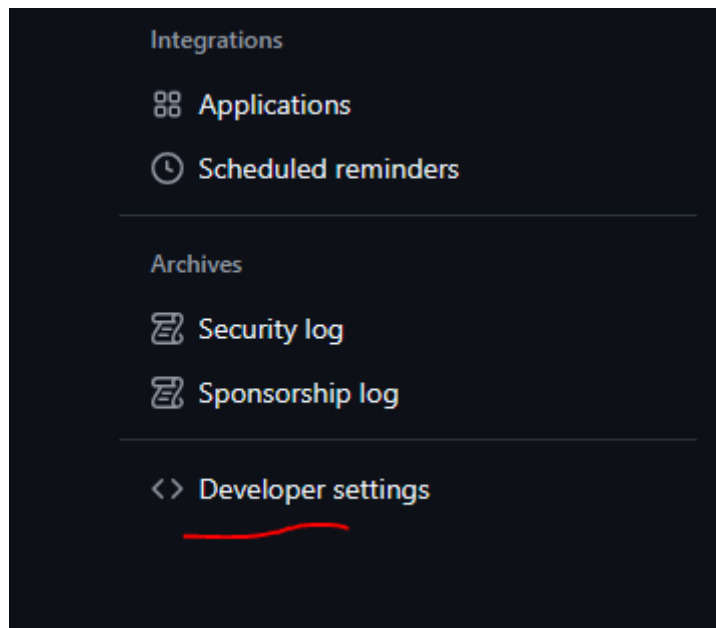
Estamos en la rama master, por lo que el push tendremos que hacerlo con esta rama y usando “origin” para referirnos al repositorio original en Github. Con el uso de push mandaremos el archivo añadido a nuestro repositorio en Github.

Antes de hacer el push, tenemos que crear un TOKEN como contraseña para poder enviar el documento.

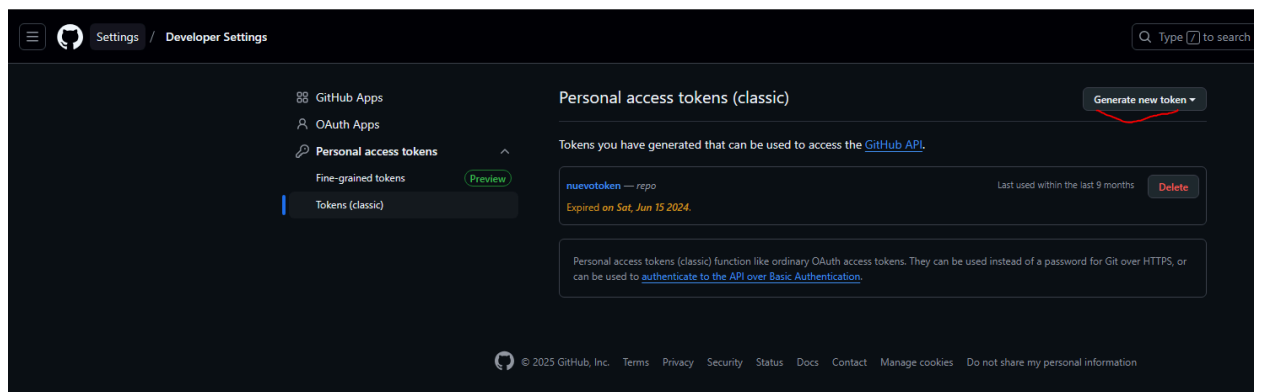
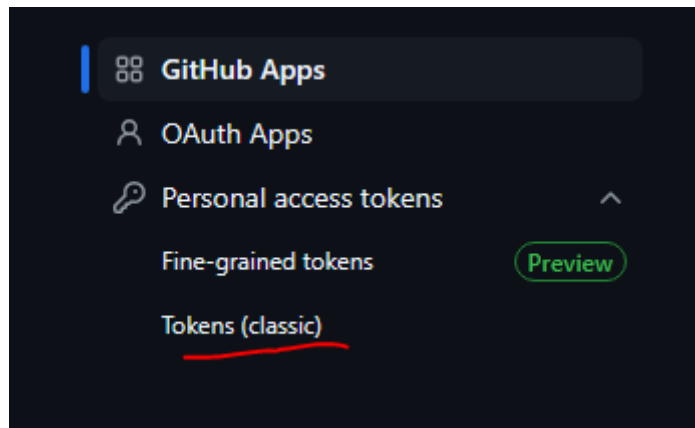
Para ello nos vamos a los settings de nuestro Github:



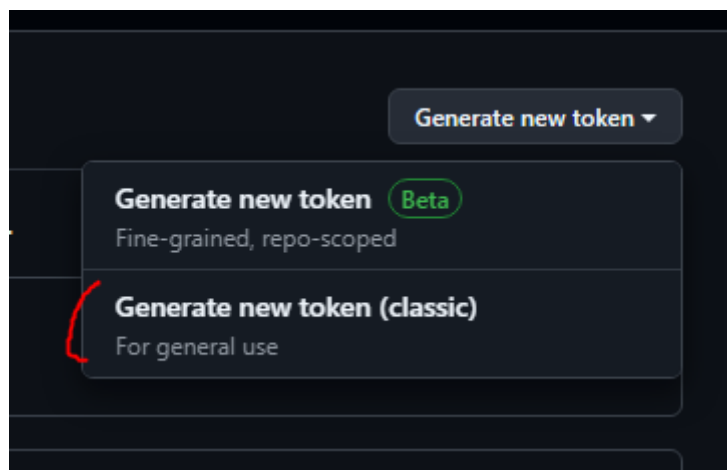
Aquí bajaremos y nos iremos a “Developer settings”:




Aquí nos vamos a “Personal access tokens” y a “Tokens (classic)”:




Y seguimos los siguientes pasos marcados con rojo para crear nuestro token:





Confirm access



Signed in as @Sonia-E

Password

[Forgot password?](#)

Confirm

Tip: You are entering [sudo mode](#). After you've performed a sudo-protected action, you'll only be asked to re-authenticate again after a few hours of inactivity.

GitHub Apps

OAuth Apps

Personal access tokens Preview

Fine-grained tokens

Tokens (classic)

## New personal access token (classic)

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

**Note**

DAW

What's this token for?

**Expiration \***

30 days The token will expire on Thu, Feb 27 2025

**Select scopes**

Scopes define the access for personal tokens. [Read more about OAuth scopes.](#)

<input checked="" type="checkbox"/> repo	Full control of private repositories
<input checked="" type="checkbox"/> repo:status	Access commit status
<input checked="" type="checkbox"/> repo_deployment	Access deployment status
<input checked="" type="checkbox"/> public_repo	Access public repositories
<input checked="" type="checkbox"/> repo:invite	Access repository invitations
<input checked="" type="checkbox"/> security_events	Read and write security events
<input type="checkbox"/> workflow	Update GitHub Action workflows
<input type="checkbox"/> write:packages	Upload packages to GitHub Package Registry
<input type="checkbox"/> read:packages	Download packages from GitHub Package Registry
<input type="checkbox"/> delete:packages	Delete packages from GitHub Package Registry
<input type="checkbox"/> admin:org	Full control of orgs and teams, read and write org projects
<input type="checkbox"/> write:org	Read and write org and team membership, read and write org projects
<input type="checkbox"/> read:org	Read org and team membership, read org projects

Le ponemos un nombre, dejamos los 30 días de validez y solo marcamos la opción "repo". Le damos a generarlo:

☐ write:gpg\_key

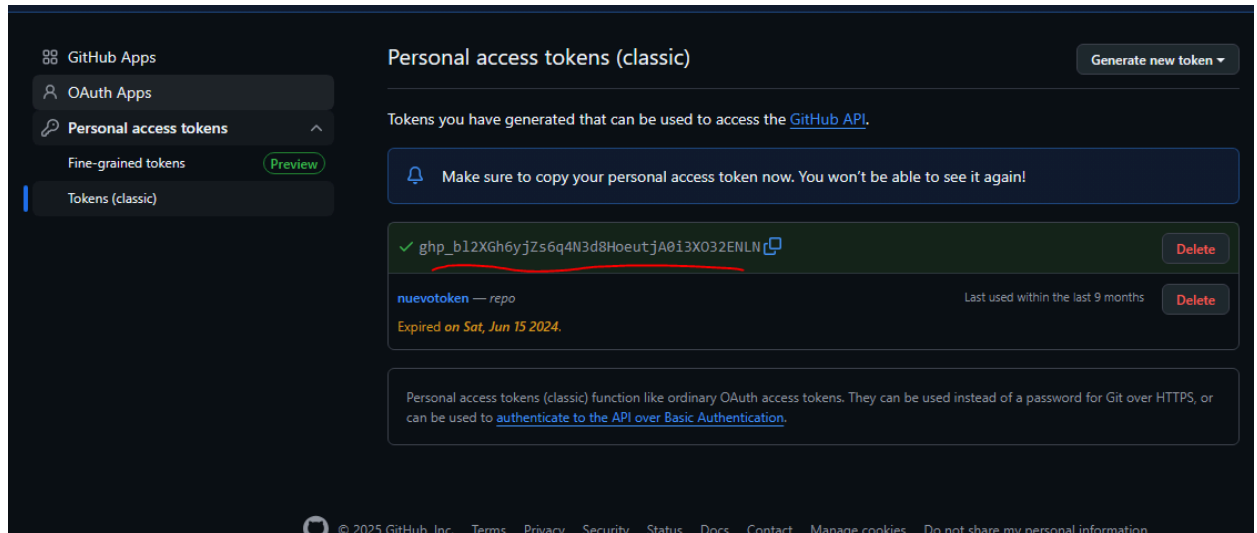
☐ read:gpg\_key

☐ admin:ssh\_signing\_key

☐ write:ssh\_signing\_key

☐ read:ssh\_signing\_key

**Generate token** **Cancel**



Copiamos lo marcado en rojo, nos volvemos a la terminal y hacemos git push:

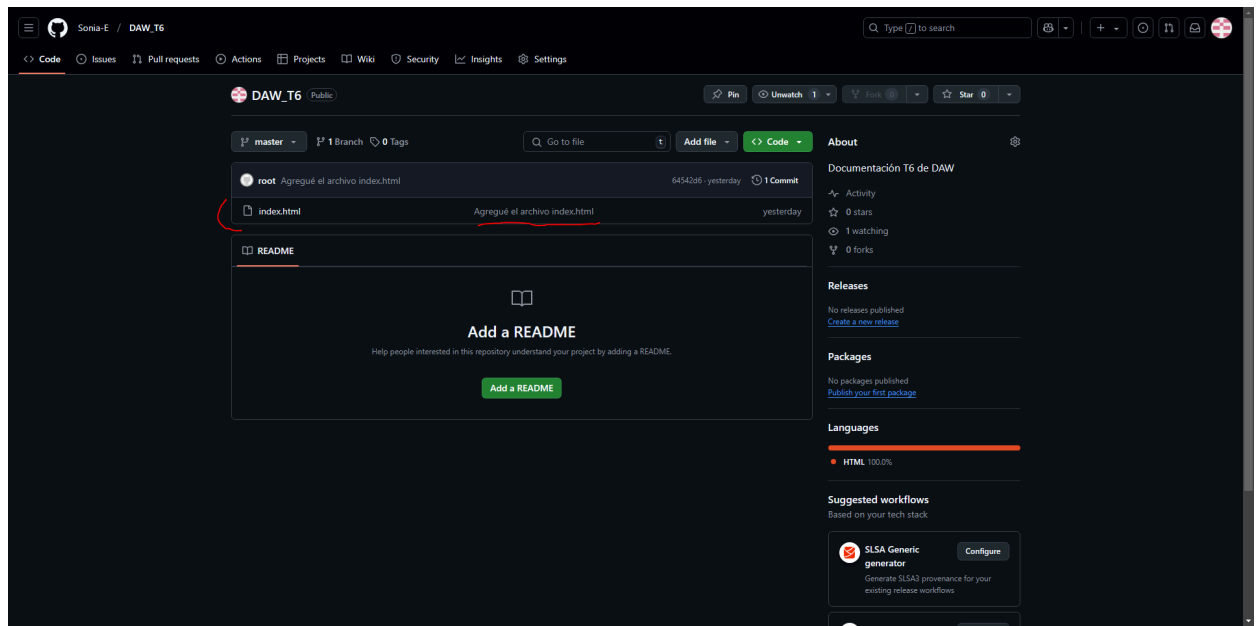
```
soniafocdaw@soniaenjutogil:~/DAW_T6$ git push origin master
```

En la contraseña pegamos el TOKEN:

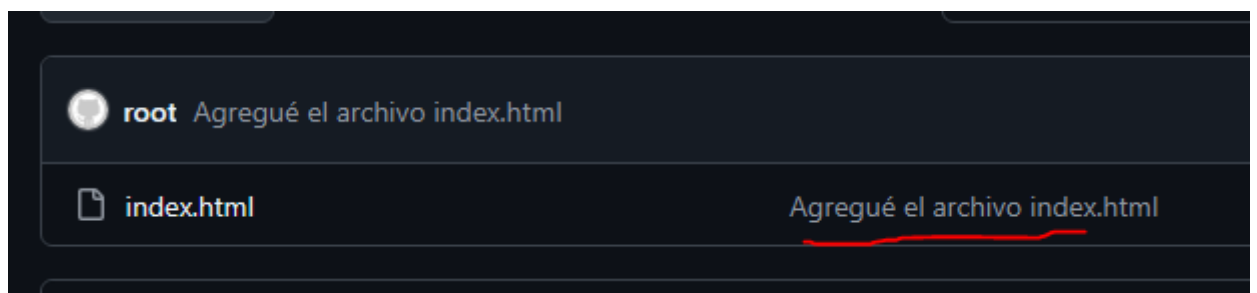
```
soniafocdaw@soniaenjutogil:~/DAW_T6$ git push origin master
Username for 'https://github.com': Sonia-E
Password for 'https://Sonia-E@github.com':
```

```
soniafocdaw@soniaenjutogil:~/DAW_T6$ git push origin master
Username for 'https://github.com': Sonia-E
Password for 'https://Sonia-E@github.com':
Contando objetos: 3, listo.
Comprimiendo objetos: 100% (2/2), listo.
Escribiendo objetos: 100% (3/3), 481 bytes | 0 bytes/s, listo.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/Sonia-E/DAW_T6.git
 * [new branch]      master -> master
error: update_ref failed for ref 'refs/remotes/origin/master': cannot lock ref '
refs/remotes/origin/master': unable to create directory for .git/refs/remotes/or
igin/master
soniafocdaw@soniaenjutogil:~/DAW_T6$
```

Podemos comprobar que se ha añadido el documento HTML a nuestro repositorio en Github:

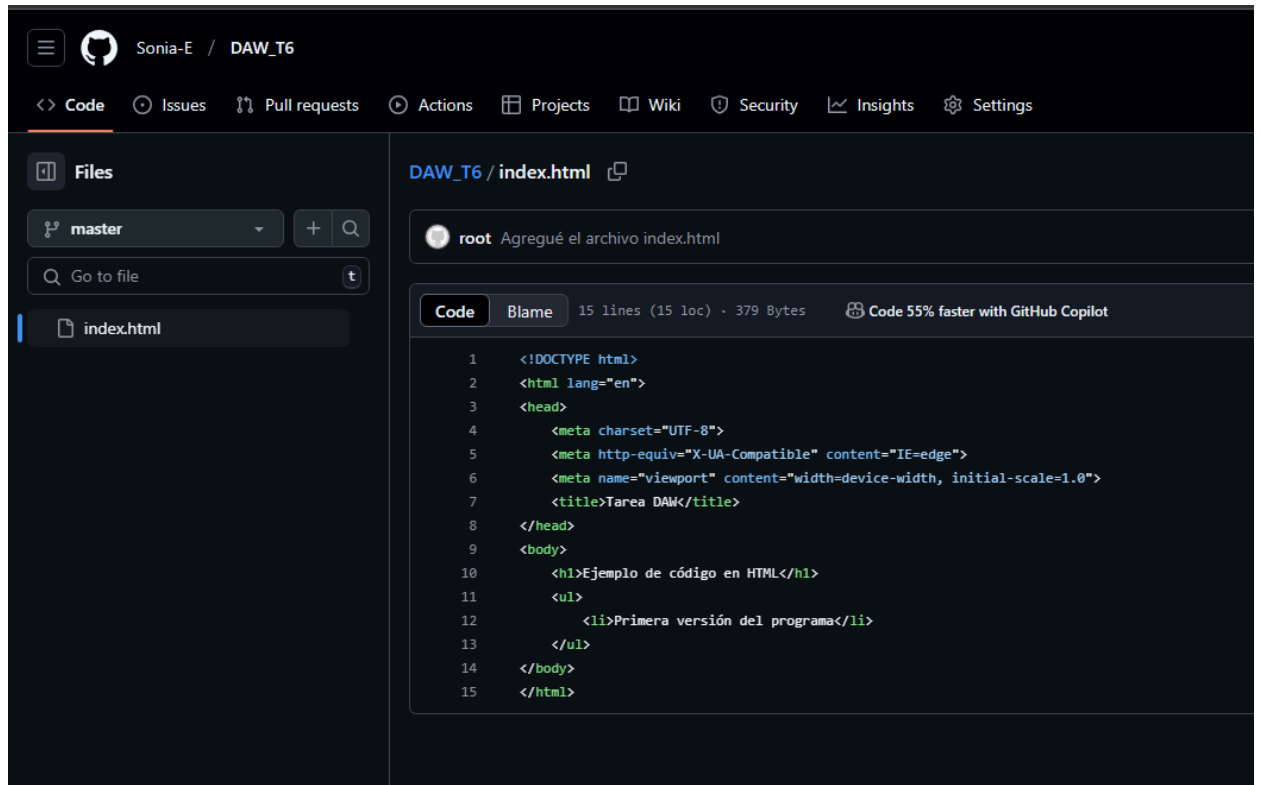


Podemos ver hasta el mensaje que añadimos con el commit:



Si nos metemos dentro del documento, veremos su código:





## ACTUALIZACIÓN 2

Ahora vamos a realizar el mismo proceso pero con el PDF de este informe. Se va subir a ambos repositorios una versión antigua del informe, puesto que para mostrar las capturas del proceso, he tenido que trabajar con esta versión no final de mi informe.

Nuevamente copiamos desde el directorio "Descargas" el PDF hasta nuestro repositorio local:

Realizamos git add:

Hacemos git commit:

Por último usamos git push haciendo uso del TOKEN para la contraseña:

Resultado en Github:

## 7. (RA6\_g) Se ha documentado la instalación, configuración y uso del sistema de control de versiones utilizado.

- El alumno/a debe desarrollar y recoger en el informe los pasos necesarios para satisfacer los RA6\_d, RA6\_e y RA6\_f

Primero en el RA6\_d debíamos registrarnos en Github, pero como yo ya tenía cuenta por una tarea del año pasado, he compartido en este informe las capturas del registro. Después, hemos iniciado sesión para crear un repositorio que hemos llamado "DAW\_T6".

Después en el RA6\_e hemos instalado una versión mucho más actual de Git que la que viene por defecto en Ubuntu Linux. Después, para sincronizarnos con nuestro repositorio de Github, hemos usado git clone para crear una copia en local que ya mantendrá una sincronización con nuestro repositorio en Github, teniendo la referencia "origin" para referirse a este repositorio original.

Por último, en el RA6\_f, hemos realizado dos actualizaciones:

- Actualización 1: hemos añadido el documento HTML a nuestro repositorio local, luego lo hemos incluido en el control de versiones con git add, hecho git commit y finalmente git push hacia "origin", nuestro repositorio original en Github.
- Actualización 2: hemos añadido el PDF (correspondiente a este informe) a nuestro repositorio local, luego lo hemos incluido en el control de versiones con git add, hecho git commit y finalmente git push hacia "origin", nuestro repositorio original en Github.

En este caso, se ha subido a ambos repositorios una versión antigua del informe, puesto que para mostrar las capturas del proceso, he tenido que trabajar con esta versión no final de mi informe.

## 8. CONSEJOS Y RECOMENDACIONES PARA REALIZAR LA TAREA.

Para realizar esta tarea tener en cuenta las siguientes recomendaciones:

1. Leer la unidad 6 con detenimiento, en especial la sección donde se trata Git.
2. Una vez registrados en GitHub, configurar debidamente el repositorio creado para la actividad.