

2020/12

Practica 02 DAW

Git control de versiones

Preparado por:

Isabel González Anzano

1) PREPARACIÓN DEL ESCENARIO. 0 PUNTOS.	2
1.a) Creación de una cuenta en Github	2
1.b) Asociar las claves ssh	2
1.c) Configuración de la terminal con zsh y oh my Zsh (no es obligatorio pero sí recomendable)	4
2) TRABAJANDO CON COMANDOS DE GIT EN LOCAL. (0,5 CADA APARTADO)	5
2.a) Estructura de carpetas del directorio .git	5
Añadir usuario y email a git	5
Carpetas principales .git	5
2.b) Creando archivos - Stage Area	7
Comandos interesantes	8
2.c) Commit	8
2.d) Trabajando con commits	8
Comandos interesantes	9
2.e) Historial de Commits - git log	9
Historial mostrando una línea por commit git log --oneline	9
Historial de commits - git log	9
Comandos interesantes	10
2.f) Tags	10
Comandos interesantes	10
Información adicional	10
2.g) Archivo .gitignore	11
2.h) Descartando cambios	11
Comandos interesantes	12
2.i) Cambios y commits	12
Comandos interesantes	13
2.j) Sobreescibir un commit	14
Comandos interesantes	14
3) TRABAJANDO CON REPOSITORIOS REMOTOS Y RAMAS (0.5PTOS CADA APARTADO)	15
3.a) Creando el repositorio en Github	15
3.b) Conexión remoto-local conexion ssh	15
3.c) Uso de la Interfaz web	16
3.d) Fusión de ramas	17
Pull request	18
Información útil	20
4) NETBEANS Y GIT (0.3 PTOS CADA APARTADO)	21
4.a) Proyecto en Netbeans e inicializacion de Git	21
4.b) Primer commit	22
4.c) Comparando commits	22
4.d) Subir proyecto a Repositorio por ssh	23
4.e) Modifica el proyecto en Github y actualízalo en local	24

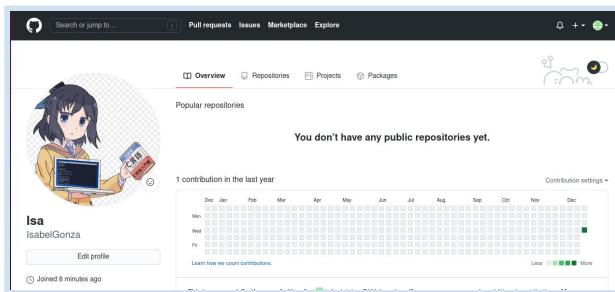
5) VISUAL STUDIO CODE (0.3 PTOS CADA APARTADO)	24
5.a) Proyecto en Visual Studio Code e inicialización de Git	24
5.b) Commit desde VS	25
5.c) git log --oneline	26
5.d) Subir proyecto por ssh - Push -	27
5.e) Modificaciones en GitHub y descarga a local - PULL -	28
6) ANEXO	30
Crear claves ssh en windows	30
Asignarlas a la cuenta de github	30
Información interesante	31
NetBeans & GIT	31
VisualStudio Code & GIT	31

1) PREPARACIÓN DEL ESCENARIO. 0 PUNTOS.

Este apartado solo es necesario para aquellas personas que no hayan podido crear y configurar su cuenta en Github.com durante el transcurso del curso. En caso de ya disponer de una cuenta configurada pasar al apartado 2.

1.a) Creación de una cuenta en Github

- Entre en <https://github.com/> y registra una cuenta en github.
- Acepta el email de confirmación.
- Logueate con tu cuenta en Github.com



Instalar git:

```
isa@isaserver:/var/log/apache2/empresa$ sudo apt-get install git
Reading package lists... Done
Building dependency tree
Reading state information... Done
git is already the newest version (1:2.25.1-1ubuntu3).
git set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 19 not upgraded.
```

1.b) Asociar las claves ssh

- Chequea si tienes claves ssh ya generadas.
- <https://help.github.com/articles/checking-for-existing-ssh-keys/>

```
ls -al ~/.ssh
```

```
isa@isaserver:/var/log/apache2/empresa$ ls -al ~/.ssh
total 8
drwx----- 2 isa isa 4096 Dec  9 10:03 .
drwxr-xr-x 16 isa isa 4096 Dec 12 17:11 ..
```

- Genera una nueva clave ssh y añadelas al ssh-agent.

<https://docs.github.com/es/free-pro-team@latest/github/authenticating-to-github/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent>

Generar la clave ssh

```
ssh-keygen -t rsa -b 4096 -C "igonzalezanzano@gmail.com"
```

```
isa@isaserver:/var/log/apache2/empresa$ ssh-keygen -t rsa -b 4096 -C "igonzalezanzano@gmail.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/isa/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/isa/.ssh/id_rsa
Your public key has been saved in /home/isa/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:IcbE0aIzVqb+LkpPFADcOcACM4yQTmhkCDQ1hA/L7/Q igonzalezanzano@gmail.com
The key's randomart image is:
+---[RSA 4096]---+
|~B=...
|Bxo++o+
|B ++..+.
|.+==... .
| o.= S
|. +o
|..oo..
|o o. E
|..
+---[SHA256]---+
```

Añadir la ssh al agente

```
ssh-add ~/.ssh/id_rsa
```

```
;isa@isaserver:/var/log/apache2/empresa$ ssh-add ~/.ssh/id_rsa
Enter passphrase for /home/isa/.ssh/id_rsa:
Identity added: /home/isa/.ssh/id_rsa (igonzalezanzano@gmail.com)
```

- Añade tu nueva clave ssh a tu perfil de github.com

<https://docs.github.com/es/free-pro-team@latest/github/authenticating-to-github/adding-a-new-ssh-key-to-your-github-account>

Copiar la key en el portapapeles, para ello usamos clip

```
$ sudo apt-get install xclip
# Downloads and installs xclip. If you don't have `apt-get`, you might need
to use another installer (like `yum`)
$ xclip -sel clip < home/isa/.ssh/id_rsa.pub
# Copies the contents of the id_rsa.pub file to your clipboard
```

Pegarla en Settings/SSH and GPG keys => SSH keys

SSH keys

New SSH key

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

	MVUbuntu20 DAW	Delete
SSH	SHA256:WVuWAcF7IubezFbGnGNXGCu6KINAgNsB7BJ08H02orQ	
	Added on Dec 23, 2020	
	Never used — Read/write	

Check out our guide to [generating SSH keys](#) or troubleshoot [common SSH Problems](#).

1.c) Configuración de la terminal con zsh y oh my Zsh (no es obligatorio pero sí recomendable)

“Oh my Zsh!” es una herramienta que nos permite aplicar temas a nuestro prompt y agregar diversos plugins a nuestra shell Zsh. Nosotros lo vamos a usar para tener información integrada de GIT en nuestro terminal, como por ejemplo información del repositorio Git en el que nos encontramos.

Oh my zsh requiere instalar el shell zsh o z shell.

Fuente: <https://github.com/robbyrussell/oh-my-zsh/blob/master/README.md>

Instalar zsh

```
sudo apt install zsh
```

```
isa@isaserver:~$ sudo apt install zsh  
Reading package lists... Done
```

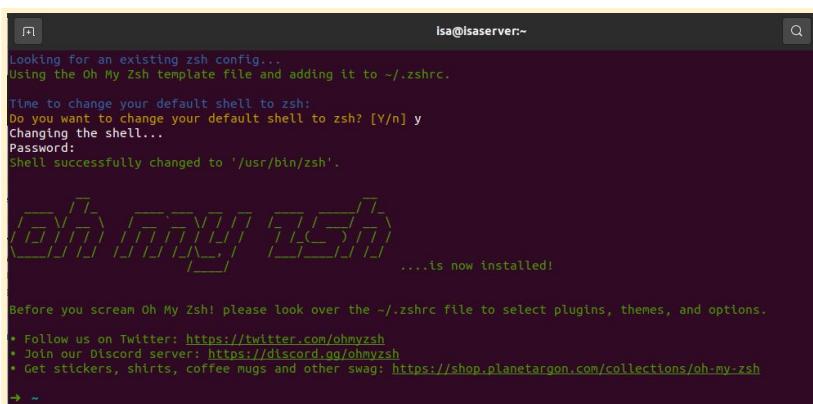
Configurarlo como el shell por defecto

```
chsh -s $(which zsh)
```

```
isa@isaserver:~$ chsh -s $(which zsh)
Password:
```

Instalación oh my zsh

```
sh -c "$(wget https://raw.github.com/ohmyzsh/ohmyzsh/master/tools/install.sh -O -)"
```



2) TRABAJANDO CON COMANDOS DE GIT EN LOCAL. (0,5 CADA APARTADO)

2.a) Estructura de carpetas del directorio .git

Crea un directorio nuevo de trabajo llamado “proyectoGit” que será la zona de trabajo del proyecto que vas a controlar con Git. Entra en el directorio “proyectoGit” e inicializa un repositorio. Investiga sobre el directorio .git creado al inicializar el proyecto. ¿Qué carpetas contiene y para qué sirven?

Creo el directorio en /home/isa

```
→ ~ ls
Desktop Documents Downloads Music Pictures Public Templates Videos mod-pagespeed-beta_current_amd64.deb
→ ~ cd ..
→ ~/home ls
empleado isa marketing prueba
→ ~/home cd isa
→ ~ mkdir proyectoGit
→ ~ ls
Desktop Documents Downloads Music Pictures Public Templates Videos mod-pagespeed-beta_current_amd64.deb proyectoGit
```

Añadir usuario y email a git

```
git config --global user.name "IsabelGonza"
git config --global user.email "igonalezanzano@gmail.com"
```

Verificar: sudo git config --list

```
git config --list
user.name=IsabelGonza
user.email=igonalezanzano@gmail.com
```

git init en /home/isa/proyectoGit

```
→ ~ git init /home/isa/proyectoGit
Initialized empty Git repository in /home/isa/proyectoGit/.git/
→ proyectoGit git:(master) ls -la
total 12
drwxr-xr-x 3 isa isa 4096 Dec 23 11:27 .
drwxr-xr-x 19 isa isa 4096 Dec 23 12:27 ..
drwxr-xr-x 7 isa isa 4096 Dec 23 11:27 .git
```

Carpetas principales .git

```
→ .git git:(master) ls
HEAD branches config description hooks info objects refs
→ .git git:(master) tree
.
├── HEAD
├── branches
├── config
├── description
└── hooks
    ├── applypatch-msg.sample
    ├── commit-msg.sample
    ├── fsmonitor-watchman.sample
    ├── post-update.sample
    ├── pre-applypatch.sample
    ├── pre-commit.sample
    ├── pre-merge-commit.sample
    ├── pre-push.sample
    ├── pre-rebase.sample
    ├── pre-receive.sample
    └── prepare-commit-msg.sample
        └── update.sample
.
├── info
│   └── exclude
.
├── objects
│   └── Info
│       └── pack
.
└── refs
    ├── heads
    └── tags
.
9 directories, 16 files
```

1. **HEAD**: El ref actual. En la mayoría de los casos éste es probablemente refs/heads/master.
2. **branches**: Contiene las ramas de nuestro repositorio para poder modificar nuestro proyecto sin modificar el proyecto original.
3. **config**: Contiene las opciones de configuración de este repositorio. La configuración específica a este repositorio se define en este archivo. El uso más común de este archivo es especificar la ubicación de cada repositorio remoto y definir algunas opciones clave, si tu repositorio posee o no un área de trabajo.
4. **description**: Si utilizas gitweb o ejecutas git instaweb, esta descripción se mostrará cuando estés viendo tu repositorio o la lista de repositorios.
5. **hooks**: Contiene distintos scripts que son ejecutados en determinados momentos al trabajar con Git, como por ejemplo después de cada commit.
6. **info**: Contiene el archivo exclude que sirve para ignorar archivos en este proyecto. Este archivo no está versionado como sí lo estaría un archivo .gitignore.
7. **objects**: El depósito interno de objetos, indexados por SHAs.
8. **refs**: La copia maestra de todos los refs que existen en tu repositorio, ya sean stashes, tags, ramas locales o remotas.

2.b) Creando archivos - Stage Area

Crea 2 archivos vacíos index_a.html y index_b.html en el directorio de trabajo.

Comprueba el estado de tu proyecto git. Añade los 2 ficheros al área de preparación (stage área) y confirma los cambios con el mensaje. “Inicializados ficheros index_a e index_b” ¿Cuántas ficheros cambiados, insertions (+) y deletions (-) aparecen? Añade contenido al index_a.html y al fichero index_b.html.

Git status para comprobar el estado

```
→ proyectoGit git:(master) git status
On branch master
No commits yet
nothing to commit (create/copy files and use "git add" to track)
```

Creamos los dos archivos vacíos en el directorio

```
→ proyectoGit git:(master) sudo touch index_a.html
[sudo] password for isa:
→ proyectoGit git:(master) X ls
index_a.html
→ proyectoGit git:(master) X sudo touch index_b.html
→ proyectoGit git:(master) X
```

Git status

```
→ proyectoGit git:(master) X git status
On branch master
No commits yet
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    index_a.html
    index_b.html
nothing added to commit but untracked files present (use "git add" to track
)
```

Añadir archivos a la zona de preparación: git add

```
→ proyectoGit git:(master) X git add index_a.html index_b.html
→ proyectoGit git:(master) X git status
On branch master
No commits yet
Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   index_a.html
    new file:   index_b.html
```

Commit con la frase indicada

```
→ proyectoGit git:(master) X git commit -m "Inicializados ficheros index_a e index_b"
[master (root-commit) 6852ead] Inicializados ficheros index_a e index_b
  2 files changed, 0 insertions(+), 0 deletions(-)
    create mode 100644 index_a.html
    create mode 100644 index_b.html
```

Añado contenido a los archivos

```
→ proyectoGit git:(master) sudo nano index_a.html
→ proyectoGit git:(master) X sudo nano index_b.html
→ proyectoGit git:(master) X git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   index_a.html
    modified:   index_b.html
no changes added to commit (use "git add" and/or "git commit -a")
```

Aparecen como modificados pero no comiteados.

Comandos interesantes

```
git status - para comprobar el estado del repositorio
git add nombreArchivo - añadir archivos a la zona de preparación
git commit -m "Comentario del commit" - Commit con comentario
```

2.c) Commit

Pasa solo el fichero el index_a.html a la zona de preparación (stage area). Vuelve a comprobar el estado de tu repo. Confirma solo los cambios del fichero index_a.html con el mensaje “Añadidos detalles a index_a.html”. Vuelve a comprobar el estado de tu repo.

Pasamos index_a a stage area y lo vemos con git status

```
git add index_a.html
```

```
→ proyectoGit git:(master) ✘ git add index_a.html
→ proyectoGit git:(master) ✘ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   index_a.html

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   index_b.html
```

Comiteamos los cambios en index_a.html y comprobamos con git status

```
git commit -m "Añadidos detalles a index_a.html"
```

```
→ proyectoGit git:(master) ✘ git commit -m "Añadidos detalles a index_a.html"
[master 2601e97] Añadidos detalles a index_a.html
 1 file changed, 1 insertion(+)
→ proyectoGit git:(master) ✘ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   index_b.html

no changes added to commit (use "git add" and/or "git commit -a")
```

2.d) Trabajando con commits

Comprueba las diferencias entre el último commit y tu directorio de trabajo. Añade el index_b.html a la zona de preparación. Comprueba los cambios entre la zona de preparación y el último commit. Confirma los cambios con el mensaje “Añadidos detalles a index_b.html”.

Del paso anterior a este no he realizado ningún cambio, voy a utilizar los comandos git diff y git diff --staged para verificar esto.

```
→ git diff --git a/index_b.html b/index_b.html
index e69de29..96b648d 100644
--- a/index_b.html
+++ b/index_b.html
@@ -0,0 +1 @@
+soy index_b.html
```

Con el comando git diff --staged la salida es nula ya que no se añade ningún contenido nuevo al archivo y por tanto, es el mismo.

Añado el index_b.html al stage area: git add index_b.html y git status

```
→ proyectoGit git:(master) X git add index_b.html
→ proyectoGit git:(master) X git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   index_b.html
```

Commit con git commit -m “Añadidos detalles a index_b.html”

```
→ proyectoGit git:(master) X git commit -m "Añadidos detalles a index_b.html"
[master 54b366e] Añadidos detalles a index_b.html
 1 file changed, 1 insertion(+)
→ proyectoGit git:(master) git status
On branch master
nothing to commit, working tree clean
```

Comandos interesantes

git diff - establece las diferencias en las fuentes de datos de Git. Dichas fuentes de datos pueden ser confirmaciones, ramas y archivos, entre otras posibilidades.

2.e) Historial de Commits - git log

Comprueba el historial de commits del repositorio mostrando una línea por cada commit. Comprueba el historial de commits del repositorio mostrando los cambios en cada commit.

Historial mostrando una línea por commit git log --oneline

```
54b366e (HEAD -> master) Añadidos detalles a index_b.html
2601e97 Añadidos detalles a index_a.html
6852ead Inicializados ficheros index_a e index_b
```

Historial de commits - git log

```
commit 54b366e3099276c19aca1cb78e55f0325e2f3a09 (HEAD -> master)
Author: IsabelGonza <igonzalezanzano@gmail.com>
Date:   Wed Dec 23 14:05:09 2020 +0000

  Añadidos detalles a index_b.html

commit 2601e9780738381b806961732a97946c90b52db8
Author: IsabelGonza <igonzalezanzano@gmail.com>
Date:   Wed Dec 23 12:55:02 2020 +0000

  Añadidos detalles a index_a.html

commit 6852ead0b9256ae22cb8a21e73dcbdd4cbf1e30
Author: IsabelGonza <igonzalezanzano@gmail.com>
Date:   Wed Dec 23 12:46:21 2020 +0000

  Inicializados ficheros index_a e index_b
```

Comandos interesantes

git log - Muestra un historial de commits mostrando los cambios en cada commit
 git log --oneline - Comprueba el historial de commits mostrando una línea por cada commit

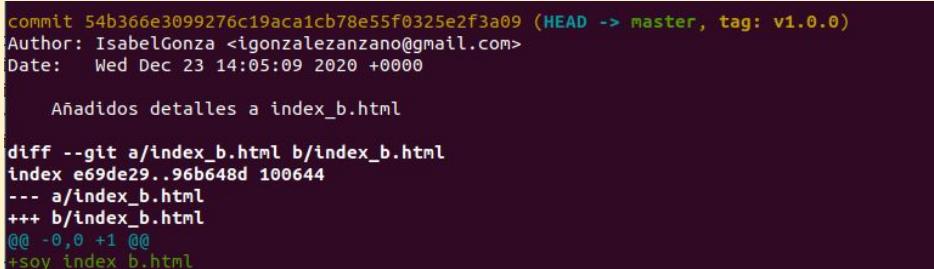
2.f) Tags

Taguea la versión v1.0.0 (busca información sobre cómo hacerlo si no está en apuntes)

Con el comando `git tag v1.0.0 54b366e` creamos el tag y se lo asignamos al último commit



lo visualizamos con `git show`

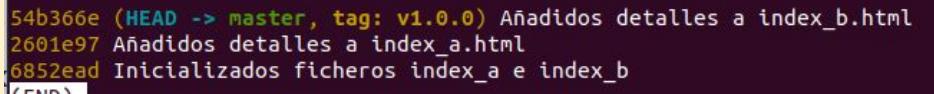


```
commit 54b366e3099276c19aca1cb78e55f0325e2f3a09 (HEAD -> master, tag: v1.0.0)
Author: IsabelGonza <igonzalezanzano@gmail.com>
Date:   Wed Dec 23 14:05:09 2020 +0000

    Añadidos detalles a index_b.html

diff --git a/index_b.html b/index_b.html
index e69de29..96b648d 100644
--- a/index_b.html
+++ b/index_b.html
@@ -0,0 +1 @@
+soy index b.html
```

Podemos usar también `git log`



```
54b366e (HEAD -> master, tag: v1.0.0) Añadidos detalles a index_b.html
2601e97 Añadidos detalles a index_a.html
6852ead Inicializados ficheros index_a e index_b
```

Comandos interesantes

git tag nombreTag - Crea un tag.
 git tag nombreTag identificadorCommit - añade el tag al commit concreto.
 git tag - lista las tags creadas.
 git tag -d nombreTag - Permite borrar un tag
 git show (nombreTag) - permite ver cómo estaba el repositorio en cada estado que has etiquetado anteriormente, o el del tag concreto

Información adicional

<https://desarrolloweb.com/articulos/especificar-versiones-git-tag.html>
<https://git-scm.com/book/es/v2/Fundamentos-de-Git-Etiquetado>

2.g) Archivo .gitignore

Ignora todos los ficheros de log. (extensión .log), pruébalo.

Hay que crear el archivo .gitignore, añadirlo a stage area y comitearlo:

```
sudo nano .gitignore
git add .gitignore
git commit -m "Añadido archivo .gitignore"
```

```
→ proyectoGit git:(master) ✘ git add .gitignore
→ proyectoGit git:(master) ✘ git commit -m "Añadido archivo .gitignore"
[master dc913ee] Añadido archivo .gitignore
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 .gitignore
```

Para que ignore todos los ficheros de extensión .log hay que añadir al .gitignore la siguiente información: sudo nano .gitignore

.log no aparecerán los archivos relacionados con los logs

prueba/ además he añadido un directorio para poder realizar pruebas sin que git lleve su seguimiento ni haya que preocuparse por realizar commit.

```
sudo nano .gitignore
.GNU nano 4.8
.log
.gitignore
prueba/
```

Añadimos el cambio: git add .gitignore

comiteamos el cambio: git commit -m "Añadiremos dos líneas de código para ignorar los archivos .log y el directorio /prueba .gitignore"

```
→ proyectoGit git:(master) ✘ git add .gitignore
→ proyectoGit git:(master) ✘ git commit -m "Modificado .gitignore para ignorar ficheros de log, se incluye tambien una carpeta de pruebas"
[master e1821b1] Modificado .gitignore para ignorar ficheros de log, se incluye tambien una carpeta de pruebas
 1 file changed, 2 insertions(+)
→ proyectoGit git:(master)
```

Comprobamos el commit

```
e1821b1 (HEAD -> master) Modificado .gitignore para ignorar ficheros de log, se incluye tambien una carpeta de pruebas
dc913ee Añadido archivo .gitignore
54b366e (tag: v1.0.0) Añadidos detalles a index_b.html
2601e97 Añadidos detalles a index_a.html
0852ead Inicializados ficheros index_a e index_b
```

Creamos la carpeta prueba con un fichero .html

```
→ proyectoGit git:(master) tree
.
├── index_a.html
├── index_b.html
└── prueba
    └── prueba.html
```

Con git status vemos si incluye esta nueva carpeta

```
→ proyectoGit git:(master) git status
On branch master
nothing to commit, working tree clean
```

2.h) Descartando cambios

Comprueba el estado de tu repo del apartado 2 y confirma que no tienes cambios pendientes. Añade cambios al fichero index_a.html y vuelve a comprobar el estado de

tu repo. Descarta los cambios del fichero index_a.html del directorio de trabajo y vuelve a comprobar el estado de tu repo.

git status

```
→ proyectoGit git:(master) git status
On branch master
nothing to commit, working tree clean
```

modifico index_a.html y git status

```
→ proyectoGit git:(master) sudo nano index_a.html
→ proyectoGit git:(master) X git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   index_a.html

no changes added to commit (use "git add" and/or "git commit -a")
```

git diff

```
diff --git a/index_a.html b/index_a.html
index 3b508bc..1fe543f 100644
--- a/index_a.html
+++ b/index_a.html
@@ -1 +1,2 @@
 soy index_a.html
+Añadida linea nueva
```

Descarto el cambio con `git checkout -- index_a.html`

```
→ proyectoGit git:(master) X git checkout -- index_a.html
```

Git status

```
→ proyectoGit git:(master) git status
On branch master
nothing to commit, working tree clean
```

git diff sale vacío ya que hemos eliminado la modificación.

```
sudo nano index_a.html
soy index_a.html
+Añadida linea nueva
```

Comandos interesantes

`git checkout -- nombreArchivo` - Descarta cambios en el archivo indicado

2.i) Cambios y commits

Añade cambios al fichero index_b.html. Añade el fichero index_b.html al área de preparación y comprueba el estado de tu repo. Saca el fichero index_b.html del área de preparación y comprueba el estado de tu repo. Añade más modificaciones al fichero index_b.html y con un solo comando confirma las modificaciones pendientes del fichero index_b.html con el mensaje “Añadidos cambios al fichero b”

Modifico index_b.html: `sudo nano index_b.html` y `git status`

Se añade al área de preparación: `git add index_b.html` y `git status`

```

→ proyectoGit git:(master) X sudo nano index_b.html
→ proyectoGit git:(master) X git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   index_b.html

no changes added to commit (use "git add" and/or "git commit -a")
→ proyectoGit git:(master) X git add index_b.html
→ proyectoGit git:(master) X git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   index_b.html

```

Sacar el fichero del área de preparación

- git reset HEAD index_b.html

```

→ proyectoGit git:(master) X git reset HEAD index_b.html
Unstaged changes after reset:
M      index_b.html
→ proyectoGit git:(master) X git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   index_b.html

no changes added to commit (use "git add" and/or "git commit -a")

```

Modifico otra vez el fichero y comparo diferencias con git diff

```

diff --git a/index_b.html b/index_b.html
index 96b648d..95773a4 100644
--- a/index_b.html
+++ b/index_b.html
@@ -1 +1,3 @@
 soy index_b.html
+Añadida nueva línea de código al archivo B.
+Tercera modificación en el archivo B.

```

Con un comando añadir todos los cambios (fusion git add + git commit -m)

git commit -am "Añadidos cambios al fichero b"

```

→ proyectoGit git:(master) X git commit -am "Añadidos cambios al fichero b"
[master b54b38a] Añadidos cambios al fichero b
 1 file changed, 2 insertions(+)
→ proyectoGit git:(master) git status
On branch master
nothing to commit, working tree clean

```

Comandos interesantes

git reset HEAD nombreFichero - saca el fichero del área de preparacion

git commit -am "comentario del commit" -Este comando es una fusión de "git add + git commit -m"

2.j) Sobreescibir un commit

Añade modificaciones al fichero index_a.html y añade al área de preparación el fichero index_a.html. Modifica el último commit (Añadidos cambios al fichero b) hasta el momento para que incluya también los cambios en index_a.html y que el mensaje sea “Añadidos cambios a los ficheros a y b”. Comprueba en el histórico que ha sobreescrito el commit.

Modifico index_a.html: `sudo nano index_a.html` y `git status`

Se añade al área de preparación: `git add index_a.html` y `git status`

```
→ proyectoGit git:(master) sudo nano index_a.html
→ proyectoGit git:(master) X git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   index_a.html

no changes added to commit (use "git add" and/or "git commit -a")
→ proyectoGit git:(master) X git add index_a.html
→ proyectoGit git:(master) X git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   index_a.html
```

`git log --oneline`

```
b54b38a (HEAD -> master) Añadidos cambios al fichero b
e1821b1 Modificado .gitignore para ignorar ficheros de log, se incluye tambien una carpeta de pruebas
dc913ee Añadido archivo .gitignore
54b366e (tag: v1.0.0) Añadidos detalles a index_b.html
2601e97 Añadidos detalles a index_a.html
6852ead Inicializados ficheros index_a e index_b
```

Comando para sobreescibir el último commit

`git commit --amend -m “Añadimos los cambios a los ficheros a y b”`

```
→ proyectoGit git:(master) X git commit --amend -m "Añadimos los cambios a los ficheros a y b"
[master 7a8474a] Añadimos los cambios a los ficheros a y b
  Date: Wed Dec 23 15:15:31 2020 +0000
  2 files changed, 4 insertions(+)
→ proyectoGit git:(master) git status
On branch master
nothing to commit, working tree clean
```

`git log --oneline`

```
7a8474a (HEAD -> master) Añadimos los cambios a los ficheros a y b
e1821b1 Modificado .gitignore para ignorar ficheros de log, se incluye tambien una carpeta de pruebas
dc913ee Añadido archivo .gitignore
54b366e (tag: v1.0.0) Añadidos detalles a index_b.html
2601e97 Añadidos detalles a index_a.html
6852ead Inicializados ficheros index_a e index_b
```

Se aprecia claramente que el último commit ha sido sobreescrito.

Comandos interesantes

`git commit --amend -m “comentario”` - Sobreescribe el último commit y añade la nueva información.

3) TRABAJANDO CON REPOSITORIOS REMOTOS Y RAMAS (0.5PTOS CADA APARTADO)

3.a) Creando el repositorio en Github

Crea un repositorio remoto de nombre “proyectoGit” en tu cuenta de Github desde la interfaz web de github.

Crear un repositorio en GitHub

The screenshot shows the GitHub interface for creating a new repository. In the top section, there's a form to enter the owner (IsabelGonza) and repository name (proyectoGit). Below this, the repository page for 'IsabelGonza / proyectoGit' is displayed. It includes navigation tabs for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. A 'Quick setup' section provides instructions for cloning the repository via HTTPS or SSH, and a command-line setup guide is also shown.

3.b) Conexión remoto-local conexión ssh

Añade el repositorio remoto en local y sube el proyecto "proyectoGit" a github conectandote por ssh

Inicio del repositorio remoto

```
git remote add origin git@github.com:IsabelGonza/proyectoGit.git
```

Compartir local con GitHub

```
git push -u origin master
```

```

→ proyectoGit git:(master) git remote add origin git@github.com:IsabelGonza/proyectoGit.git
→ proyectoGit git:(master) git push -u origin master
The authenticity of host 'github.com (140.82.121.3)' can't be established.
RSA key fingerprint is SHA256:Tfhbg6kXUpJWGL7E1GOcspROMTxzCARLViKw6ESSY8.
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added 'github.com,140.82.121.3' (RSA) to the list of known hosts.
Enumerating objects: 18, done.
Counting objects: 100% (18/18), done.
Delta compression using up to 4 threads
Compressing objects: 100% (14/14), done.
Writing objects: 100% (18/18), 1.87 KiB | 957.00 KiB/s, done.
Total 18 (delta 0), reused 0 (delta 0)
To github.com:IsabelGonza/proyectoGit.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
→ proyectoGit git:(master)

```

3.c) Uso de la Interfaz web

Modifica el archivo index_a.html desde la interfaz web de github y actualiza tu repositorio local a la última versión con ese último cambio.

Desde la interfaz web modificaremos una línea de código en index_a.html

Recuperar los archivos del repositorio remoto: `git fetch origin`

Unir la rama remota con rama actual en local: `git pull`

```
→ proyectoGit git:(master) git fetch origin
Warning: Permanently added the RSA host key for IP address '140.82.121.4' to the list of known hosts.
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 825 bytes | 825.00 KiB/s, done.
From github.com:IsabelGonza/proyectoGit
  7a8474a..98b1796 master      -> origin/master
→ proyectoGit git:(master) git pull
Updating 7a8474a..98b1796
Fast-forward
 index_a.html | 1 +
  1 file changed, 1 insertion(+)
→ proyectoGit git:(master) git status
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean
```

3.d) Fusión de ramas

Crea una nueva rama llamada "feature". Desde la interfaz de Github modifica el archivo `index_a.html`, confirma los cambios y fusiona la rama feature con la rama master. Comenta para qué se usa el concepto "pull request".

Crear rama feature `git branch feature`

verificar con `git branch`, que nos dice la rama en la que estamos:

```
→ proyectoGit git:(master) git branch feature
→ proyectoGit git:(master) git branch
```

```
* feature
 * master
```

Cambiar de rama `git checkout feature`

```
* feature
 * master
```

Modificar `index_a.html` y confirmar los cambios.

```
→ proyectoGit git:(feature) sudo nano index_a.html
[sudo] password for isa:
→ proyectoGit git:(feature) X git commit -am "Cambios en la rama feature a fichero index_a.html"
[feature e6d6cc8] Cambios en la rama feature a fichero index_a.html
 1 file changed, 1 insertion(+), 1 deletion(-)
```

Compartir repositorio local con el remoto: `git push --set-upstream origin feature`

```
→ proyectoGit git:(feature) git push --set-upstream origin feature
Total 0 (delta 0), reused 0 (delta 0)
remote:
remote: Create a pull request for 'feature' on GitHub by visiting:
remote:     https://github.com/IsabelGonza/proyectoGit/pull/new/feature
remote:
To github.com:IsabelGonza/proyectoGit.git
 * [new branch]      feature -> feature
Branch 'feature' set up to track remote branch 'feature' from 'origin'.
```

Fusionar la rama feature con master, cambiamos de rama a master y lanzamos git merge feature

```
→ proyectoGit git:(master) git merge feature
Updating 98b1796..e6d6cc8
Fast-forward
 index_a.html | 2 ++
 1 file changed, 1 insertion(+), 1 deletion(-)
```

Abrimos el fichero y vemos que está la línea que añadimos en la rama feature:

```
sudo nano index_a.html
index_a.html
GNU nano 4.8
Soy index_a.html
Añado una segunda linea al fichero A.
Modificación desde la interfaz web de github :3
Modificación desde local en la rama feature. :0
```

Pull request

Un pull request es una petición para integrar nuestras propuestas o cambios de código a un proyecto, que luego serán aceptadas o no.

Para probar a hacer un pull request, vamos a cambiar a la rama feature y a crear un nuevo archivo feature.html

```
→ proyectoGit git:(feature) sudo touch feature.html
[sudo] password for isa:
→ proyectoGit git:(feature) X sudo nano feature.html
→ proyectoGit git:(feature) X git status
On branch feature
Your branch is ahead of 'origin/feature' by 1 commit.
 (use "git push" to publish your local commits)

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    feature.html

nothing added to commit but untracked files present (use "git add" to track)
```

Enviamos la rama al repositorio

```
→ proyectoGit git:(feature) X git add --all
→ proyectoGit git:(feature) X git commit -m "creada pagina feature.html"
[feature 549b2d6] creada pagina feature.html
 1 file changed, 1 insertion(+)
 create mode 100644 feature.html
```

```
→ proyectoGit git:(feature) git push -u origin feature
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 397 bytes | 397.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To github.com:IsabelGonza/proyectoGit.git
 98b1796..549b2d6  feature -> feature
Branch 'feature' set up to track remote branch 'feature' from 'origin'.
```

Creando un pull request

The screenshot shows a GitHub repository page for 'IsabelGonza / proyectoGit'. At the top, there's a yellow banner indicating 'feature' had recent pushes 1 minute ago. Below the banner, there are buttons for 'Code', 'Issues', 'Pull requests', 'Actions', 'Projects', 'Wiki', 'Security', 'Insights', and 'Settings'. Under the 'Pull requests' section, it says 'master' branch, '2 branches', and '0 tags'. There are three commits listed:

- IsabelGonza Cambios en la rama feature a fichero index_a.html** (e6d6cc8, 37 minutes ago) - Modificado .gitignore para ignorar ficheros de log, se incluye tambie...
- index_a.html** (Cambios en la rama feature a fichero index_a.html, 37 minutes ago)
- index_b.html** (Añadimos los cambios a los ficheros a y b, 1 hour ago)

At the bottom, there's a note: 'Help people interested in this repository understand your project by adding a README.' with a 'Add a README' button.

Ahora nos sale un botón compare and pull request que antes no estaba.

The screenshot shows the 'Open a pull request' interface. It has fields for 'base: master' and 'compare: feature'. A green checkmark indicates 'Able to merge'. The main area contains a text editor with the title 'creada pagina feature.html' and a preview tab. To the right, there are sections for 'Reviewers', 'Assignees', 'Labels', 'Projects', 'Milestone', and 'Linked issues'. At the bottom, there's a note about GitHub Community Guidelines and a 'Create pull request' button.

Luego veremos este mensaje:

creada pagina feature.html #1

 Open | IsabelGonza wants to merge 1 commit into `master` from `feature` 

 Conversation 0  Commits 1  Checks 0  Files changed 1

 IsabelGonza commented now Owner  ...

No description provided.

 **creada pagina feature.html** 549b2d6

Add more commits by pushing to the `feature` branch on [IsabelGonza/proyectoGit](#).

 **Continuous integration has not been set up**
GitHub Actions and [several other apps](#) can be used to automatically catch bugs and enforce style.

 **This branch has no conflicts with the base branch**
Merging can be performed automatically.

Merge pull request  or view command line instructions.

 Write Preview            

Leave a comment

Si aceptamos los cambios presionamos merge pull request.

creada pagina feature.html #1

Merged IsabelGonza merged 1 commit into master from feature now

Conversation 0 Commits 1 Checks 0 Files changed 1

IsabelGonza commented 2 minutes ago
No description provided.

creada pagina feature.html 549b2d6

IsabelGonza merged commit 5fc241 into master now Revert

Pull request successfully merged and closed
You're all set—the feature branch can be safely deleted. Delete branch

Write Preview

Leave a comment

Video support! Upload MP4 and MOV file types. Attach files by dragging & dropping, selecting or pasting them.

Información útil

[https://styde.net/pull-request-en-github/#:~:text=Los%20pull%20request%20permiten%20novedades%C3%B3niguan%20a%20un%20proyecto.](https://styde.net/pull-request-en-github/#:~:text=Los%20pull%20request%20permiten%20novedades%C3%B3niguan%20a%20un%20proyecto)

4) NETBEANS Y GIT (0.3 PTOS CADA APARTADO)

4.a) Proyecto en Netbeans e inicializacion de Git

Crea un proyecto con Netbeans (o bien utiliza un proyecto que hayas desarrollado para dwes), inicializa el proyecto Git.

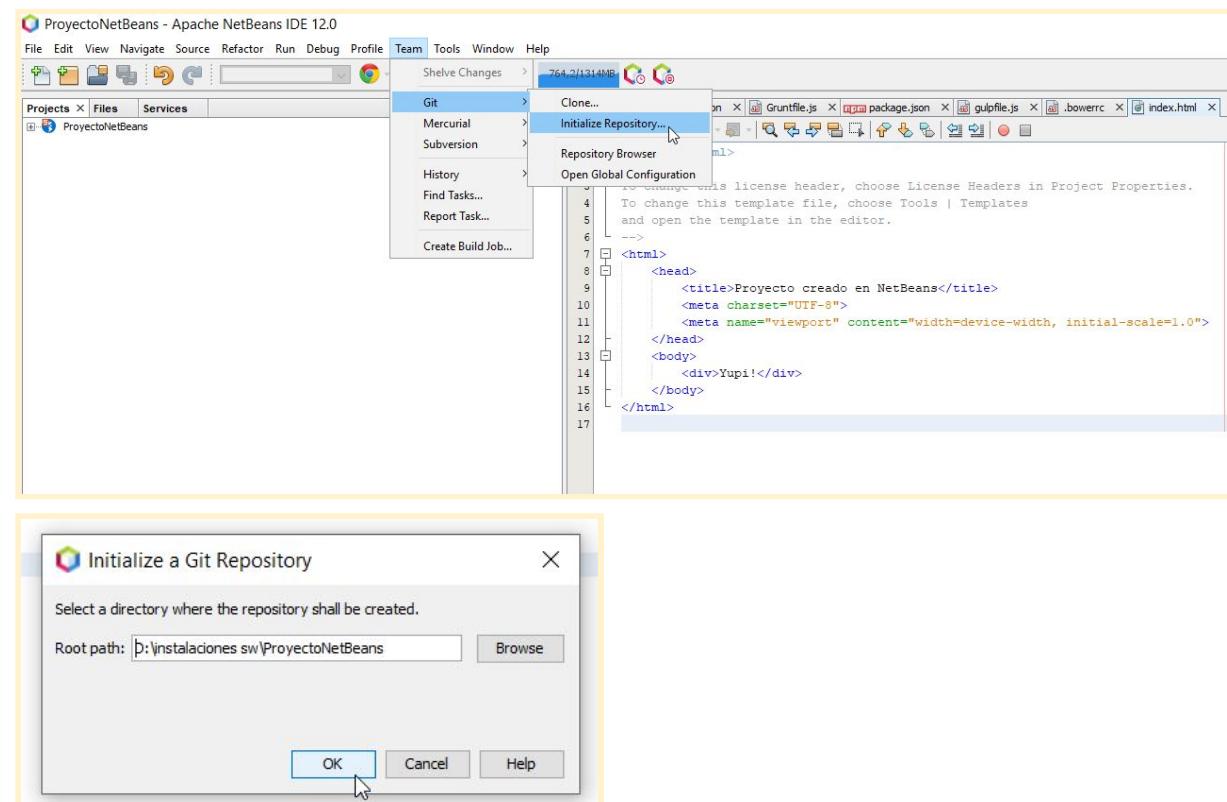
Ya que dispongo de una instalación del ide en mi máquina host, voy a realizar esta parte de la práctica desde ahí.

En el anexo están los pasos para añadir las claves ssh de windows a github.

1º Crear el repositorio en GitHub:



2º Crear el proyecto en NetBeans e iniciar el repositorio: Team=>Git=>Initialize Repository



4.b) Primer commit

Añade algún cambio a tu proyecto y confirma los cambios

He creado un nuevo archivo

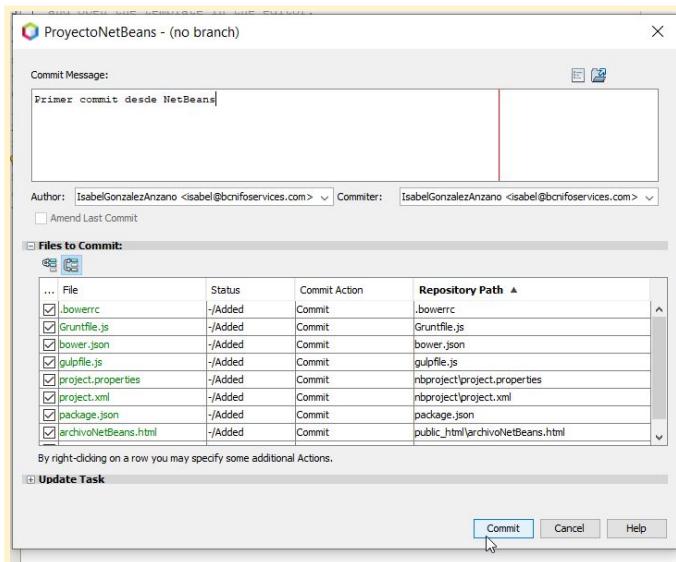
The screenshot shows the NetBeans IDE interface. On the left, the 'Projects' panel displays a project named 'ProyectoNetBeans' with a 'Site Root' node containing files like 'archivoNetBeans.html' and 'index.html'. On the right, the 'Start Page' editor shows the content of 'archivoNetBeans.html'. The code is as follows:

```

1 <!DOCTYPE html>
2 <!--
3 To change this license header, choose License Headers in Project Properties.
4 To change this template file, choose Tools | Templates
5 and open the template in the editor.
6 -->
7 <html>
8   <head>
9     <title>Hola</title>
10    <meta charset="UTF-8">
11    <meta name="viewport" content="width=device-width, initial-scale=1.0">
12  </head>
13  <body>
14    <div>Creado desde NetBeans</div>
15  </body>
16 </html>
17

```

Git => Commit

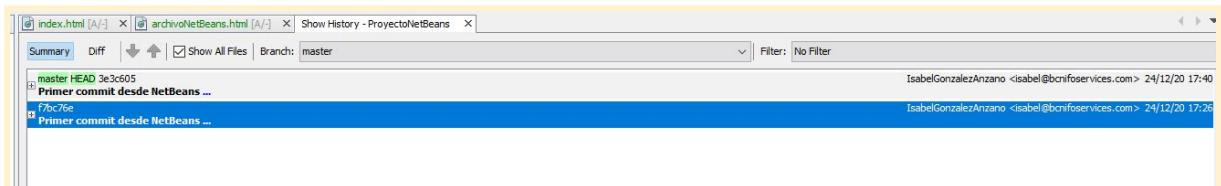


4.c) Comparando commits

Vuelve a modificar algún fichero y compáralo con la última versión commiteada.

He modificado el fichero index.html para ver los comits: Git => show History

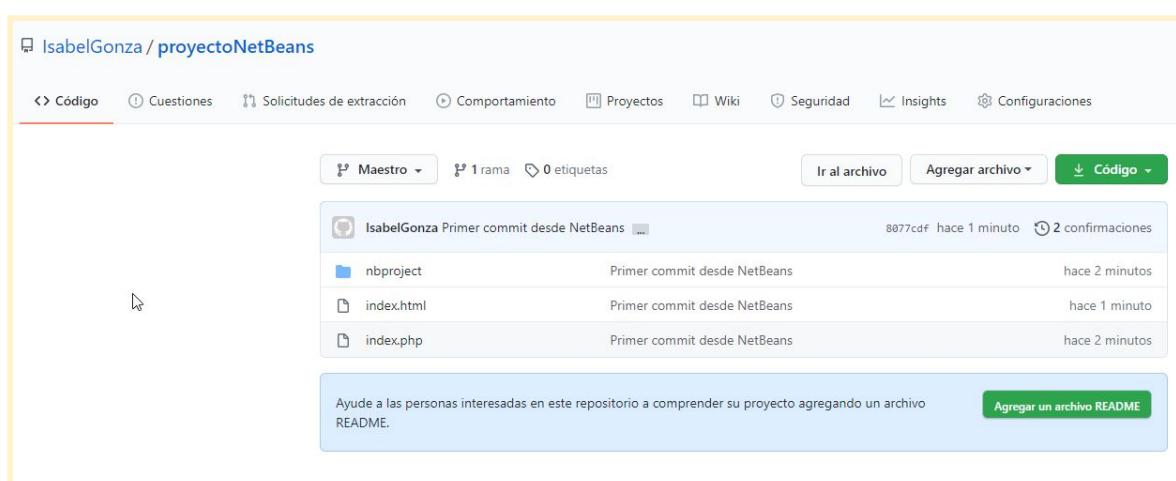
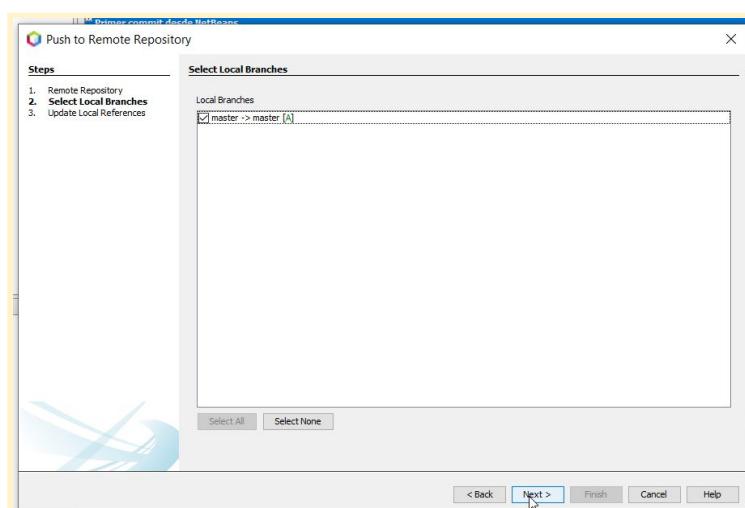
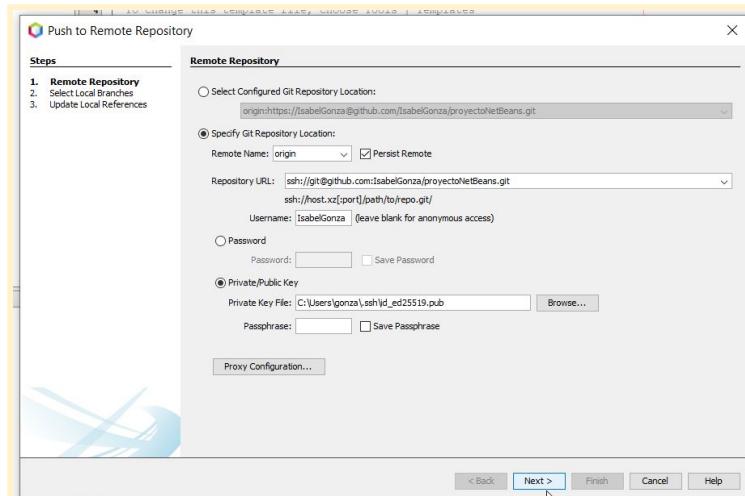
Aquí se pueden poner parámetros pero sino dándole al search ya sale el histórico:



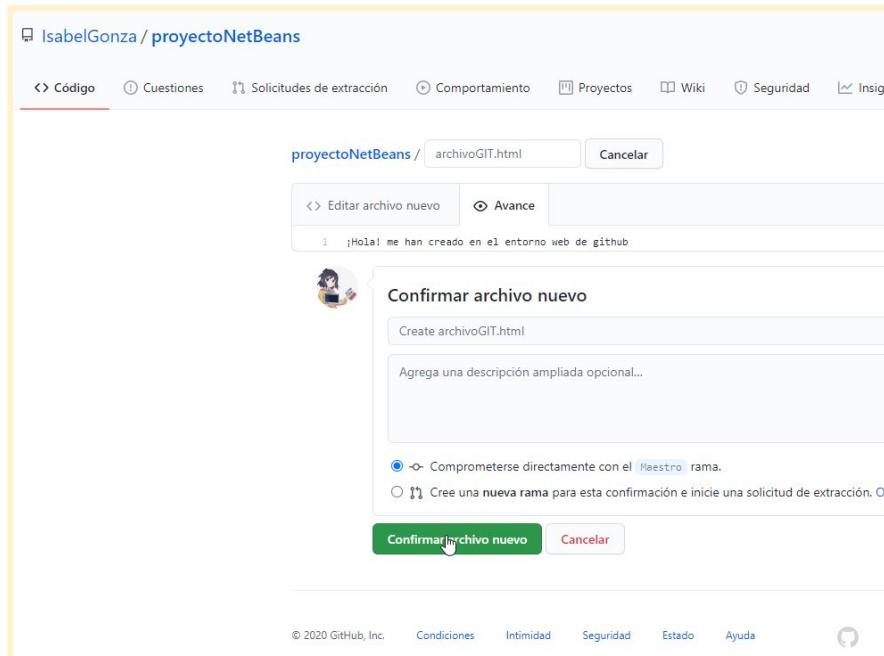
4.d) Subir proyecto a Repositorio por ssh

Sube tu proyecto a un repositorio en Github por ssh.

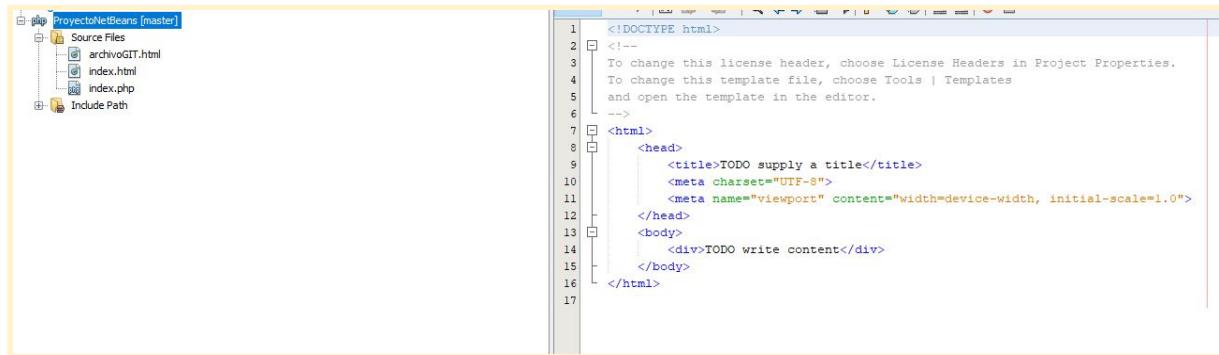
clic derecho sobre el proyecto => git => Remote => push para confirmar los cambios



4.e) Modifica el proyecto en Github y actualízalo en local



Ahora en el proyecto Git => remote => fech
y git => remote => pull



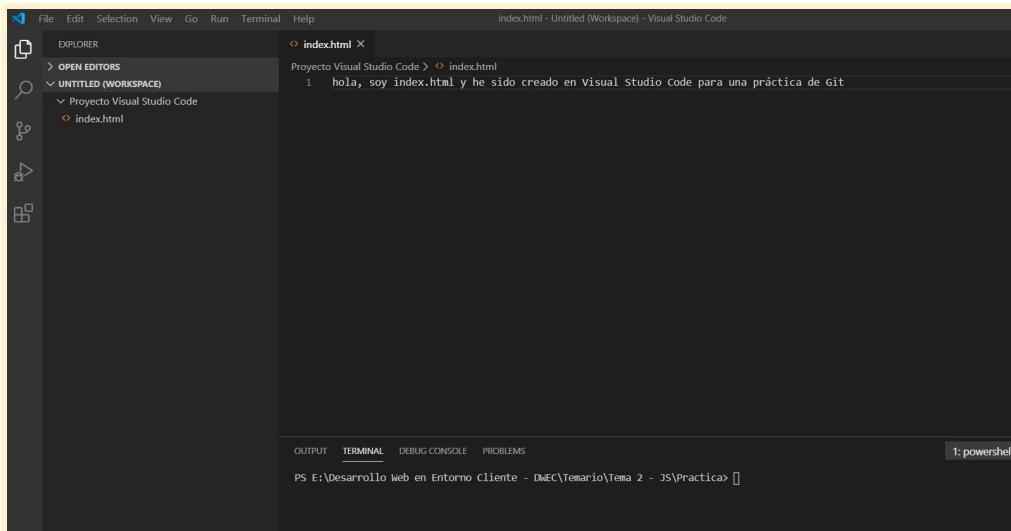
5) VISUAL STUDIO CODE (0.3 PTOS CADA APARTADO)

5.a) Proyecto en Visual Studio Code e inicialización de Git

Crea un proyecto con Visual Studio Code (o bien utiliza un proyecto que hayas desarrollado para dwec/diw), inicializa el proyecto Git

Ya que dispongo de una instalación del ide en mi máquina host, voy a realizar esta parte de la práctica desde ahí.

Creo un proyecto de 0 y le añado un index:



En la consola de comandos del ide:

```
git init
```

```
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS E:\Despliegue de Aplicaciones Web - DAW\02 - Clase\Proyecto Visual Studio Code> cd "e:\Despliegue de Aplicaciones Web - DAW\02 - Clase"
PS E:\Despliegue de Aplicaciones Web - DAW\02 - Clase> git init
Initialized empty Git repository in E:/Despliegue de Aplicaciones Web - DAW/02 - Clase/.git/
PS E:\Despliegue de Aplicaciones Web - DAW\02 - Clase> git add index.html
fatal: pathspec 'index.html' did not match any files
PS E:\Despliegue de Aplicaciones Web - DAW\02 - Clase> cd "..\Proyecto Visual Studio Code"
PS E:\Despliegue de Aplicaciones Web - DAW\02 - Clase\Proyecto Visual Studio Code> git add index.html
PS E:\Despliegue de Aplicaciones Web - DAW\02 - Clase\Proyecto Visual Studio Code> git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   index.html
```

Comiteamos e iniciamos el repositorio

```
PS E:\Despliegue de Aplicaciones Web - DAW\02 - Clase\Proyecto Visual Studio Code> git commit -m "Proyecto de VisualStudioCode"
[master (root-commit) 13f3670] Proyecto de VisualStudioCode
 1 file changed, 1 insertion(+)
 create mode 100644 Proyecto Visual Studio Code/index.html
PS E:\Despliegue de Aplicaciones Web - DAW\02 - Clase\Proyecto Visual Studio Code> git remote add origin https://github.com/IsabelGonza/VisualStudioCode
PS E:\Despliegue de Aplicaciones Web - DAW\02 - Clase\Proyecto Visual Studio Code>
```

5.b) Commit desde VS

Añade algún cambio a tu proyecto y confirma los cambios

Añado cambios:

```

EXPLORER OPEN EDITORS UNTITLED (WORKSPACE) Proyecto Visual Studio Code > index.html > html
index.html <!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Prueba VSC</title>
</head>
<body>
    holá, soy index.html y he sido creado en Visual Studio Code para una práctica de Git.
    <p>He añadido un cambio :)</p>
</body>
</html>

OUTPUT TERMINAL DEBUG CONSOLE PROBLEMS 2: powershell + ⌂

No commits yet

changes to be committed:
(use "git rm --cached <file>..." to unstage)
new file:   index.html

PS E:\Despliegue de Aplicaciones Web - DAW\02 - Clase\Proyecto Visual Studio Code> git commit -m "Proyecto de VisualStudioCode"
[master 13f3670] Proyecto de VisualStudioCode
 1 file changed, 1 insertion(+)
create mode 100640 Proyecto Visual Studio Code/index.html
PS E:\Despliegue de Aplicaciones Web - DAW\02 - Clase\Proyecto Visual Studio Code> git remote add origin https://github.com/IsabelGonza/VisualstudioCode.git
origin 2d2e185 [master]
PS E:\Despliegue de Aplicaciones Web - DAW\02 - Clase\Proyecto Visual Studio Code> git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
    (use "git restore <file>..." to discard changes in working directory)
      modified:   index.html

```

Commit:

```
git add index.html
git commit -m "Nueva línea de código, refactorizado a HTML"
```

```

PS E:\Despliegue de Aplicaciones Web - DAW\02 - Clase\Proyecto Visual Studio Code> git commit -m "Nueva línea de código, refactorizado a HTML"
[master 2d2e185] Nueva línea de código, refactorizado a HTML
 1 file changed, 15 insertions(+), 1 deletion(-)

```

Revisamos con git log:

```

PS E:\Despliegue de Aplicaciones Web - DAW\02 - Clase\Proyecto Visual Studio Code> git status
On branch master
nothing to commit, working tree clean
PS E:\Despliegue de Aplicaciones Web - DAW\02 - Clase\Proyecto Visual Studio Code> git log
commit 2d2e1850c3687dd9dda4d5a556ff64812f70f154 (HEAD -> master)
Author: IsabelGonzalezAnzano <isabel@bcnifoservices.com>
Date:   Thu Dec 24 16:32:42 2020 +0100

    Nueva línea de código, refactorizado a HTML

commit 13f3670e449ef262f20fc... (from 2d2e185)
Author: IsabelGonzalezAnzano <isabel@bcnifoservices.com>
Date:   Thu Dec 24 16:26:11 2020 +0100

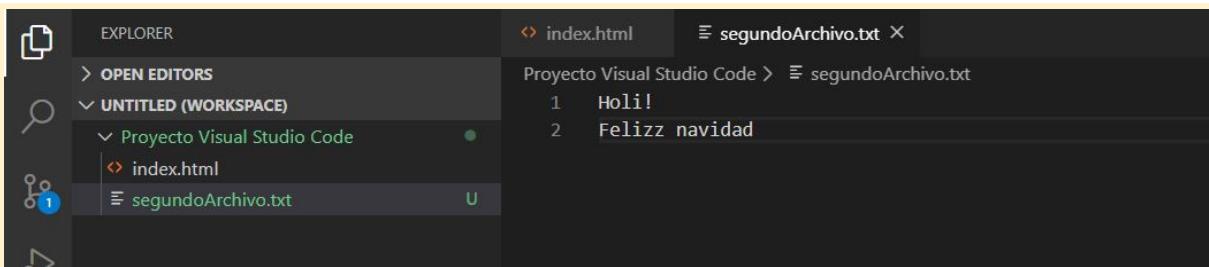
    Proyecto de VisualStudioCode

```

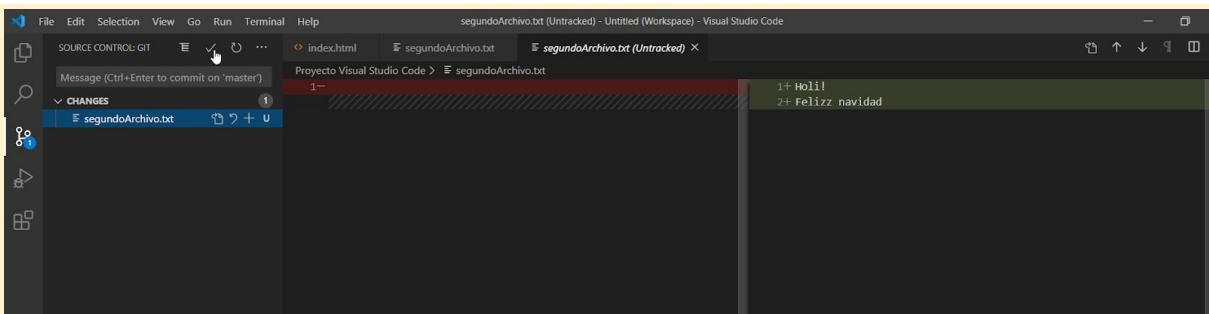
5.c) git log --oneline

Vuelve a modificar algún fichero y compáralo con la última versión commiteada.

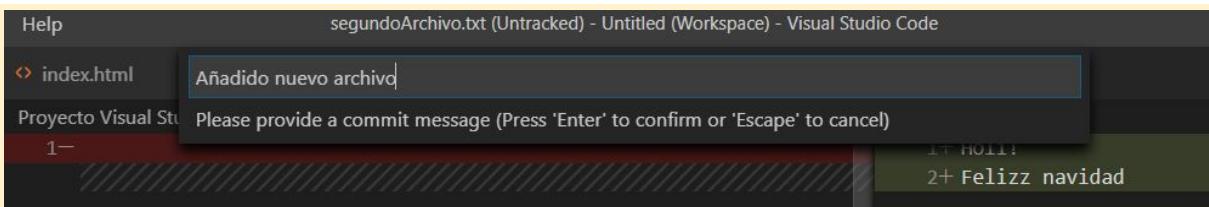
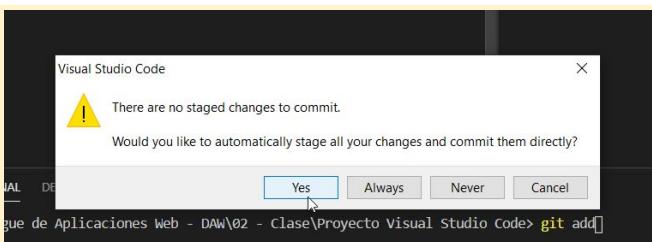
Creamos un nuevo archivo



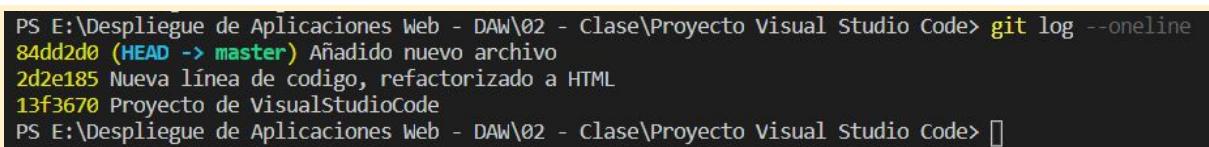
Vemos que a la izquierda se marca en la rama un nuevo cambio, lo seleccionamos y le damos al commit.



Como no lo habíamos pasado a la zona de preparación nos salta un aviso:



Comprobamos con `git log --oneline`

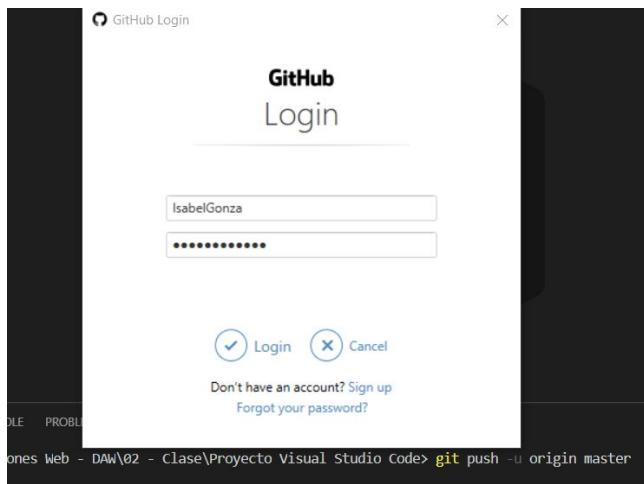


5.d) Subir proyecto por ssh - Push -

Sube tu proyecto a un repositorio en Github por ssh.

push a rama master

```
git push -u origin master
```



Nos pide loguearnos en github

```
PS E:\Despliegue de Aplicaciones Web - DAW\02 - Clase\Proyecto Visual Studio Code> git push -u origin master
Logon failed, use ctrl+c to cancel basic credential prompt.
Username for 'https://github.com': IsabelGonza
Password for 'https://IsabelGonza@github.com':
Enumerating objects: 12, done.
Counting objects: 100% (12/12), done.
Delta compression using up to 12 threads
Compressing objects: 100% (9/9), done.
Writing objects: 100% (12/12), 1.27 KiB | 186.00 KiB/s, done.
Total 12 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/IsabelGonza/visualStudioV1.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
```

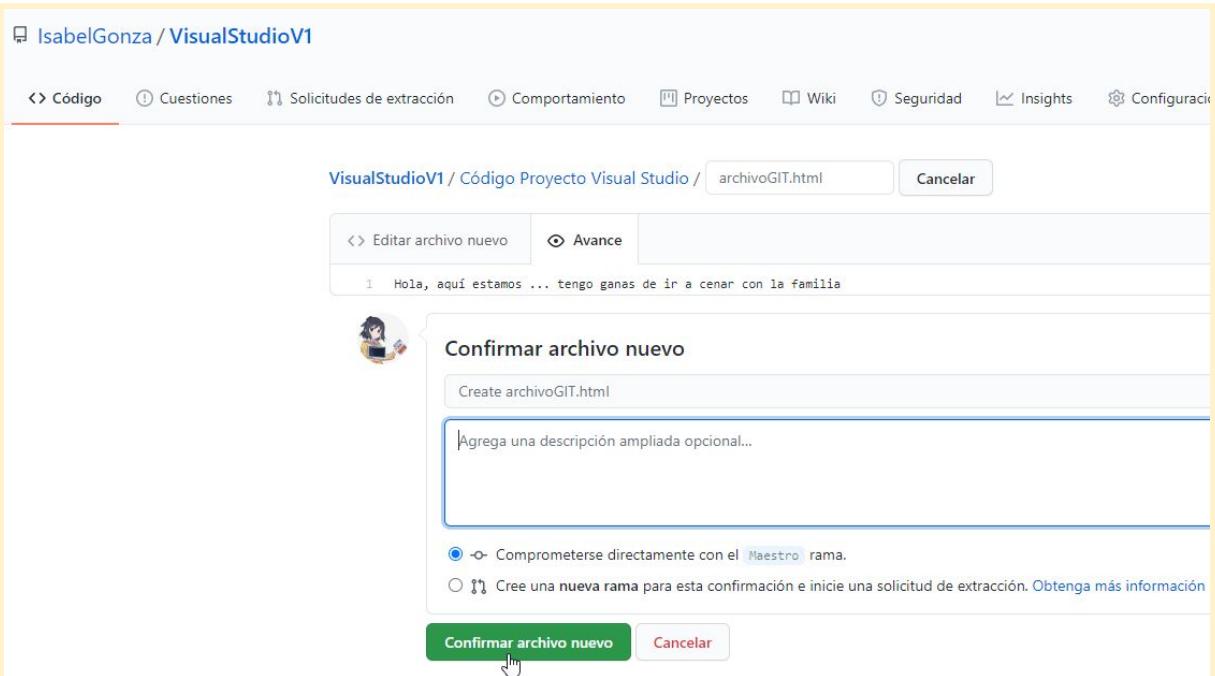
A screenshot of a GitHub repository page for 'IsabelGonza / VisualStudioV1'. The 'Código' tab is selected. The commit history shows a single commit by 'IsabelGonzalezAnzano' adding two new files: 'index.html' and 'segundoArchivo.txt'. The commit message is 'Añadido nuevo archivo'. The commit was made 'hace 10 minutos'.

File	Message	Time
index.html	Nueva linea de codigo. refactorizado a HTML	hace 19 minutos
segundoArchivo.txt	Añadido nuevo archivo	hace 10 minutos

5.e) Modificaciones en GitHub y descarga a local - PULL -

Modifica el proyecto en Github y actualízalo en local.

Añado un nuevo archivo desde Github:



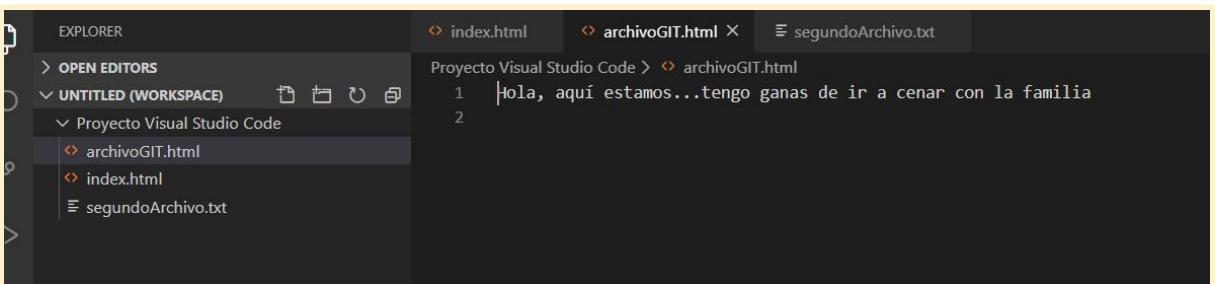
Desde la terminal de VisualStudio extraemos los cambios de remoto: `git fetch origin`

```
PS E:\Despliegue de Aplicaciones Web - DAW\02 - Clase\Proyecto Visual Studio Code> git fetch origin
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (4/4), 792 bytes | 1024 bytes/s, done.
From https://github.com/IsabelGonza/VisualStudioV1
  84dd2d0..0d957c0  master      -> origin/master
```

Unimos la rama remota con la rama local: `git pull`

```
PS E:\Despliegue de Aplicaciones Web - DAW\02 - Clase\Proyecto Visual Studio Code> git pull
Updating 84dd2d0..0d957c0
Fast-forward
 "C\303\263digo Proyecto Visual Studio/archivoGIT.html" | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 "C\303\263digo Proyecto Visual Studio/archivoGIT.html"
PS E:\Despliegue de Aplicaciones Web - DAW\02 - Clase\Proyecto Visual Studio Code>
```

Y ya nos aparece:

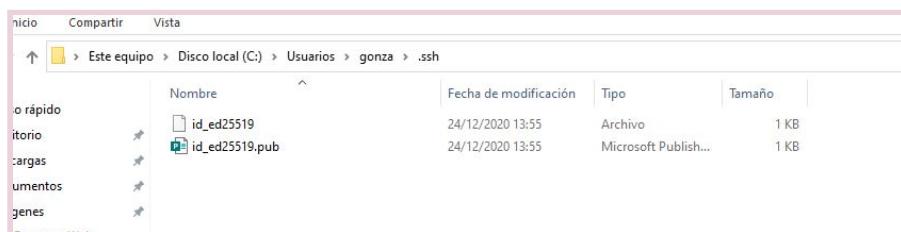


6) ANEXO

Crear claves ssh en windows

```
ssh-keygen -t ed25519 -C "igonzalezanzano@gmail.com"
```

```
gonza@Isabel-PC MINGW64 ~/Documents/claves ssh
$ ssh-keygen -t ed25519 -C "igonzalezanzano@gmail.com"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/c/Users/gonza/.ssh/id_ed25519):
Created directory '/c/Users/gonza/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/gonza/.ssh/id_ed25519
Your public key has been saved in /c/Users/gonza/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:0rcsUh87KeSOTqw9Z25+YsijhBb8cGERMHn40ow+1EY igonzalezanzano@gmail.com
The key's randomart image is:
+--[ED25519 256]--+
|   o+o.          |
|   o...           |
|   oo             |
```



```
gonza@Isabel-PC MINGW64 ~/Documents/claves ssh
$ eval `ssh-agent -s`
Agent pid 559

gonza@Isabel-PC MINGW64 ~/Documents/claves ssh
$ ssh-add ~/.ssh/id_ed25519
Identity added: /c/Users/gonza/.ssh/id_ed25519 (igonzalezanzano@gmail.com)
```

Asignarlas a la cuenta de github

Copiamos la clave :

```
gonza@Isabel-PC MINGW64 ~/Documents/claves ssh
$ clip < ~/.ssh/id_ed25519.pub
```



Información interesante

NetBeans & GIT

<https://hashbloeando.wordpress.com/2019/01/10/usando-git-en-netbeans/>

<https://javiergarciaescobedo.es/programacion-en-java/29-trucos/354-alajar-proyecto-netbeans-en-github>

VisualStudio Code & GIT

<https://lemoncode.net/lemoncode-blog/2017/12/12/git-y-visual-studio-code>