

Git 2023

Creado viernes 14 abril 2023

Preliminares:

1) Descargar e instalar Git:

<https://git-scm.com/downloads>

2) Ver la versión instalada:

```
git --version
```

3) Configuración:

Los siguientes comandos nos permiten realizar la configuración inicial de git:

```
git config --global user.name "ArielCodo"
git config --global user.email "ariel.palazzesi@bue.edu.ar"
git config --global core.editor "code --wait"
```

Para ver la configuración seteada, se puede introducir el comando sin el parámetro a configurar:

```
git config --global user.name
ArielCodo

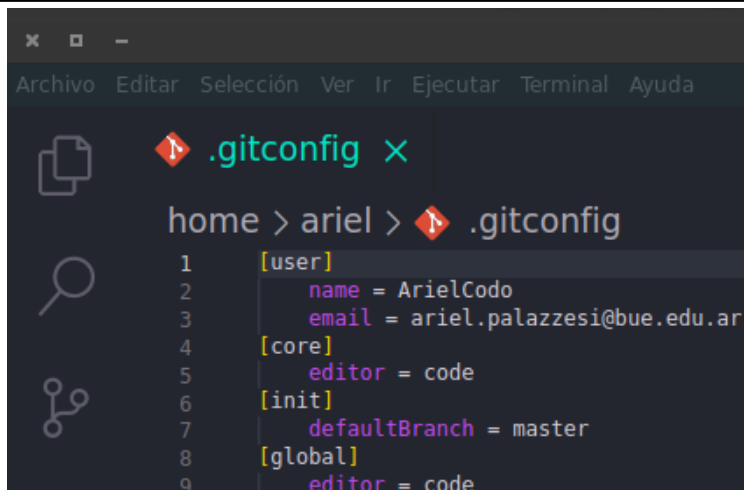
git config --global user.email
ariel.palazzesi@bue.edu.ar

git config --global core.editor "code --wait"
code

REM Ver toda la configuración:
git config --global --list
user.name=ArielCodo
user.email=ariel.palazzesi@bue.edu.ar
core.editor=code
init.defaultbranch=master
global.editor=code
```

Ver configuracion, la va a mostrar en VSCode por lo que configuramos antes:

```
git config --global -e
```



Ayuda de la opcion "config":

```
git config -h
```

Actividad 1: Trabajo local

1) Creamos una carpeta y nos movemos a ella:

```
mkdir git01  
cd git01
```

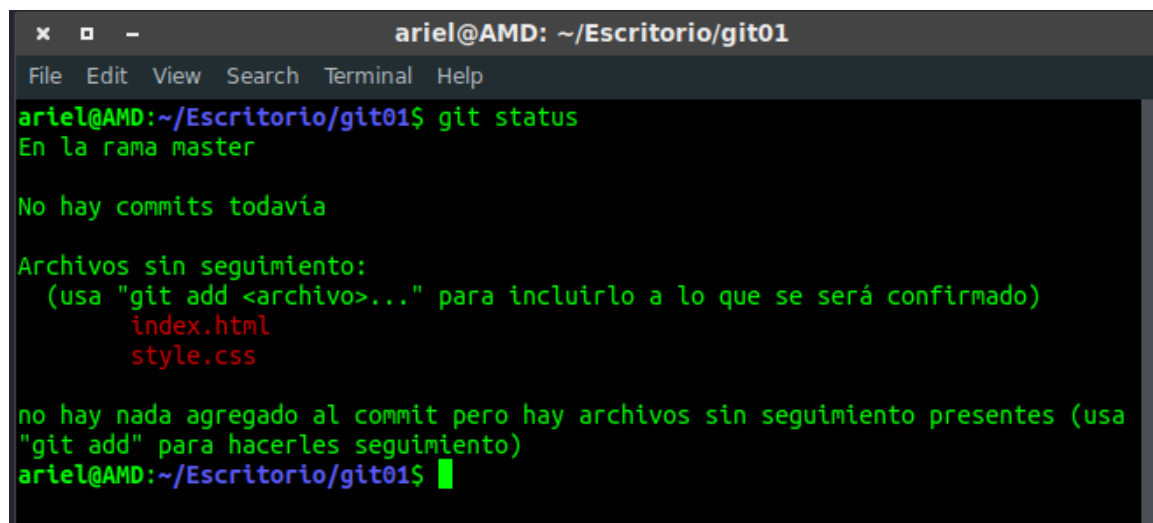
2) Inicializamos el repositorio:

```
git init
```

3) Creamos algunos archivos y vemos el estado del repositorio:

index.html
style.css

```
git status
```

A screenshot of a terminal window titled 'ariel@AMD: ~/Escritorio/git01'. The terminal shows the output of the 'git status' command. The output indicates that the user is on the 'master' branch and there are no commits yet. It lists two files, 'index.html' and 'style.css', as 'Archivos sin seguimiento' (untracked files). A message explains that these files are not yet added to the commit and suggests using 'git add' to track them. The prompt 'ariel@AMD:~/Escritorio/git01\$' is visible at the bottom.

```
ariel@AMD:~/Escritorio/git01$ git status  
En la rama master  
  
No hay commits todavía  
  
Archivos sin seguimiento:  
  (usa "git add <archivo>..." para incluirlo a lo que se será confirmado)  
    index.html  
    style.css  
  
no hay nada agregado al commit pero hay archivos sin seguimiento presentes (usa  
"git add" para hacerles seguimiento)  
ariel@AMD:~/Escritorio/git01$
```

Podemos agregar ambos archivos al staging area con add y ver nuevamente el estado:

```
git add index.html  
git add style.css
```

```
ariel@AMD: ~/Escritorio/git01
File Edit View Search Terminal Help

ariel@AMD:~/Escritorio/git01$ git add index.html
ariel@AMD:~/Escritorio/git01$ git add style.css
ariel@AMD:~/Escritorio/git01$ git status
En la rama master

No hay commits todavía

Cambios a ser confirmados:
(usa "git rm --cached <archivo>..." para sacar del área de stage)
nuevos archivos: index.html
nuevos archivos: style.css

ariel@AMD:~/Escritorio/git01$
```

Hay dos archivos listos para ser commiteados. Lo hacemos con git commit:

```
git commit -m "Primer commit - Estado inicial"
```

```
ariel@AMD: ~/Escritorio/git01
File Edit View Search Terminal Help

ariel@AMD:~/Escritorio/git01$ git commit -m "Primer commit - Estado inicial"
[master (commit-raiz) a6b1826] Primer commit - Estado inicial
2 files changed, 15 insertions(+)
create mode 100644 index.html
create mode 100644 style.css
ariel@AMD:~/Escritorio/git01$
```

Un nuevo git status no muestra cambios ni archivos en el staging area:

```
git status
```

```
ariel@AMD: ~/Escritorio/git01
File Edit View Search Terminal Help

ariel@AMD:~/Escritorio/git01$ git status
En la rama master
nada para hacer commit, el árbol de trabajo está limpio
ariel@AMD:~/Escritorio/git01$
```

Modificamos los archivos de trabajo, vemos el estado, agregamos con git add * y hacemos un nuevo commit:

```
git status
git add *
git commit -m "Segundo commit - H1 y P, mas comentario"
```

```
ariel@AMD: ~/Escritorio/git01
File Edit View Search Terminal Help

ariel@AMD:~/Escritorio/git01$ git status
En la rama master
Cambios no rastreados para el commit:
  (usa "git add <archivo>..." para actualizar lo que será confirmado)
  (usa "git restore <archivo>..." para descartar los cambios en el directorio de
trabajo)
    modificados:    index.html
    modificados:    style.css

sin cambios agregados al commit (usa "git add" y/o "git commit -a")
ariel@AMD:~/Escritorio/git01$ git add *
ariel@AMD:~/Escritorio/git01$ git commit -m "Segundo commit - H1 y P, mas coment
ario"
[master b5c077a] Segundo commit - H1 y P, mas comentario
 2 files changed, 3 insertions(+), 1 deletion(-)
ariel@AMD:~/Escritorio/git01$
```

Podemos usar git log para ver los commits realizados, su hash, etc:

```
git log
```

```
ariel@AMD: ~/Escritorio/git01
File Edit View Search Terminal Help

ariel@AMD:~/Escritorio/git01$ git log
commit b5c077a0e87a9dfc76624ac496c6d64b6d676264 (HEAD -> master)
Author: ArielCodo <ariel.palazzesi@bue.edu.ar>
Date:   Fri Apr 14 10:26:10 2023 -0300

    Segundo commit - H1 y P, mas comentario

commit a6b1826c6dff9d1ca7d21a1d2f0865f8975f2a6a
Author: ArielCodo <ariel.palazzesi@bue.edu.ar>
Date:   Fri Apr 14 10:21:50 2023 -0300

    Primer commit - Estado inicial
ariel@AMD:~/Escritorio/git01$
```

3) Creamos una nueva rama llamada "cambios":

```
git branch cambios
```

Crea la rama, pero no nos mueve a ella:

```
ariel@AMD: ~/Escritorio/git01
File Edit View Search Terminal Help

ariel@AMD:~/Escritorio/git01$ git branch cambios
ariel@AMD:~/Escritorio/git01$ git log
commit b5c077a0e87a9dfc76624ac496c6d64b6d676264 (HEAD -> master, cambios)
Author: ArielCodo <ariel.palazzesi@bue.edu.ar>
Date:   Fri Apr 14 10:26:10 2023 -0300

    Segundo commit - H1 y P, mas comentario

commit a6b1826c6dff9d1ca7d21a1d2f0865f8975f2a6a
Author: ArielCodo <ariel.palazzesi@bue.edu.ar>
Date:   Fri Apr 14 10:21:50 2023 -0300

    Primer commit - Estado inicial
ariel@AMD:~/Escritorio/git01$ git status
En la rama master
nada para hacer commit, el árbol de trabajo está limpio
ariel@AMD:~/Escritorio/git01$
```

4) Pasamos a la nueva rama:

```
git checkout cambios
```

```
ariel@AMD: ~/Escritorio/git01
File Edit View Search Terminal Help

ariel@AMD:~/Escritorio/git01$ git checkout cambios
Cambiado a rama 'cambios'
ariel@AMD:~/Escritorio/git01$
```

5) Modificamos los archivos de trabajo, vemos el estado, agregamos con git add * y hacemos un nuevo commit:

```
git status
git add *
git commit -m "Primer commit en la rama cambios"
```

```
ariel@AMD: ~/Escritorio/git01
File Edit View Search Terminal Help

ariel@AMD:~/Escritorio/git01$ git status
En la rama cambios
Cambios no rastreados para el commit:
  (usa "git add <archivo>..." para actualizar lo que será confirmado)
  (usa "git restore <archivo>..." para descartar los cambios en el directorio de
trabajo)
    modificados:   index.html
    modificados:   style.css

sin cambios agregados al commit (usa "git add" y/o "git commit -a")
ariel@AMD:~/Escritorio/git01$ git add *
ariel@AMD:~/Escritorio/git01$ git commit -m "Primer commit en la rama cambios"
ariel@AMD:~/Escritorio/git01$
```

Hacemos un git log:

```
git log
```

```
ariel@AMD: ~/Escritorio/git01
File Edit View Search Terminal Help

ariel@AMD:~/Escritorio/git01$ git commit -m "Primer commit en la rama cambios"
[cambios 9db0b1b] Primer commit en la rama cambios
2 files changed, 5 insertions(+), 1 deletion(-)
ariel@AMD:~/Escritorio/git01$ git log
commit 9db0b1bcf725995c011e7fa2b81433be16887070 (HEAD -> cambios)
Author: ArielCodo <ariel.palazzesi@bue.edu.ar>
Date: Fri Apr 14 10:39:20 2023 -0300

    Primer commit en la rama cambios

commit b5c077a0e87a9dfc76624ac496c6d64b6d676264 (master)
Author: ArielCodo <ariel.palazzesi@bue.edu.ar>
Date: Fri Apr 14 10:26:10 2023 -0300

    Segundo commit - H1 y P, mas comentario

commit a6b1826c6dff9d1ca7d21a1d2f0865f8975f2a6a
Author: ArielCodo <ariel.palazzesi@bue.edu.ar>
Date: Fri Apr 14 10:21:50 2023 -0300

    Primer commit - Estado inicial
ariel@AMD:~/Escritorio/git01$
```

Vemos el HEAD que indica donde está el "extremo" de la rama **master** y de la rama **cambios**. Estamos en la rama **cambios**.

Volvamos a la rama master y hacemos un git log:

```
git checkout master
git log
```

```
ariel@AMD: ~/Escritorio/git01
File Edit View Search Terminal Help

ariel@AMD:~/Escritorio/git01$ git checkout master
Cambiado a rama 'master'
ariel@AMD:~/Escritorio/git01$ git log
commit b5c077a0e87a9dfc76624ac496c6d64b6d676264 (HEAD -> master)
Author: ArielCodo <ariel.palazzesi@bue.edu.ar>
Date: Fri Apr 14 10:26:10 2023 -0300

    Segundo commit - H1 y P, mas comentario

commit a6b1826c6dff9d1ca7d21a1d2f0865f8975f2a6a
Author: ArielCodo <ariel.palazzesi@bue.edu.ar>
Date: Fri Apr 14 10:21:50 2023 -0300

    Primer commit - Estado inicial
ariel@AMD:~/Escritorio/git01$
```

Vemos el HEAD que indica donde está el "extremo" de la rama **master** y de la rama **cambios**. Estamos en la rama **master**.

Si miramos los archivos, los cambios que se hicieron mientras estábamos en la rama **cambios** no se ven. Esto es porque han tenido lugar en otra rama de nuestro proyecto, y ahora estamos nuevamente en **master**.

Modificamos los archivos de trabajo nuevamente, ahora en la rama master, vemos el estado, agregamos con git add * y hacemos un nuevo commit:

```
git status
git add *
git commit -m "Tercer commit en la rama master"
```

```
ariel@AMD: ~/Escritorio/git01
File Edit View Search Terminal Help

ariel@AMD:~/Escritorio/git01$ git status
En la rama master
Cambios no rastreados para el commit:
  (usa "git add <archivo>..." para actualizar lo que será confirmado)
  (usa "git restore <archivo>..." para descartar los cambios en el directorio de
trabajo)
    modificados:    index.html
    modificados:    style.css

sin cambios agregados al commit (usa "git add" y/o "git commit -a")
ariel@AMD:~/Escritorio/git01$ git add *
ariel@AMD:~/Escritorio/git01$ git commit -m "Tercer commit en la rama master"
[master 291d9f2] Tercer commit en la rama master
 2 files changed, 5 insertions(+), 1 deletion(-)
ariel@AMD:~/Escritorio/git01$
```

En este punto tenemos dos ramas de nuestro proyecto, **master** y **cambios**. Podemos cambiar de una a otra con **git checkout <rama>** y ver como son independientes una de la otra. Podemos avazar con cambios en ambas, agregando archivos, haciendo commits, etc.

Cada vez que pasamos de una rama a la otra, en la carpeta del proyecto solo tenemos acceso a los cambios que correspondan a esa rama.

6) Fusionando ambas ramas:

Estando en master, fusionamos con ella la rama cambios:

```
git merge cambios
```

En este punto, pueden surgir conflictos o colisiones entre ambas versiones de uno o mas archivos.

```
ariel@AMD: ~/Escritorio/git01
File Edit View Search Terminal Help

ariel@AMD:~/Escritorio/git01$ git merge cambios
Auto-fusionando index.html
CONFLICTO (contenido): Conflicto de fusión en index.html
Auto-fusionando style.css
CONFLICTO (contenido): Conflicto de fusión en style.css
Fusión automática falló; arregle los conflictos y luego realice un commit con el
resultado.
```

Como hemos configurado a VSCode como editor, veremos ahí los conflictos a solucionar:

```
index.html - git01 - Visual Studio Code
Archivo  Editar  Selección  Ver  Ir  Ejecutar  Terminal  Ayuda

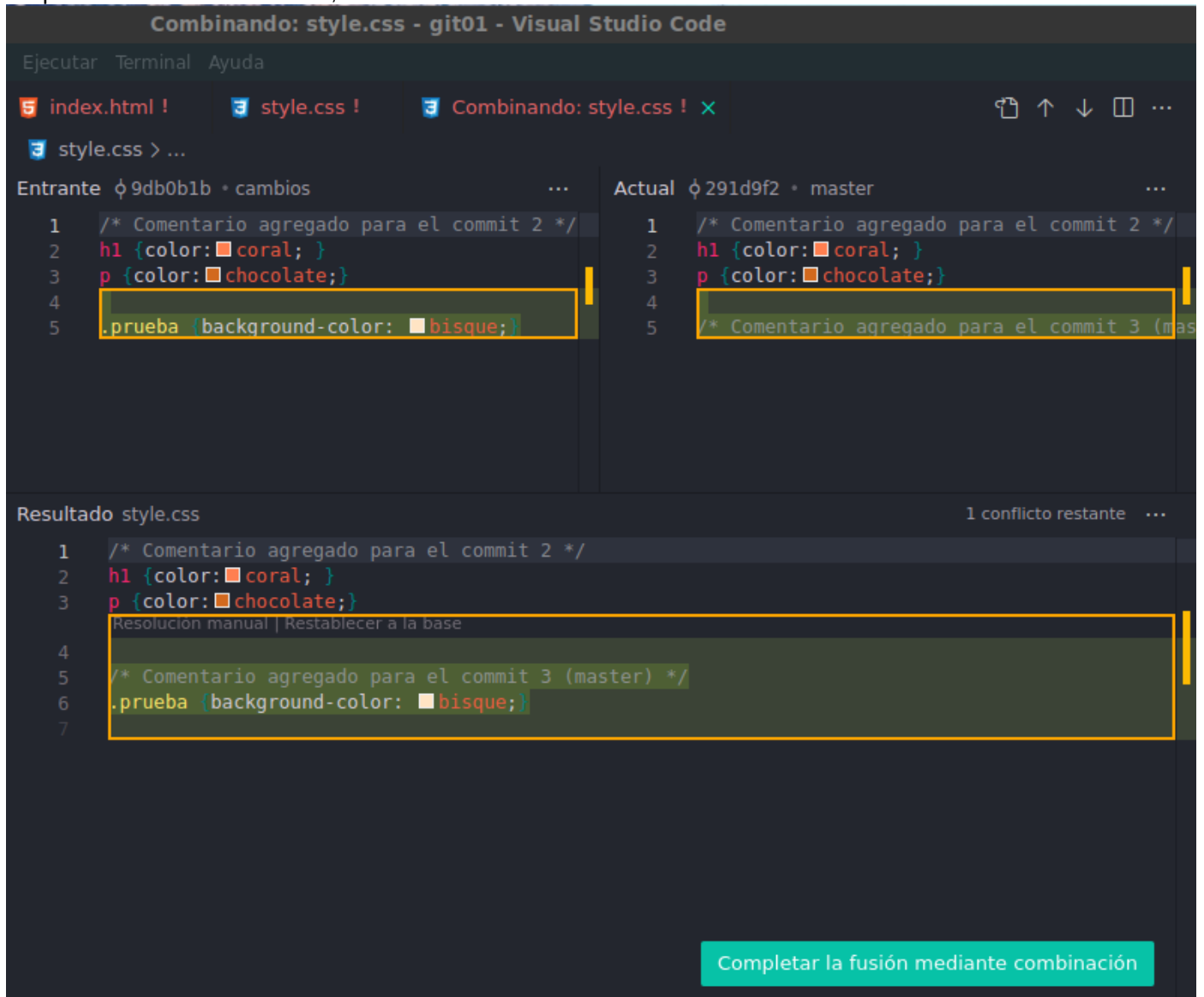
EXPLORADOR
  GIT01
    index.html !
    style.css 4, !

index.html > html > body > ?
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <meta http-equiv="X-UA-Compatible" content="IE=edge">
6    <meta name="viewport" content="width=device-width, initial-sc
7    <title>Pruebas con GIT</title>
8    <link rel="stylesheet" href="style.css">
9  </head>
10 <body>
11   <h1>Git y GitHub</h1>
12   <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. 0
13
14 <<<<<< HEAD (Cambio actual)
15   <p>Este párrafo se agregó en el tercer commit de la rama mast
16 =====
17   <p>Esto está en la rama cambios.</p>
18 >>>>>> cambios (Cambio entrante)
19 </body>
20 </html>
```

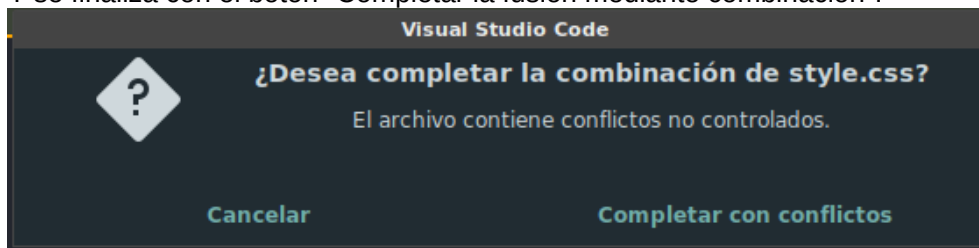
Acceptar cambio actual | Aceptar cambio entrante | Aceptar ambos cambios | Comparar...

Resolver en el Editor de combinación

Se pueden resolver ahí mismo, o ver en el "Editor de combinación":



Y se finaliza con el botón "Completar la fusión mediante combinación":



Un git log muestra que estamos en la rama master, con los archivos fusionados (se los puede ver en el editor de textos), y todos los commits realizados hasta el momento:

```
git log
```

```
ariel@AMD: ~/Escritorio/git01
File Edit View Search Terminal Help

commit c8f63f2794c72a0640455301672e0963572ea025 (HEAD -> master)
Merge: 291d9f2 9db0b1b
Author: ArielCodo <ariel.palazzesi@bue.edu.ar>
Date: Fri Apr 14 11:00:33 2023 -0300

    Merge branch 'cambios'

commit 291d9f230087c9e1e660b5b6cd17b29cd3f21b55
Author: ArielCodo <ariel.palazzesi@bue.edu.ar>
Date: Fri Apr 14 10:48:08 2023 -0300

    Tercer commit en la rama master

commit 9db0b1bcf725995c011e7fa2b81433be16887070 (cambios)
Author: ArielCodo <ariel.palazzesi@bue.edu.ar>
Date: Fri Apr 14 10:39:20 2023 -0300

    Primer commit en la rama cambios

commit b5c077a0e87a9dfc76624ac496c6d64b6d676264
Author: ArielCodo <ariel.palazzesi@bue.edu.ar>
Date: Fri Apr 14 10:26:10 2023 -0300

    Segundo commit - H1 y P, mas comentario

commit a6b1826c6dff9d1ca7d21a1d2f0865f8975f2a6a
Author: ArielCodo <ariel.palazzesi@bue.edu.ar>
Date: Fri Apr 14 10:21:50 2023 -0300

    Primer commit - Estado inicial
```

Nota: se puede usar git log --oneline para ver logs mas compactos.

```
ariel@AMD: ~/Escritorio/git01
File Edit View Search Terminal Help

ariel@AMD:~/Escritorio/git01$ git log --oneline
c8f63f2 (HEAD -> master) Merge branch 'cambios'
291d9f2 Tercer commit en la rama master
9db0b1b (cambios) Primer commit en la rama cambios
b5c077a Segundo commit - H1 y P, mas comentario
a6b1826 Primer commit - Estado inicial
ariel@AMD:~/Escritorio/git01$
```

7) Visualizar un estado anterior:

Nota: se puede usar git log --oneline para ver logs mas compactos.

Para ver a un estado cualquiera, usamos su hash **y un punto final**:

```
git checkout a6b1826c6dff9d1ca7d21a1d2f0865f8975f2a6a .
```

Los archivos vuelven a la versión indicada:

```
ariel@AMD: ~/Escritorio/git01
File Edit View Search Terminal Help

ariel@AMD:~/Escritorio/git01$ git checkout a6b1826c6dff9d1ca7d21a1d2f0865f8975f2a6a .
Actualizadas 2 rutas desde e12e787
```

Puedes consultar y editar archivos, compilar el proyecto y realizar pruebas sin preocuparte de perder el estado actual del proyecto. Nada de lo que hagas aquí se guardará en tu repositorio. Para continuar con el desarrollo, debes volver al estado “actual” del proyecto:

```
git checkout master
```

Y los archivos vuelven a estar como en la versión más reciente.

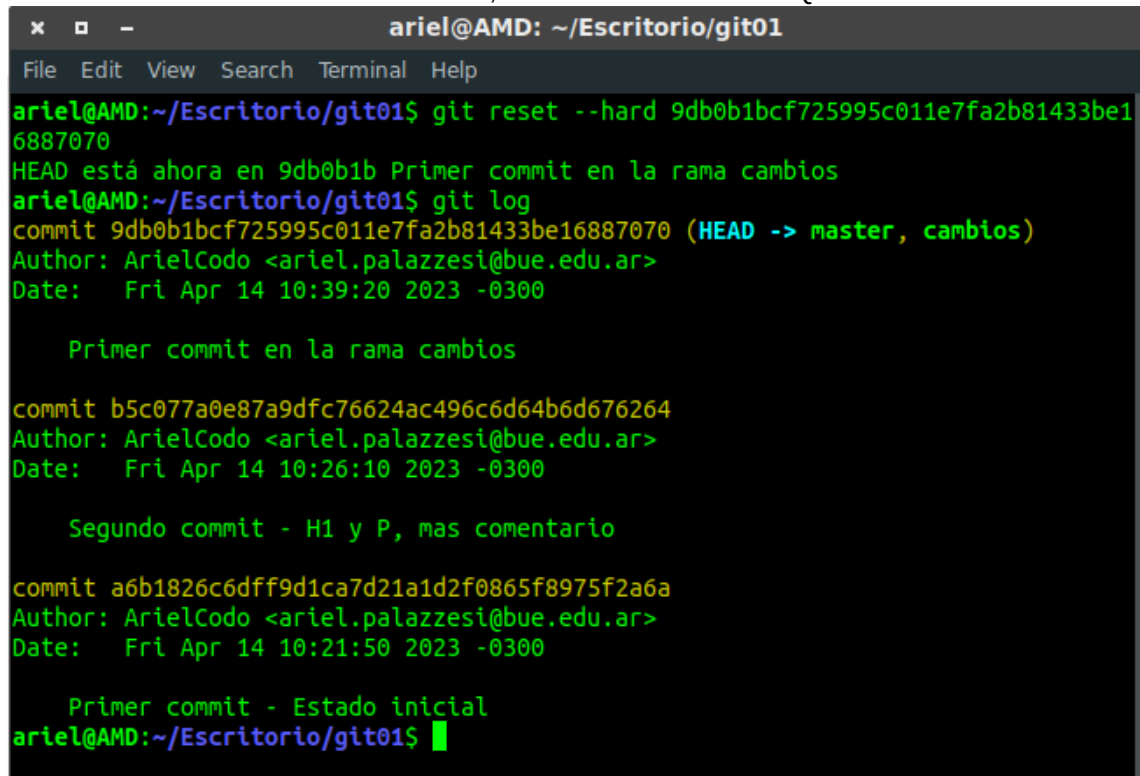
8) Deshacer cambios:

Mas info: <https://www.atlassian.com/es/git/tutorials/undoing-changes/git-reset>

git reset: El comando **git reset HEAD~2** retrocede la rama actual dos confirmaciones, con lo que se elimina de forma eficaz las dos instantáneas que acabamos de crear a partir del historial del proyecto. Recuerda que este tipo de restablecimiento solo debe utilizarse en las confirmaciones no publicadas. No hagas nunca la operación anterior si ya has enviado tus confirmaciones a un repositorio compartido.

```
git reset --hard 9db0b1bcf725995c011e7fa2b81433be16887070
```

SE VUELVE A UN PUNTO ANTERIOR, PERO SE PIERDE LO QUE HAYA HACIA ADELANTE.



The screenshot shows a terminal window titled 'ariel@AMD: ~/Escritorio/git01'. The user runs 'git reset --hard 9db0b1bcf725995c011e7fa2b81433be16887070'. The output indicates that HEAD is now at 9db0b1b and that the first commit in the 'cambios' branch is being reset. Then, the user runs 'git log', which shows the commit history, including the commit that was reset. The terminal output is as follows:

```
ariel@AMD:~/Escritorio/git01$ git reset --hard 9db0b1bcf725995c011e7fa2b81433be16887070
HEAD está ahora en 9db0b1b Primer commit en la rama cambios
ariel@AMD:~/Escritorio/git01$ git log
commit 9db0b1bcf725995c011e7fa2b81433be16887070 (HEAD -> master, cambios)
Author: ArielCodo <ariel.palazzesi@bue.edu.ar>
Date:   Fri Apr 14 10:39:20 2023 -0300

    Primer commit en la rama cambios

commit b5c077a0e87a9dfc76624ac496c6d64b6d676264
Author: ArielCodo <ariel.palazzesi@bue.edu.ar>
Date:   Fri Apr 14 10:26:10 2023 -0300

    Segundo commit - H1 y P, mas comentario

commit a6b1826c6dff9d1ca7d21a1d2f0865f8975f2a6a
Author: ArielCodo <ariel.palazzesi@bue.edu.ar>
Date:   Fri Apr 14 10:21:50 2023 -0300

    Primer commit - Estado inicial
ariel@AMD:~/Escritorio/git01$
```

git revert: Al comando git revert se le puede considerar un comando para "deshacer", pero lo cierto es que no lo es en el sentido tradicional. En lugar de eliminar la confirmación del historial del proyecto, resuelve cómo invertir los cambios introducidos por la confirmación y añade una nueva con el contenido inverso resultante. Así, se evita que Git pierda el historial, lo cual resulta importante para la integridad del historial de revisiones y para que la colaboración sea fiable.

```
git revert b5c077a0e87a9dfc76624ac496c6d64b6d676264
```

Avisa que hay conflictos. Esto es por que el revert tiene un **merge** implicito... Y hay que solucionarlo: Podemos continuar o abortar:

```
ariel@AMD: ~/Escritorio/git01
File Edit View Search Terminal Help

ariel@AMD:~/Escritorio/git01$ git revert b5c077a0e87a9dfc76624ac496c6d64b6d676264
Auto-fusionando index.html
CONFLICTO (contenido): Conflicto de fusión en index.html
Auto-fusionando style.css
error: no se pudo revertir b5c077a... Segundo commit - H1 y P, mas comentario
ayuda: Tras resolver los conflictos, márquelos con
ayuda: "git add/rm <paths>", luego ejecute
ayuda: "git revert --continue".
ayuda: O puede saltar este commit con "git revert --skip".
ayuda: Para abortar y regresar al estado anterior a "git revert",
ayuda: ejecute "git revert --abort".
ariel@AMD:~/Escritorio/git01$
```

continuar:

```
git add .
git revert --continue
```

```
ariel@AMD: ~/Escritorio/git01
File Edit View Search Terminal Help

ariel@AMD:~/Escritorio/git01$ git add .
ariel@AMD:~/Escritorio/git01$ git revert --continue
[master a4d8f81] Revert "Segundo commit - H1 y P, mas comentario"
2 files changed, 4 insertions(+), 1 deletion(-)
```

Si hacemos un **git log**, vemos un nuevo **commit**, que es el punto el que revertimos fusionado con el ultimo:

```
ariel@AMD: ~/Escritorio/git01
File Edit View Search Terminal Help

ariel@AMD:~/Escritorio/git01$ git log
commit a4d8f8110b19a8db510c93bb0ee4be676c8969a4 (HEAD -> master)
Author: ArielCodo <ariel.palazzesi@bue.edu.ar>
Date:   Fri Apr 14 15:15:47 2023 -0300

    Revert "Segundo commit - H1 y P, mas comentario"

    This reverts commit b5c077a0e87a9dfc76624ac496c6d64b6d676264.

commit 9db0b1bcf725995c011e7fa2b81433be16887070 (cambios)
Author: ArielCodo <ariel.palazzesi@bue.edu.ar>
Date:   Fri Apr 14 10:39:20 2023 -0300

    Primer commit en la rama cambios

commit b5c077a0e87a9dfc76624ac496c6d64b6d676264
Author: ArielCodo <ariel.palazzesi@bue.edu.ar>
Date:   Fri Apr 14 10:26:10 2023 -0300

    Segundo commit - H1 y P, mas comentario

commit a6b1826c6dff9d1ca7d21a1d2f0865f8975f2a6a
Author: ArielCodo <ariel.palazzesi@bue.edu.ar>
Date:   Fri Apr 14 10:21:50 2023 -0300

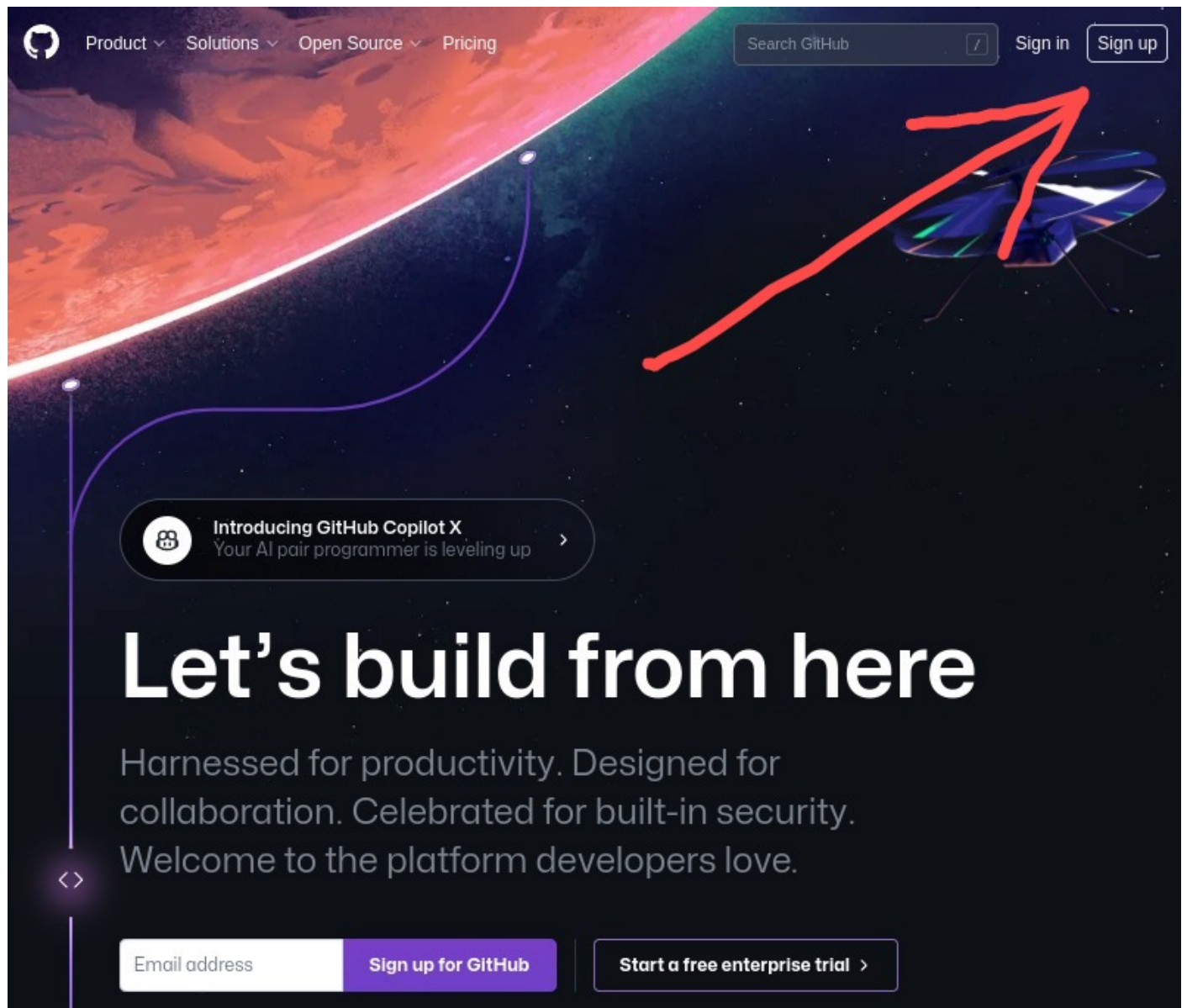
    Primer commit - Estado inicial
ariel@AMD:~/Escritorio/git01$
```

Actividad 2: Repositorios remotos

Preliminares:

1) Acceder a Github y crear una cuenta:

<https://github.com/>



2) Creamos un repositorio:

Publico, sin agregar un README, dejamos las opciones por defecto:

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository els [Import a repository](#).

Owner *

 ArielCodo ▾

Repository name *

/ mi_proyecto ✓

Great repository names are short and memorable. Need inspiration? How about [shiny-winner](#)?

Description (optional)

Trabajo práctico obligatorio



Public

Anyone on the internet can see this repository. You choose who can commit.



Private

You choose who can see and commit to this repository.

Initialize this repository with:

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



Add a README file

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

Add .gitignore

Choose which files not to track from a list of templates. [Learn more](#).

.gitignore template: None ▾

Choose a license

A license tells others what they can and can't do with your code. [Learn more](#).

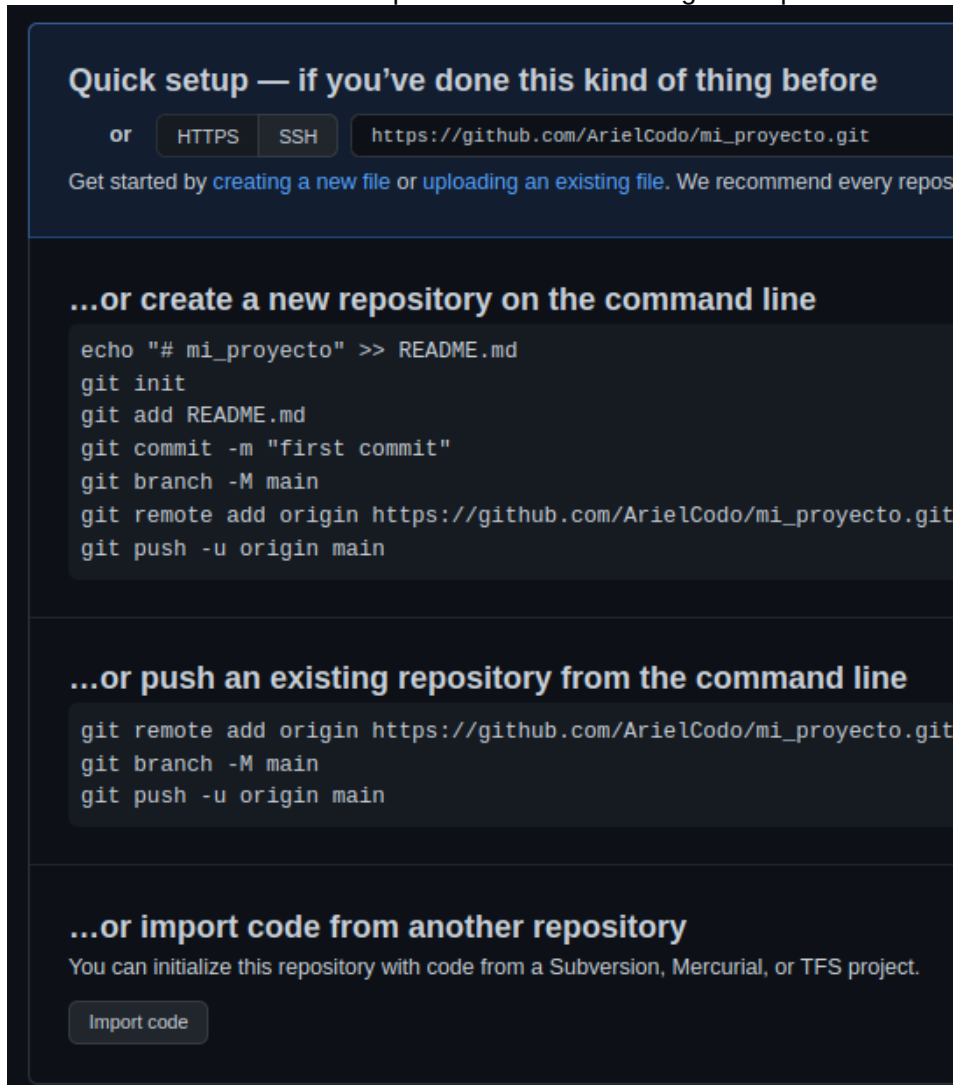
License: None ▾



You are creating a public repository in your personal account.

Create repository

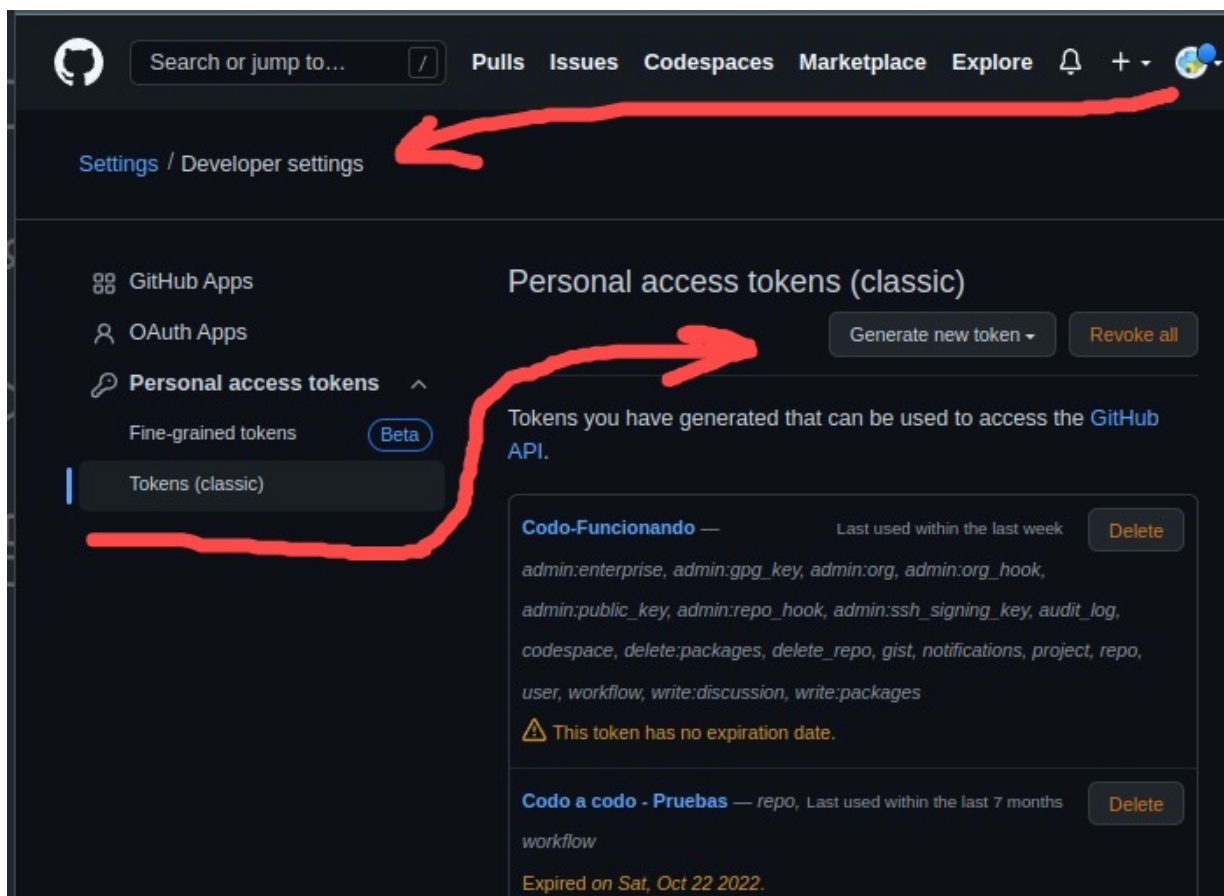
Presionando el botón "Crear repositorio" veremos la siguiente pantalla.



Ya tenemos nuestro repositorio vacío en GitHub.
Ahora podemos, por ejemplo, vincularlo con una carpeta de nuestra PC.

3) Generamos un token en GitHub:

Creemos y GUARDAMOS el token (clásico) desde GitHub.



El token tiene este aspecto: ghp_Otzjg4NFomNt5Y6Vb3KxXtE8shP9OGsx8iCn

4) Vinculamos el repositorio con un repositorio local:

En este punto, podemos crear en nuestra PC un proyecto nuevo y convertirlo en un repositorio GIT como hicimos antes, o tomar un repositorio local existente y vincularlo con el que creamos en GitHub. Vamos a usar para ello la carpeta que usamos en la primera parte de este texto: git01

Nos aseguramos de que no haya cambios pendientes. Si los hay, hacemos un git add y un nuevo commit:

```
ariel@AMD: ~/Escritorio/git01
File Edit View Search Terminal Help
ariel@AMD:~/Escritorio/git01$ git status
En la rama master
Cambios no rastreados para el commit:
(usa "git add <archivo>..." para actualizar lo que será confirmado)
(usa "git restore <archivo>..." para descartar los cambios en el directorio de
trabajo)
modificados:    index.html

sin cambios agregados al commit (usa "git add" y/o "git commit -a")
ariel@AMD:~/Escritorio/git01$ git add .
ariel@AMD:~/Escritorio/git01$ git commit -m "Commit previo a vincular con github"
[master 4eadb6a] Commit previo a vincular con github
1 file changed, 3 deletions(-)
ariel@AMD:~/Escritorio/git01$
```

La rama principal de nuestro proyecto, desde hace algún tiempo, debe llamarse **main**. Cambiamos el nombre, de ser necesario, haciendo:

```
git branch -M main
```

y finalmente vinculamos ambos repositorios:

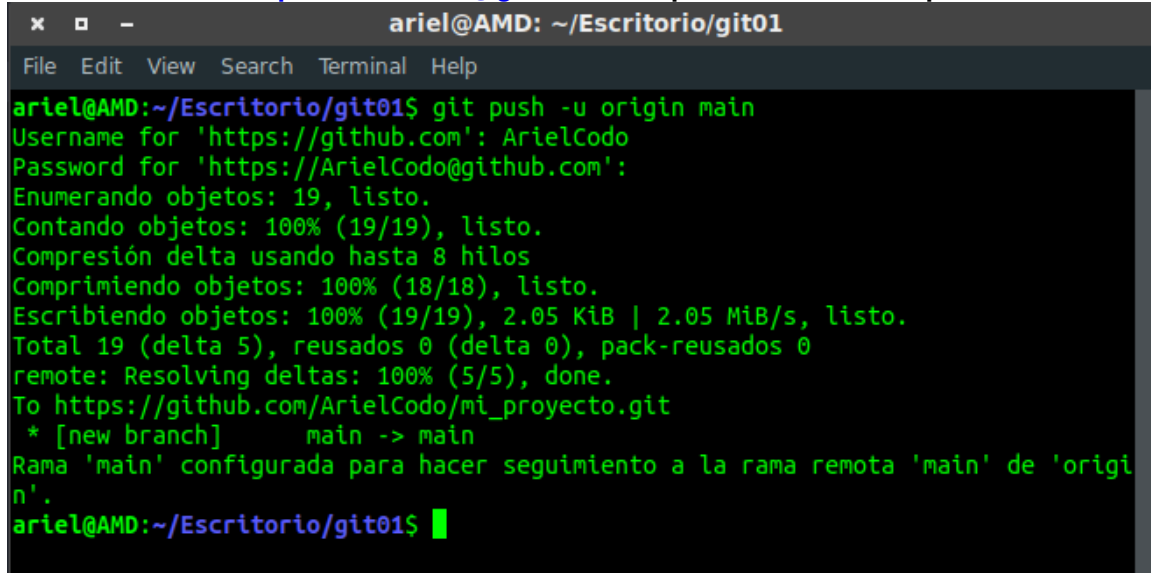

```
git remote add origin https://github.com/ArielCodo/mi_proyecto.git
```

Ahora podemos sincronizar el contenido de nuestra carpeta local con la carpeta remota:

```
git push -u origin main
```

En Username for '<https://github.com>':* ponemos nuestro usuario de [GitHub](#)

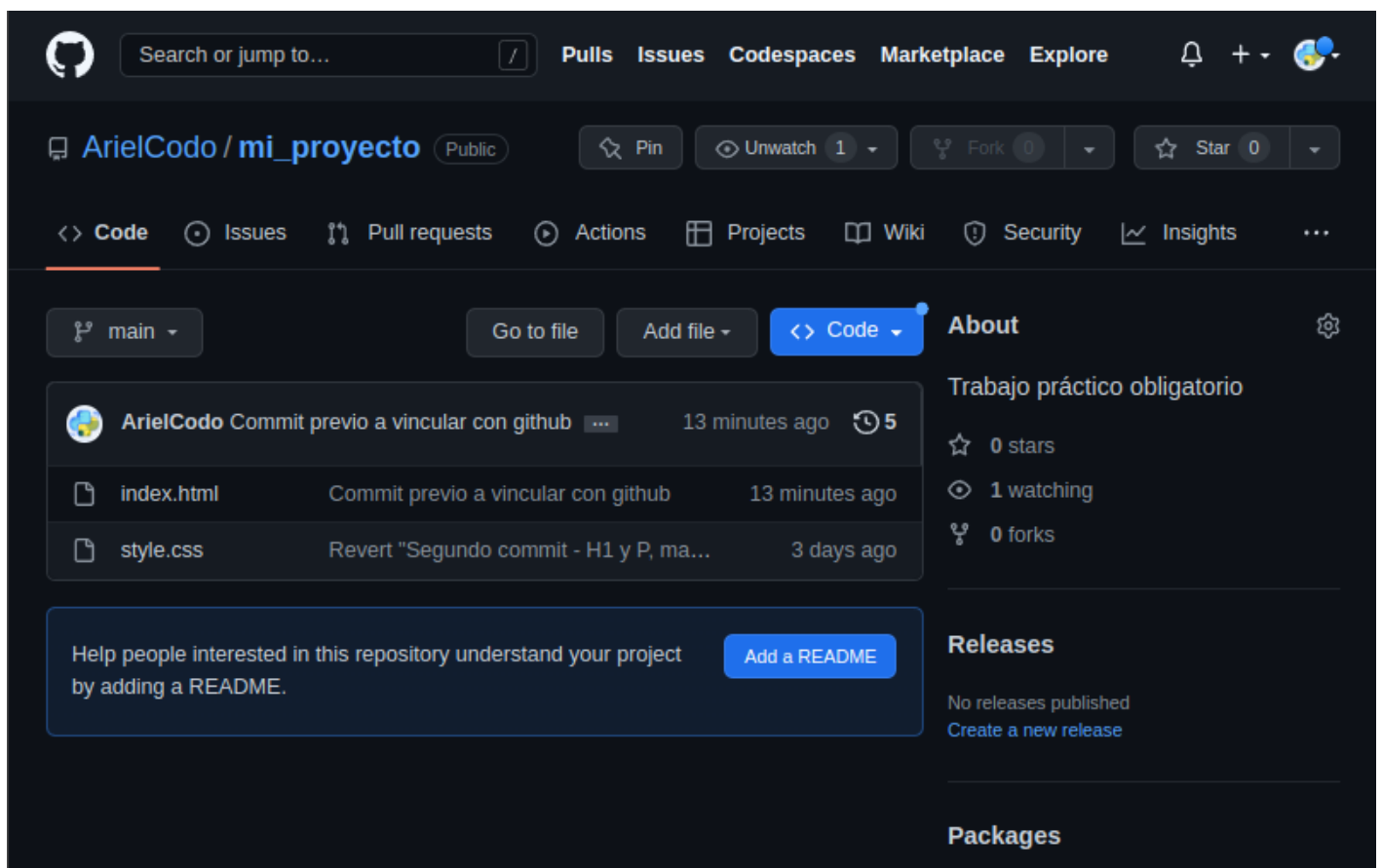
En Password for '<https://ArielCodo@github.com>': ponemos el token que obtuvimos en el paso 3)*



```
ariel@AMD: ~/Escritorio/git01
File Edit View Search Terminal Help

ariel@AMD:~/Escritorio/git01$ git push -u origin main
Username for 'https://github.com': ArielCodo
Password for 'https://ArielCodo@github.com':
Enumerando objetos: 19, listo.
Contando objetos: 100% (19/19), listo.
Compresión delta usando hasta 8 hilos
Comprimiendo objetos: 100% (18/18), listo.
Escribiendo objetos: 100% (19/19), 2.05 KiB | 2.05 MiB/s, listo.
Total 19 (delta 5), reusados 0 (delta 0), pack-reusados 0
remote: Resolving deltas: 100% (5/5), done.
To https://github.com/ArielCodo/mi_proyecto.git
 * [new branch]      main -> main
Rama 'main' configurada para hacer seguimiento a la rama remota 'main' de 'origin'.
ariel@AMD:~/Escritorio/git01$
```

En este punto, el repositorio de [GitHub](#) debería mostrar el mismo contenido que la carpeta local. Puede ser necesario refrescar la pantalla del navegador (F5) si aún no lo hicimos:



Podemos ver los commits realizados en la PC desde el ícono correspondiente:

The screenshot shows the GitHub repository interface for a project. At the top, there's a navigation bar with links to Code, Issues, Pull requests, Actions, Projects, Wiki, and Security. Below this, the 'main' branch is selected. The commit history is displayed, showing commits on April 17, 2023, and April 14, 2023. The most recent commit is titled 'Commit previo a vincular con github' by ArielCodo, committed 14 minutes ago. Below it, there's a section for commits on April 14, 2023, which includes a revert commit titled 'Revert "Segundo commit - H1 y P, mas comentario"' and three other commits: 'Primer commit en la rama cambios', 'Segundo commit - H1 y P, mas comentario', and 'Primer commit - Estado inicial'. Each commit entry shows the commit hash and a link to view the commit details.

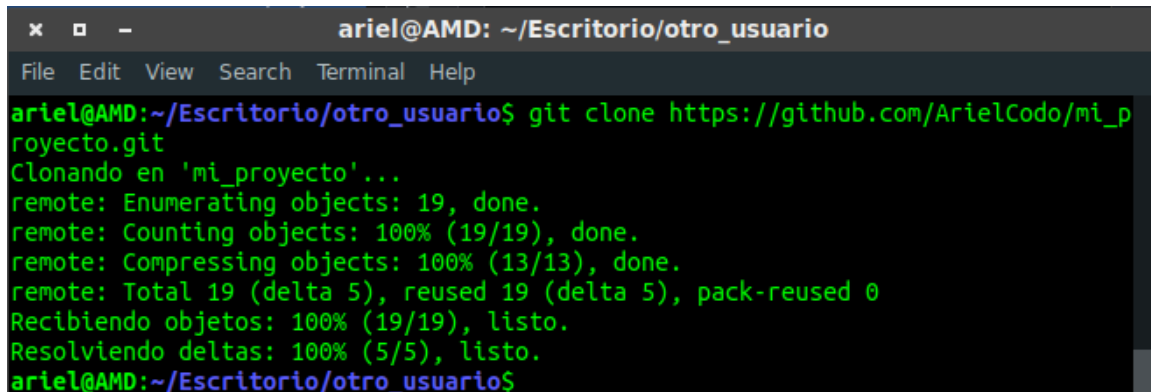
Y ver los archivos, en el estado correspondiente, en cada uno de ellos:

The screenshot shows the GitHub file diff view for the 'Primer commit en la rama cambios' commit. The commit is by ArielCodo, committed 3 days ago, and is the first parent of commit b5c077a. The diff shows changes to two files: 'index.html' and 'style.css'. For 'index.html', the diff shows a new paragraph added to the body of the HTML document. For 'style.css', the diff shows a new rule added for the 'p' element, setting its background color to 'bisque'. The interface includes a 'Browse files' button, a 'Split' button, and a 'Unified' button. At the bottom, it shows '0 comments on commit 9db0b1b' and a 'Lock conversation' button.

5) Trabajando con otro usuario en el mismo proyecto.

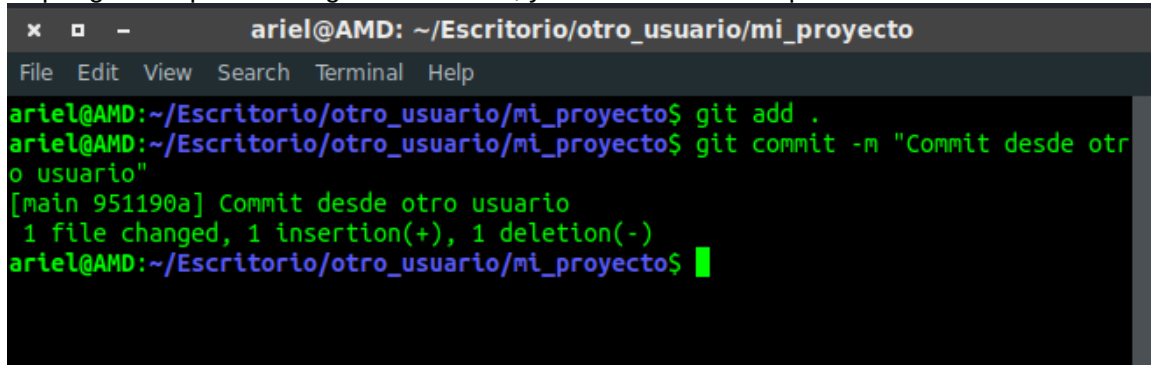
El otro usuario comienza, por ejemplo, clonando el repositorio existente:

```
git clone https://github.com/ArielCodo/mi_proyecto.git
```



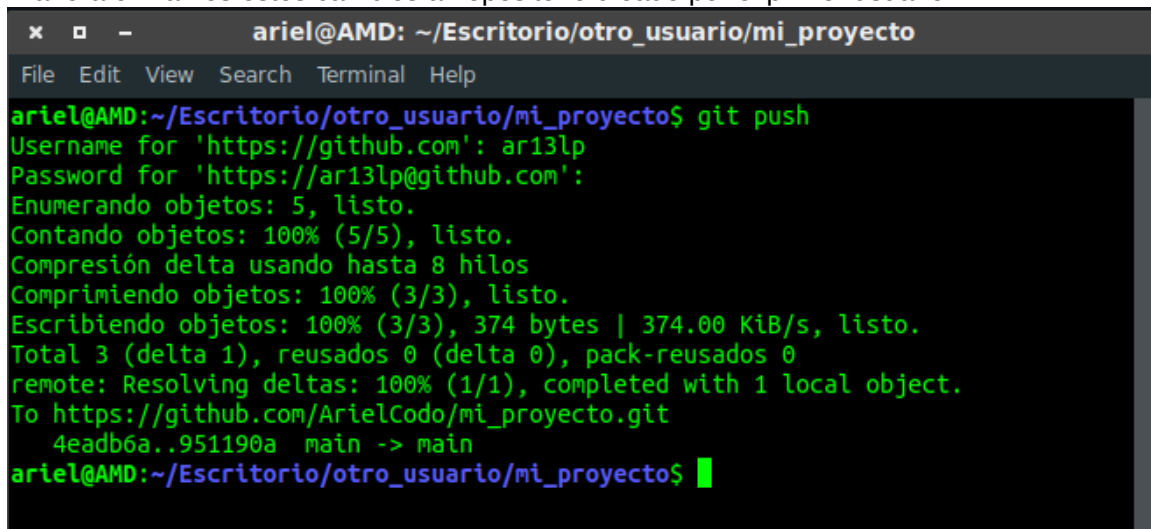
```
ariel@AMD: ~/Escritorio/otro_usuario
File Edit View Search Terminal Help
ariel@AMD:~/Escritorio/otro_usuario$ git clone https://github.com/ArielCodo/mi_p
royecto.git
Clonando en 'mi_proyecto'...
remote: Enumerating objects: 19, done.
remote: Counting objects: 100% (19/19), done.
remote: Compressing objects: 100% (13/13), done.
remote: Total 19 (delta 5), reused 19 (delta 5), pack-reused 0
Recibiendo objetos: 100% (19/19), listo.
Resolviendo deltas: 100% (5/5), listo.
ariel@AMD:~/Escritorio/otro_usuario$
```

Supongamos que hace algunos cambios, y un commit en su repositorio local:



```
ariel@AMD: ~/Escritorio/otro_usuario/mi_proyecto
File Edit View Search Terminal Help
ariel@AMD:~/Escritorio/otro_usuario/mi_proyecto$ git add .
ariel@AMD:~/Escritorio/otro_usuario/mi_proyecto$ git commit -m "Commit desde otr
o usuario"
[main 951190a] Commit desde otro usuario
1 file changed, 1 insertion(+), 1 deletion(-)
ariel@AMD:~/Escritorio/otro_usuario/mi_proyecto$
```

Y ahora enviamos estos cambios al repositorio creado por el primer usuario:



```
ariel@AMD: ~/Escritorio/otro_usuario/mi_proyecto
File Edit View Search Terminal Help
ariel@AMD:~/Escritorio/otro_usuario/mi_proyecto$ git push
Username for 'https://github.com': ar13lp
Password for 'https://ar13lp@github.com':
Enumerando objetos: 5, listo.
Contando objetos: 100% (5/5), listo.
Compresión delta usando hasta 8 hilos
Comprimiendo objetos: 100% (3/3), listo.
Escribiendo objetos: 100% (3/3), 374 bytes | 374.00 KiB/s, listo.
Total 3 (delta 1), reusados 0 (delta 0), pack-reusados 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/ArielCodo/mi_proyecto.git
4eadb6a..951190a main -> main
ariel@AMD:~/Escritorio/otro_usuario/mi_proyecto$
```

En Username for 'https://github.com':** ponemos el usuario de [GitHub](#) del colaborador (segundo usuario)
En Password for 'https://ar13lp@github.com': ponemos el token que obtuvimos en el paso 3)*

El segundo usuario puede ser invitado a colaborar, y usar sus propias credenciales:

Who has access


PUBLIC REPOSITORY
This repository is public and visible to anyone.
[Manage](#)


DIRECT ACCESS
1 has access to this repository. **0** collaborators. **1** invitation.

Manage access

☐ **Select all**



☐

**Ariel**
Awaiting ar13lp's response

Pending Invite 

Hay que reponder el correo que llega al colaborador para poder interactuar.

GitHub

 + 

@ArielCodo has invited you to collaborate on the **ArielCodo/mi_proyecto** repository

You can [accept](#) or [decline](#) this invitation. You can also head over to https://github.com/ArielCodo/mi_proyecto to check out the repository or visit [@ArielCodo](#) to learn a bit more about them.

This invitation will expire in 7 days.

[View invitation](#)

Note: This invitation was intended for misdatosonline@gmail.com. If you were not expecting this invitation, you can ignore this email. If @ArielCodo is sending you too many emails, you can [block them](#) or [report abuse](#).

Tenemos que hacer click en el botón "Join" para unirnos:

Organizations

New organization

 **mi_proyecto** ⚠ Invitation expires in 7 days. ●

Join

Decline

 **me50** outside collaborator on 1 repository

Leave



ArielCodo invited you to collaborate

Accept invitation

Decline

🔒 Owners of mi_proyecto will be able to see:

- Your public profile information
- **Certain activity** within this repository
- Country of request origin
- Your access level for this repository
- Your IP address

Is this user sending spam or malicious content?

[Block ArielCodo](#)

Ahora el segundo usuario (el colaborador) puede hacer un push con sus cambios usando su usuario y su token:

```
ariel@AMD: ~/Escritorio/otro_usuario/mi_proyecto
File Edit View Search Terminal Help

ariel@AMD:~/Escritorio/otro_usuario/mi_proyecto$ git push
Username for 'https://github.com': ar13lp
Password for 'https://ar13lp@github.com':
Enumerando objetos: 5, listo.
Contando objetos: 100% (5/5), listo.
Compresión delta usando hasta 8 hilos
Comprimiendo objetos: 100% (3/3), listo.
Escribiendo objetos: 100% (3/3), 392 bytes | 392.00 KiB/s, listo.
Total 3 (delta 1), reusados 0 (delta 0), pack-reusados 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/ArielCodo/mi_proyecto.git
  951190a..f0b57f0  main -> main
ariel@AMD:~/Escritorio/otro_usuario/mi_proyecto$
```

Si vemos el repositorio original creado por el primer usuario, vemos la colaboración realizada:

Showing **1** changed file with **1** addition and **0** deletions.

		index.html
		@@ -13,5 +13,6 @@ <h1>Git y GitHub</h1>
13	13	
14	14	<p>Esto está en la rama cambios.</p>
15	15	<p>Cambios hechos desde otro usuario....</p>
16	16	+ <p>Y otro, con el token del segundo usuario...</p>
16	17	</body>
17	18	</html>

0 comments on commit f0b57f0

```
index.html
@@ -13,5 +13,6 @@ <h1>Git y GitHub</h1>
13
14     <p>Esto está en la rama cambios.
15     <p>Cambios hechos desde otro
        usuario....</p>
16 +     <p>Y otro, con el token del segundo
        usuario...</p>
16 </body>
17 </html>
17
18 </html>
```

Ambos usuarios puede acceder al contenido del repositorio remoto.