

1. Instalar Git: <https://git-scm.com/download/win>
Siguiendo a todo, excepto en "adjusting your PATH environment", escoger la tercera opción.
Con esto se descargará una consola (bash) de git, que funciona como en linux

1.1 ¿Cómo funciona?

Working directory: área de trabajo; sistema (pc o laptop)
staging area: acumula cambios hechos
repository: versión 1.x de cambios (repositorio local)

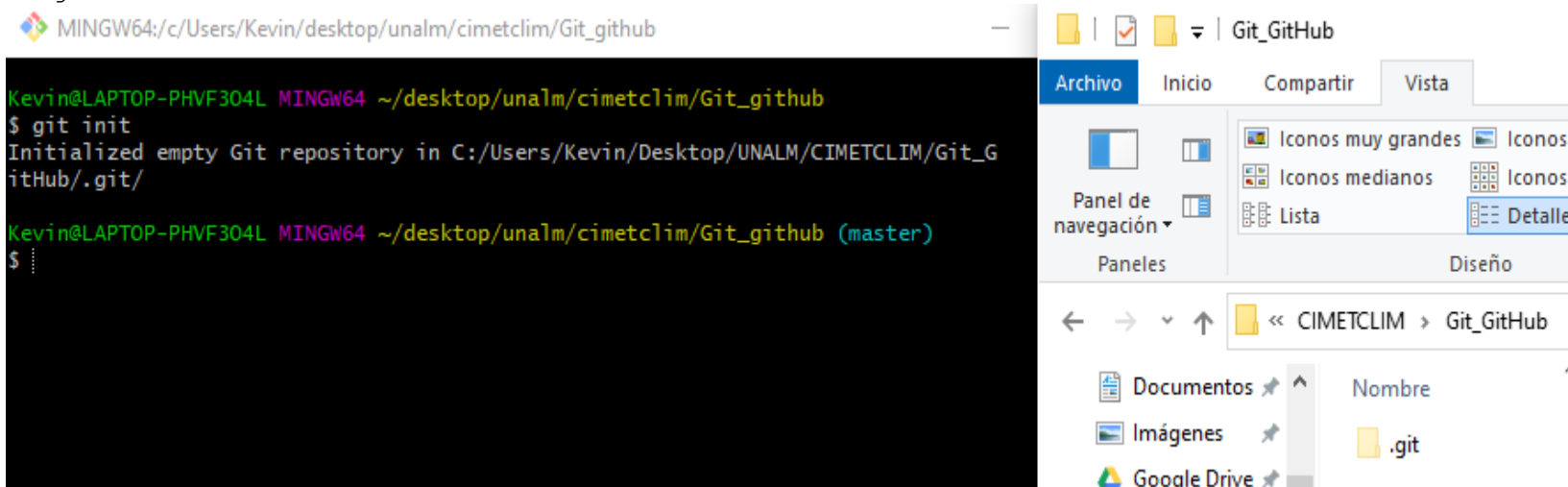
1.2 Comandos principales

git init: iniciar GIT en el proyecto
git add <file>: working directory -> staging area
git status: en qué estado está el archivo (ubicación, etc)
git commit: staging area -> repository
git push: subir a repositorio remoto, donde está todo el equipo
git pull: ver cambios de otros desarrolladores
git clone: clonar repositorio y guardarlo en tu sistema

2. Iniciar proyecto:

Usaremos el bash de git, redireccionamos a la carpeta del proyecto y ponemos:

git init



Aparecerá una carpeta con la configuración necesaria de git (no tocar nada).

3. Identificarnos

git nos exige identificarnos para saber qué miembro del equipo hizo los cambios
git config --global user.name "Nombre"
git config --global user.email "correo electrónico"

```
Kevin@LAPTOP-PHVF304L MINGW64 ~/desktop/unalm/cimetclic/Git_github (master)
$ git config --global user.name "Kevin"

Kevin@LAPTOP-PHVF304L MINGW64 ~/desktop/unalm/cimetclic/Git_github (master)
$ git config --global user.email "20180167@lamolina.edu.pe"
```

4. Agregar archivos

creamos archivos de prueba en el proyecto y si tipeamos 'git status' nos mostrará:

```
Kevin@LAPTOP-PHVF304L MINGW64 ~/desktop/unalm/cimetclic/Git_github (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
  apps.js
  index.html
  style.css

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

MEICLIM > Gi

Nombre

.git
apps
index
style

"no commits yet" nos indica que no hay archivos subidos al repository, para hacerlo:
git add <file>

```
Kevin@LAPTOP-PHVF304L MINGW64 ~/desktop/unalm/cimetclic/Git_github (master)
$ git add apps.js

Kevin@LAPTOP-PHVF304L MINGW64 ~/desktop/unalm/cimetclic/Git_github (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   apps.js

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

letras verdes: staging area; letras rojas: working directory. Agregamos el resto.
Ahora ya podemos crear nuestro primer repositorio local (commit)
git commit

```
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# On branch master
#
# Initial commit
#
# Changes to be committed:
#       new file:   apps.js
#       new file:   index.html
#       new file:   style.css
~
~
~
~
~
~
~
~
~
~
~
```

<p>UNALM/CIMETCLIM/Git_GitHub/.git/COMMIT_EDITMSG [unix] (18:01 14/12/2020)13,1 All</p><p>sktop/UNALM/CIMETCLIM/Git_GitHub/.git/COMMIT_EDITMSG" [unix] 13L, 278B</p>

La terminal cambió, se muestra similar al escribir enviar un correo, con i podemos editar presionar Esc para salir del modo editor.
sin estar en modo editar, tipeamos
:wq (así como está)

```
Kevin@LAPTOP-PHVF304L MINGW64 ~/desktop/unalm/cimetclicm/Git_github (master)
$ git commit
[master (root-commit) 981fd07] Mi primer commit
3 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 apps.js
create mode 100644 index.html
create mode 100644 style.css
```

Con ello, ya habremos subido el primer commit (repositorio local).

```
Kevin@LAPTOP-PHVF304L MINGW64 ~/desktop/unalm/cimetclim/Git_github (master)
$ git log
commit 981fd076f1e6fafc311829f78ca730a6089d6acb (HEAD -> master)
Author: Kevin <20180167@lamolina.edu.pe>
Date: Mon Dec 14 18:01:39 2020 -0500

    Mi primer commit

Kevin@LAPTOP-PHVF304L MINGW64 ~/desktop/unalm/cimetclim/Git_github (master)
$ git status
On branch master
nothing to commit, working tree clean
```

'git log' nos muestra una identificación del repositorio (o punto de guardado).

'git status' nos muestra que no hay archivo modificado o creado.

En caso no nos hayamos identificado. saltará este error:

```
Kevin@LAPTOP-PHVF304L MINGW64 ~/desktop/unalm/cimetclic/Git_github (master)
$ git commit
Author identity unknown

*** Please tell me who you are.

Run

    git config --global user.email "you@example.com"
    git config --global user.name "Your Name"

to set your account's default identity.
Omit --global to set the identity only in this repository.

fatal: unable to auto-detect email address (got 'Kevin@LAPTOP-PHVF304L.(none)')
```

5. Guardar cambios

modificaremos los archivos subidos al commit y digitamos 'git status'

```
Kevin@LAPTOP-PHVF304L MINGW64 ~/desktop/unalm/cimetclic/git_github (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.html
```

nos sale que el archivo index.html ha sido modificado, para ver los cambios:
git diff

```
Kevin@LAPTOP-PHVF304L MINGW64 ~/desktop/unalm/cimetclic/git_github (master)
$ git diff
diff --git a/index.html b/index.html
index e69de29..2351c31 100644
--- a/index.html
+++ b/index.html
@@ -0,0 +1,10 @@
+<!DOCTYPE html>
+<html lang="es" dir="ltr">
+  <head>
+    <meta charset="utf-8">
+    <title></title>
+  </head>
+  <body>
+
+  </body>
+</html>
```

+ lo que se ha agregado

- lo que se ha quitado

para agregar los nuevos cambios: `git add <file>` y `git commit`

```
Kevin@LAPTOP-PHVF304L MINGW64 ~/desktop/unalm/cimetclic/git_github (master)
$ git add index.html

Kevin@LAPTOP-PHVF304L MINGW64 ~/desktop/unalm/cimetclic/git_github (master)
$ git commit
[master 1dc2a87] Nuevo cambio
1 file changed, 10 insertions(+)
```

vemos que un archivo ha sido modificado con 10 inserciones (+)

6. Descartar cambio

Antes de guardar el cambio digitamos

`git checkout -- <file>`

```
Kevin@LAPTOP-PHVF304L MINGW64 ~/desktop/unalm/cimetclic/git_github (master)
$ git checkout -- index.html

Kevin@LAPTOP-PHVF304L MINGW64 ~/desktop/unalm/cimetclic/git_github (master)
$ git status
On branch master
nothing to commit, working tree clean
```

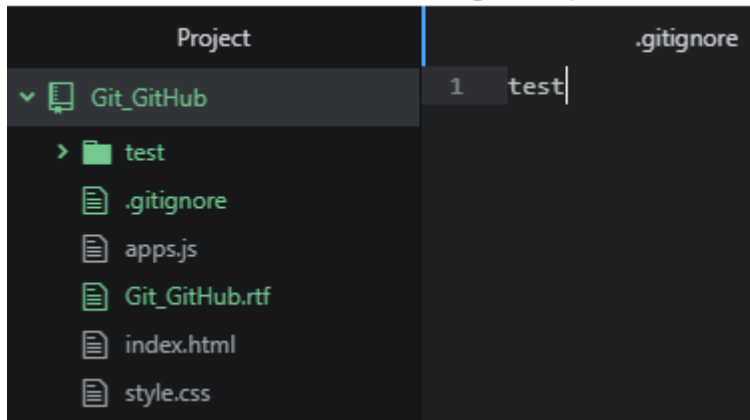
7. Ignorar ciertos archivos

Creamos una carpeta 'test' y un nuevo archivo 'prueba.js', tipeamos 'git status'

```
Kevin@LAPTOP-PHVF304L MINGW64 ~/desktop/unalm/cimetclic/git_github (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    Git_GitHub.rtf
    test/

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

nos identifica la carpeta 'test/', pero no queremos que se considere esta carpeta. por lo que creamos, en la carpeta del proyecto, un archivo llamado '.gitignore' y agregamos el nombre del archivo a ignorar.



(captura hecha desde el IDE atom)

ahora, subimos el nuevo archivo .gitignore al staging area y digitamos 'git status'

```
Kevin@LAPTOP-PHVF304L MINGW64 ~/desktop/unalm/cimetclicm/git_github (master)
$ git add .gitignore

Kevin@LAPTOP-PHVF304L MINGW64 ~/desktop/unalm/cimetclicm/git_github (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   .gitignore

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

vemos que la carpeta ya no es considerada. Mismo proceso para cualquier otro archivo. Ahora, guardemos los cambios (esta vez, omitiendo la ventana de edición con `git commit -m 'mensaje'`)

```
Kevin@LAPTOP-PHVF304L MINGW64 ~/desktop/unalm/cimetclicm/git_github (master)
$ git commit -m 'he agregado un .gitignore'
[master e30985f] he agregado un .gitignore
1 file changed, 1 insertion(+)
create mode 100644 .gitignore
```

8. Crear/cambiar ramas

Hasta ahora hemos estado trabajado en la rama 'master' (en la terminal, se ve de color turquesa) Cada rama es como si fuese una versión personalizada, enfocada a generar un cambio específico. por ejemplo, estamos simulando crear una página web (index.html, apps.js, style.css) agregaremos una nueva rama para agregar a la página web una ventana de login

`git branch login` (branch significa rama)

`git branch` (nos muestra las ramas existentes)

```
Kevin@LAPTOP-PHVF304L MINGW64 ~/desktop/unalm/cimetclic/git_github (master)
$ git branch login

Kevin@LAPTOP-PHVF304L MINGW64 ~/desktop/unalm/cimetclic/git_github (master)
$ git branch
  login
* master
```

de color verde observamos la rama en la que trabajamos actualmente, para cambiar al otro:
git checkout login

```
Kevin@LAPTOP-PHVF304L MINGW64 ~/desktop/unalm/cimetclic/git_github (master)
$ git checkout login
Switched to branch 'login'

Kevin@LAPTOP-PHVF304L MINGW64 ~/desktop/unalm/cimetclic/git_github (login)
$ git branch
* login
  master
```

A partir de aquí, cualquier cambio que hagamos estará dentro de la rama.
si crearemos archivos y guardamos los cambios

```
Kevin@LAPTOP-PHVF304L MINGW64 ~/desktop/unalm/cimetclic/git_github (login)
$ git status
On branch login
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    Git_GitHub.rtf
    login.html
    login/

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

Kevin@LAPTOP-PHVF304L MINGW64 ~/desktop/unalm/cimetclic/git_github (login)
$ git add .

Kevin@LAPTOP-PHVF304L MINGW64 ~/desktop/unalm/cimetclic/git_github (login)
$ git status
On branch login
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   Git_GitHub.rtf
    new file:   login.html
    new file:   login/login.js
```

'git add .' agrega todos los elementos del working directory
al cambiar a la otra rama veremos que los cambios hechos en login desaparecen

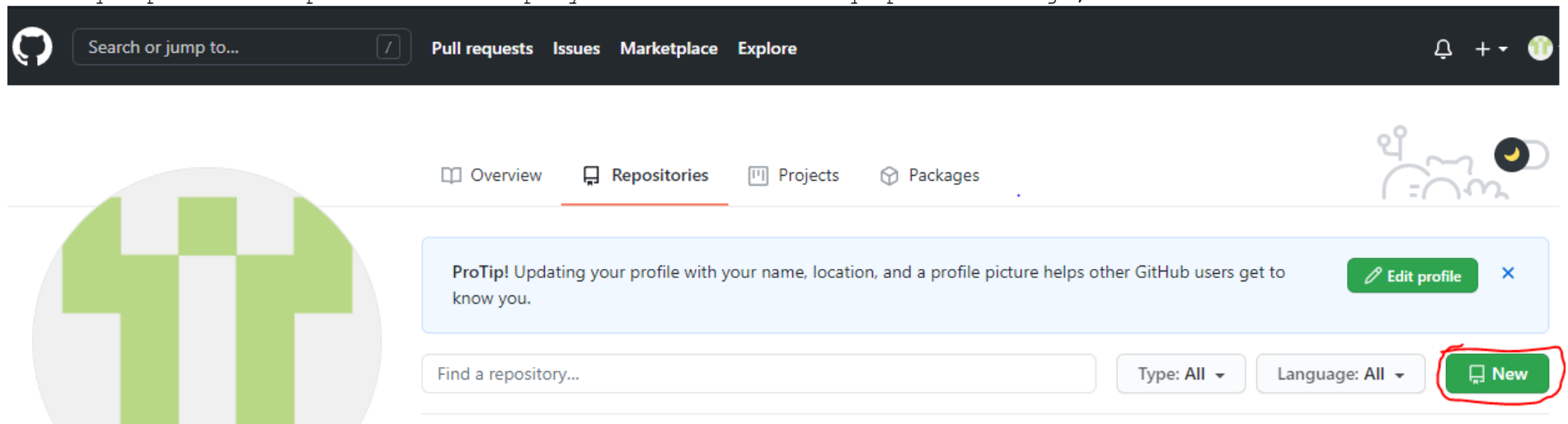
```
Project
  v Git_GitHub
    .gitignore
    apps.js
    index.html
    style.css

Kevin@LAPTOP-PHVF304L MINGW64 ~/desktop/unalm/cimetclim/git_github (login)
$ git commit -m 'login'
[login c4e300d] login
3 files changed, 79864 insertions(+)
create mode 100644 Git_GitHub.rtf
create mode 100644 login.html
create mode 100644 login/login.js


Kevin@LAPTOP-PHVF304L MINGW64 ~/desktop/unalm/cimetclim/git_github (login)
$ git checkout master
Switched to branch 'master'
```

9. Subir repositorio local a GitHub (repositorio remoto)

Aquí podemos compartir nuestro proyecto con nuestro equipo de trabajo, creamos uno nuevo





Owner * Repository name *

 KevinCampos20180167 ▾ / Prueba ✓

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

Description (optional)

☒  **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.](#)

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

Create repository

el check verde que aparece en la ventana donde ponemos nombre indica si está disponible
los repositorios públicos son gratuitos
README es un archivo de texto que ayuda al lector a entender tu proyecto (lo describes).
Ahora, para enlazar el proyecto al GitHub, ingresamos en git bash lo resaltado:

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



Set up in Desktop

or

HTTPS

SSH

<https://github.com/KevinCampos20180167/>

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository

...or create a new repository on the command line

```
echo "# Prueba" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/KevinCampos20180167/Prueba.git
git push -u origin main
```

se ejecutará una serie de comandos, al finalizar, nuestro repositorio ya estará subido.

IMPORTANTE: el último comando debe ser: 'git push -u origin master'

lo que se sube aquí es la rama 'master', no 'main'.

si funciona correctamente, se abrirá una ventana donde ingresaremos con la cuenta de GitHub



Authentication Succeeded

You may now close this tab and return to the application.

```
MINGW64:/c/Users/Kevin/desktop/unalm/cimetclim/git_github
Kevin@LAPTOP-PHVF304L MINGW64 ~
$ cd desktop/unalm/cimetclim/git_github

Kevin@LAPTOP-PHVF304L MINGW64 ~/desktop/unalm/cimetclim/git_github (master)
$ git remote add origin https://github.com/KevinCampos20180167/Prueba.git
fatal: remote origin already exists.

Kevin@LAPTOP-PHVF304L MINGW64 ~/desktop/unalm/cimetclim/git_github (master)
$ git push -u origin master
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 8 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (9/9), 774 bytes | 258.00 KiB/s, done.
Total 9 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), done.
To https://github.com/KevinCampos20180167/Prueba.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
```

Ahora, en el repositorio visualizaremos los archivos

KevinCampos20180167 / Prueba

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

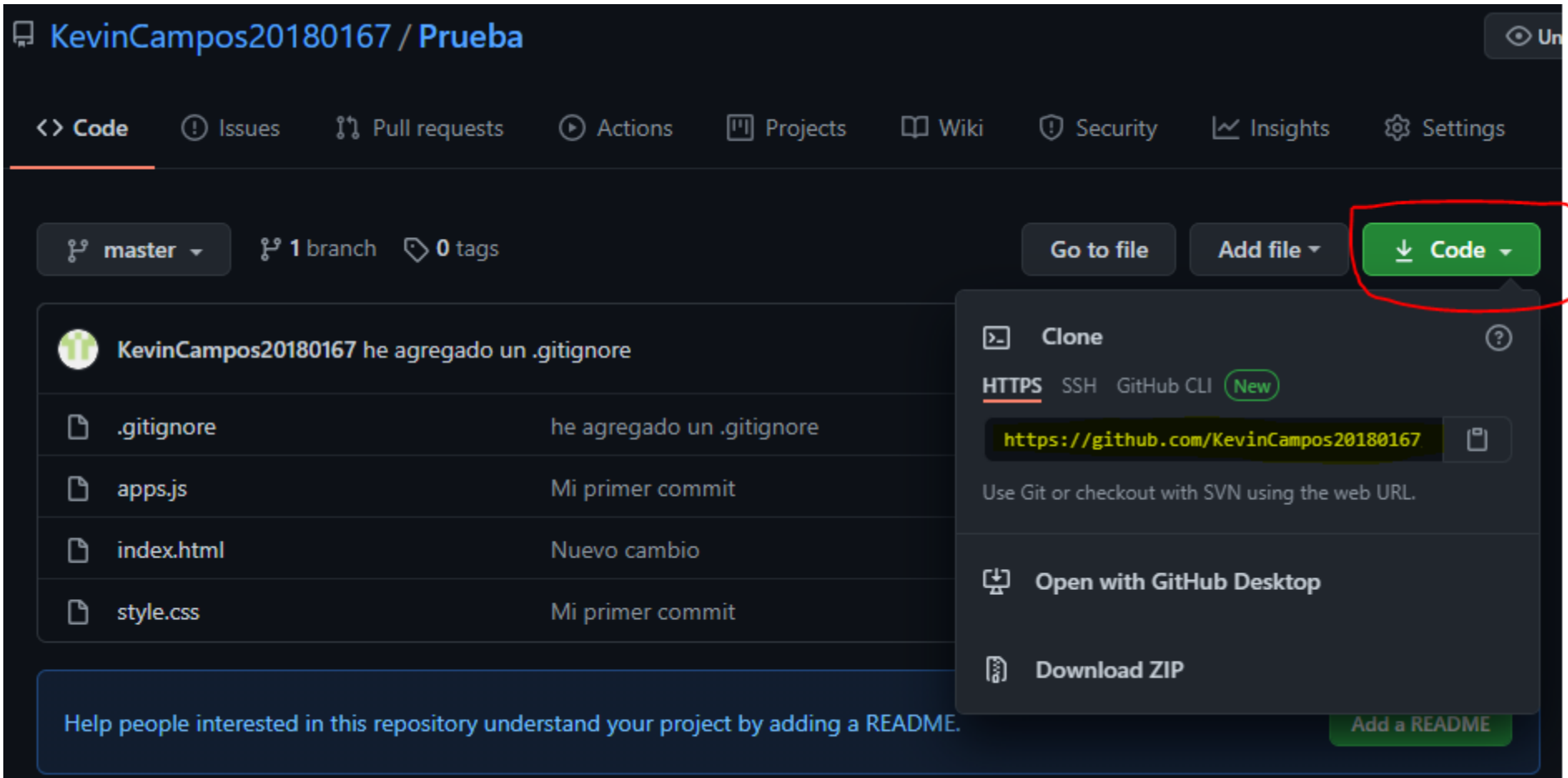
master 1 branch 0 tags Go to file Add file Code

KevinCampos20180167 he agregado un .gitignore e30985f 1 hour ago 3 commits

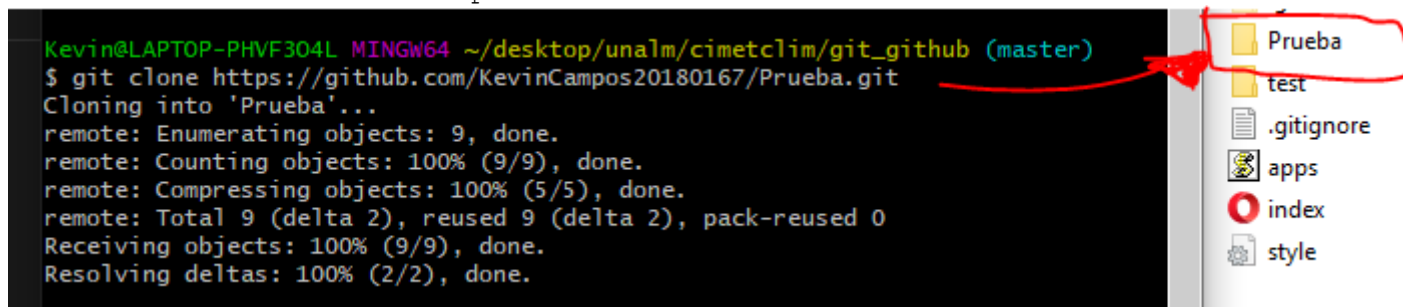
.gitignore	he agregado un .gitignore	1 hour ago
apps.js	Mi primer commit	8 hours ago
index.html	Nuevo cambio	1 hour ago
style.css	Mi primer commit	8 hours ago

Help people interested in this repository understand your project by adding a README. Add a README

10. Importar repositorio desde GitHub
Desde el repositorio, copiar lo que está subrayado



en git bash, en la carpeta donde trabajaremos, digitar:
git clone <Enlace de GitHub copiado>



Dentro de la carpeta estarán todos los archivos del repositorio, listo para trabajar.