

Tiempo para aprender juntos

Dirección de Servicios de
Infraestructura y Operaciones
Septiembre 2020

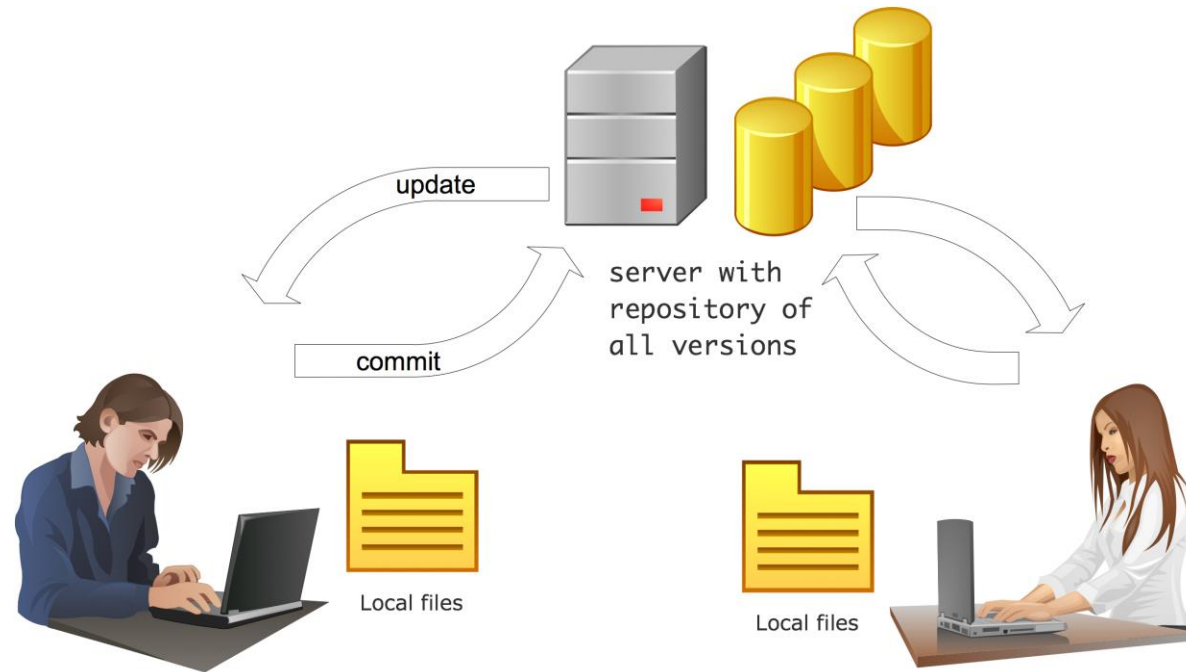


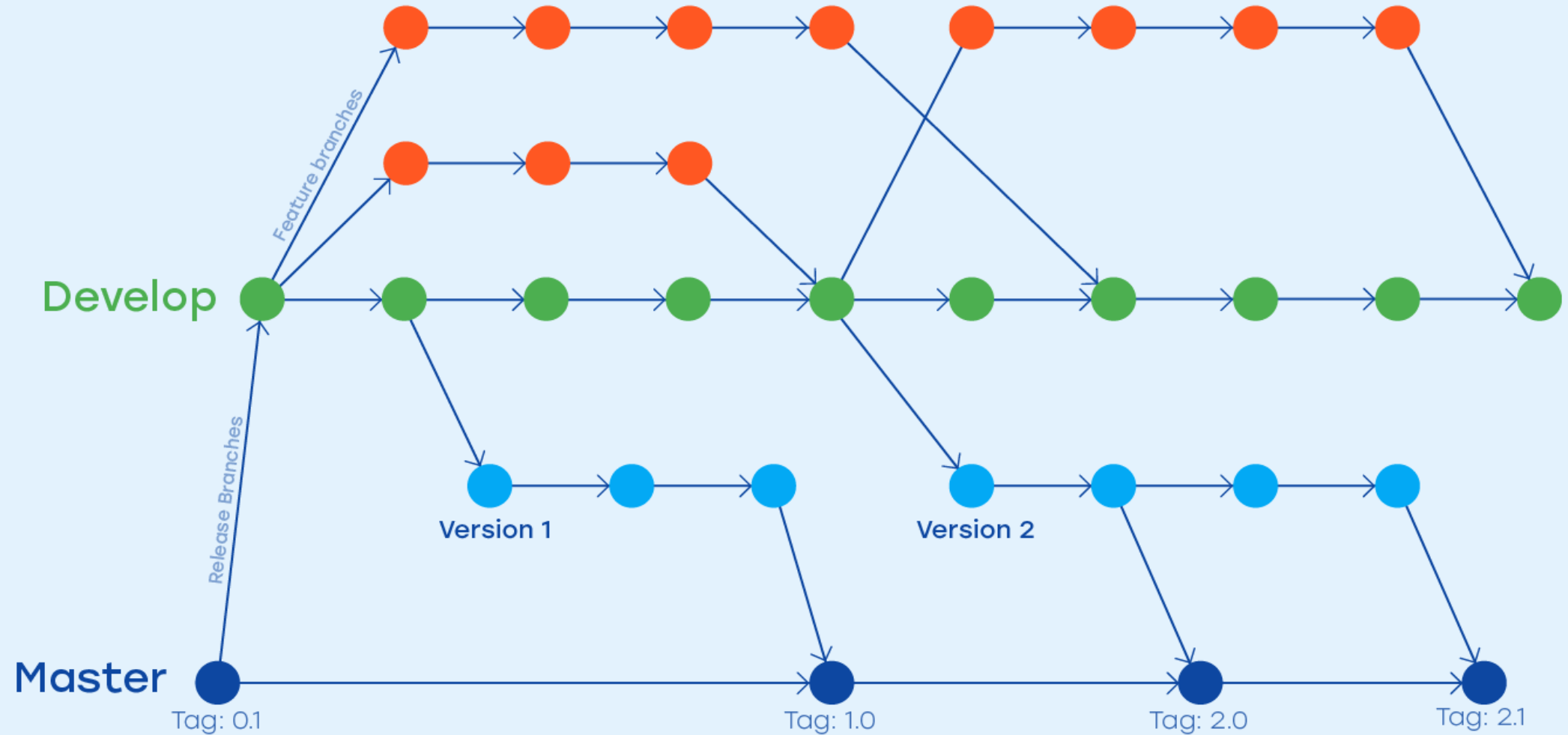
Source Code Management

- ¿Qué es SCM?
- ¿Qué es Git?
- ¿Qué es GitHub?
- Laboratorios
- Quiz

¿Qué es SCM?

- **Source Code Management (SCM)** es la practica encargada de dar *seguimiento y gestión* a los cambios hecho al código.
- Provee un historial de la ejecución del desarrollo de código.
- Ayuda a **resolver conflictos** al fusionar contribuciones de *múltiples fuentes*.

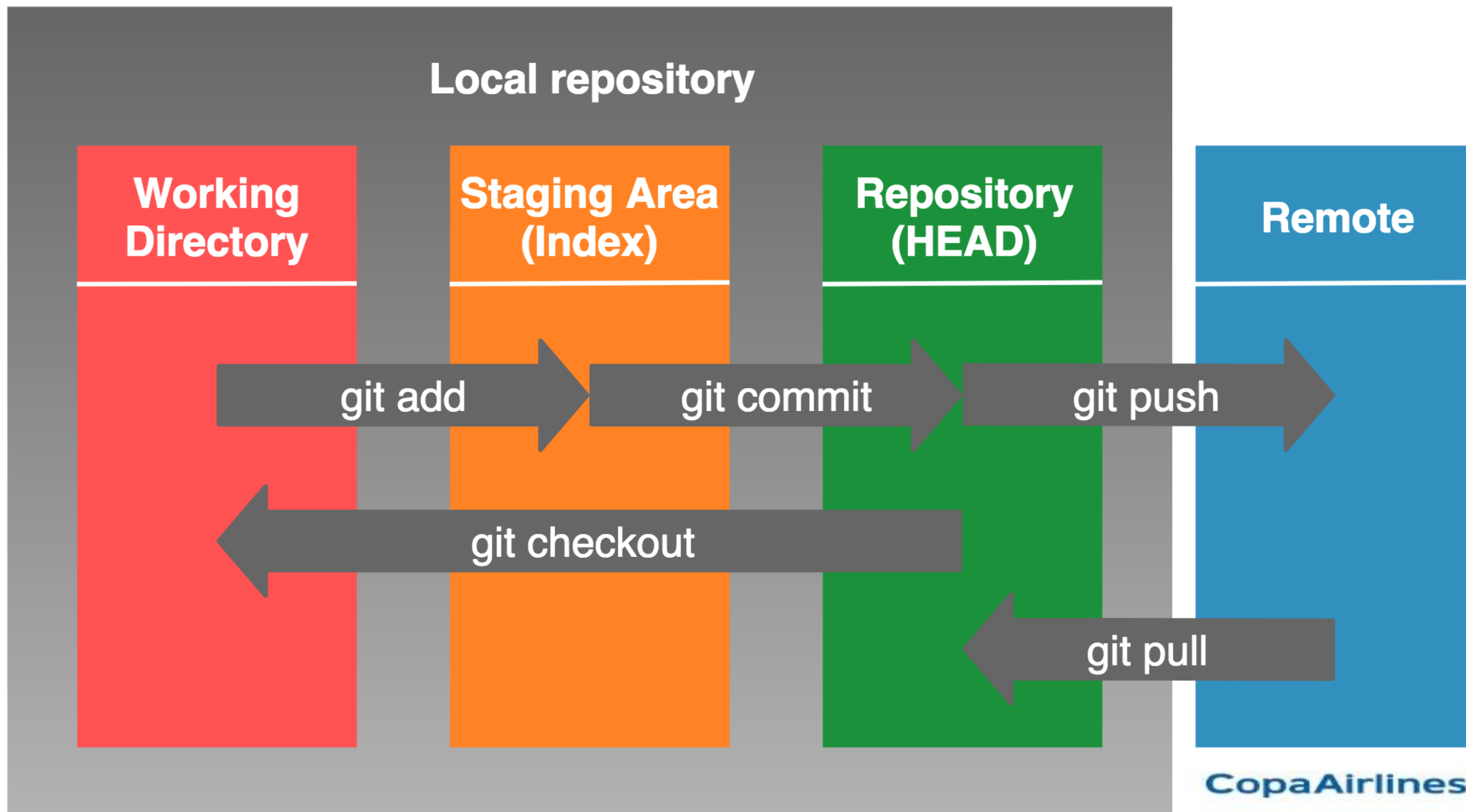




Standard development



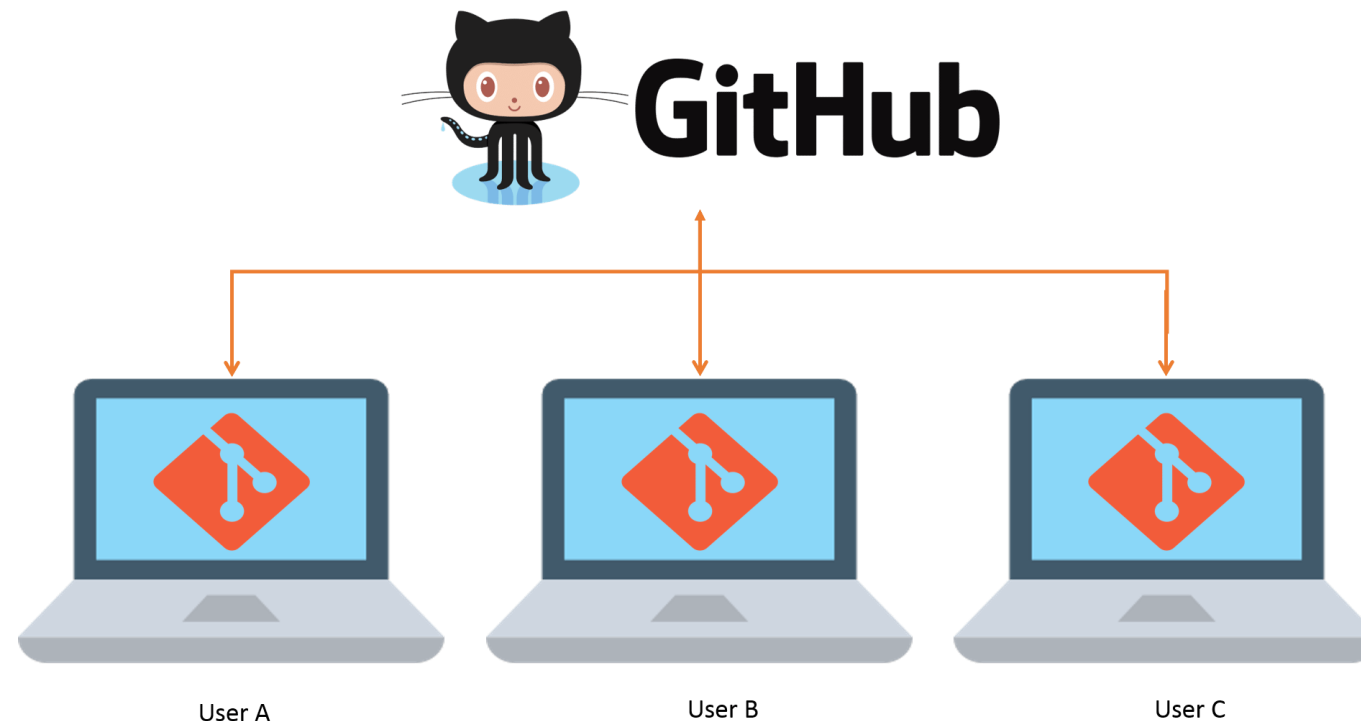
- Es un SCM *open-source*.
- Permite crear una copia de tu repositorio (**branch**).
- Puedes trabajar en ajustar el código de manera independiente para su posterior actualización (**commit**).
- Puedes empujar (**push**) o jalar (**pull**) actualizaciones hacia/de otros contribuidores.
- Puedes fusionar los cambios en el repositorio central (**merge**).



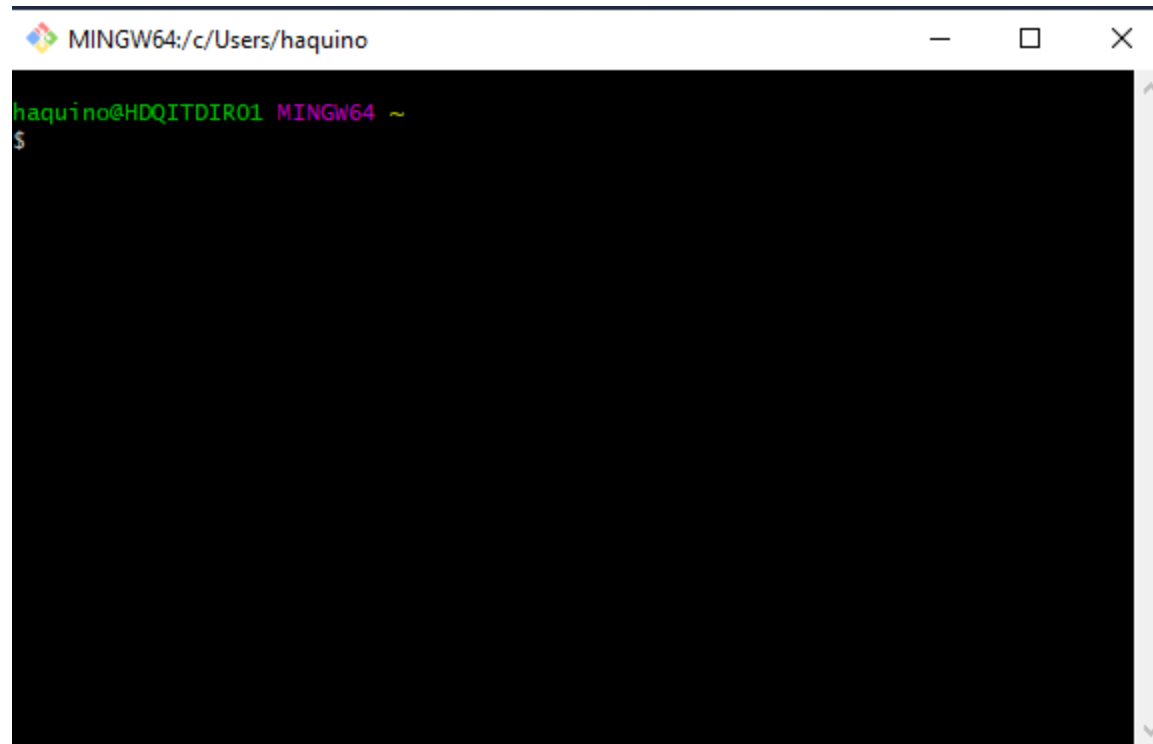


¿Qué es GitHub?

- Es una **plataforma** de **desarrollo colaborativo** para alojar proyectos utilizando Git.
- El código de los proyectos alojados en GitHub se almacena típicamente de forma **pública**.
- En 2018 Microsoft compró GitHub por **7,500 millones de dólares**.



- Instalar **Git** en tu laptop: <https://git-scm.com/download/win>
 - Instalarlo siguiendo las indicaciones del wizard
 - Abrir **Git Bash** o la consola para usar Git



- Comprobar la versión instalada: “git --version”

```
$ git --version  
git version 2.28.0.windows.1
```


- Configurar Git: nombre, correo electrónico y editor
 - `git config --global user.name "Tu nombre y apellido"`
 - `git config --global user.email "Tu dirección de correo electrónico personal"`
 - `git config --global core.editor "nano -w"`
- Para revisar la configuración
 - `git config --list`

```
$ git config --list
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=openssl
http.sslcainfo=C:/Program Files/Git/mingw64/ssl/certs/ca-bundle.crt
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
credential.helper=manager
user.name=Hugo Aquino
user.email=hugo.aquino.navarrete@gmail.com
user.mail=hugo.aquino.navarrete@gmail.com
core.editor=nano -w
```

Laboratorio – Primer repositorio

- Ve al directorio \$HOME, crea un directorio llamado “repositorio”, ve a ese directorio y muestra en pantalla el directorio donde te encuentras
 - `cd ~`
 - `mkdir repositorio`
 - `cd repositorio`
 - `pwd`
- Valida el estado de git, ¿qué te aparece?
- Inicializa git y valida nuevamente su estado
 - `git init`
 - `git status`

```
MINGW64 ~/repositorio
$ git status
fatal: not a git repository (or any of the parent directories): .git

MINGW64 ~/repositorio
$ git init
Initialized empty Git repository in c:/Users/haquino/repositorio/.git/

MINGW64 ~/repositorio (master)
```

- Crea un archivo llamado “mensaje.txt” y agrégale un sencillo texto
 - nano mensaje.txt



```
MINGW64:/c/Users/haquino/repositorio
GNU nano 4.9.3 mensaje.txt
Primer archivo de mi primer repositorio en Git !!!!

[ Read 1 line ]
^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Paste Text ^T To Spell  ^_ Go To Line
```

- Para salvar la oración teclea “Control + O” seguido de un “Enter” y para salir “Control + X”
- Valida el contenido del archivo tecleando “cat mensaje.txt”

```
$ cat mensaje.txt
Primer archivo de mi primer repositorio en Git !!!!
```

- Valida el status de git
 - git status

```
$ git status
on branch master

No commits yet

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

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

- Pon el archivo en el “**staging area**”
 - git add mensaje.txt

```
$ git add mensaje.txt
warning: LF will be replaced by CRLF in mensaje.txt.
The file will have its original line endings in your working directory

MINGW64 ~/repositorio (master)
$ git status
on branch master

No commits yet

changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   mensaje.txt
```

- Crea un commit agregando un mensaje “Iniciando notas del primer archivo en git”
 - `git commit -m “Iniciando notas del primer archivo en git”`

```
$ git commit -m "Iniciando notas del primer archivo en git"
[master (root-commit) 6e45418] Iniciando notas del primer archivo en git
1 file changed, 1 insertion(+)
create mode 100644 mensaje.txt
```

- Valida el estado del repositorio
 - `git status`

```
$ git status
On branch master
nothing to commit, working tree clean
```

- Verifica el registro de los commits llevados a cabo
 - `git log`

Laboratorio – Edita nuevamente archivo

- Agrega otro texto al archivo “mensaje.txt” y verifica el contenido del archivo
 - nano mensaje.txt
 - cat mensaje.txt
- Valida el estado del repositorio

- git status

```
$ 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:   mensaje.txt

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

- Ve las diferencias entre el estado actual del archivo y la versión guardada más reciente

- git log

```
$ git diff
warning: LF will be replaced by CRLF in mensaje.txt.
The file will have its original line endings in your working directory
diff --git a/mensaje.txt b/mensaje.txt
index ee6ded3..8ea6e75 100644
--- a/mensaje.txt
+++ b/mensaje.txt
@@ -1,2 @@
Primer archivo de mi primer repositorio en Git !!!!
+Estamos aprendiendo Git
```

- Haz otro commit agregando una breve descripción
 - `git commit -m "se agrega otra linea a mensaje.txt"`

```
$ git commit -m "Se agrega otra linea al archivo mensaje.txt"
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:   mensaje.txt

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

- Haz el commit agregando el archivo
 - `git add mensaje.txt`
 - `git commit -m "se agrega otra linea a mensaje.txt"`

```
$ git add mensaje.txt
warning: LF will be replaced by CRLF in mensaje.txt.
The file will have its original line endings in your working directory

MINGW64 ~/repositorio (master)
$ git commit -m "Se agrega otra linea al archivo mensaje.txt"
[master 9e9e512] Se agrega otra linea al archivo mensaje.txt
1 file changed, 1 insertion(+)
```

Laboratorio – Edita, stagging y commit

- Agreguen otra línea al archivo “mensaje.txt”
 - `nano mensaje.txt`
- Revisa las diferencias
 - `git diff`
- Agrega el archivo al “stagging area”
 - `git add mensaje.txt`
- Muestra la diferencia entre el último cambio que sí hizo commit y lo que está en el “stagging area”
 - `git diff --staged`
- Haz el commit agregando el archivo
 - `git commit -m “Seguimos aprendiendo”`
- Revisa el status y los logs reduciendo la cantidad de información
 - `git status`
 - `git log --oneline`

Laboratorio – Commit a varios archivos

- Agrega otra línea al archivo “mensaje.txt”
- Crea dos archivos adicionales “libro.txt” y “Netflix.txt” y agrégales una línea a cada uno
- Agrega los 3 archivos al “staging area”
- Haz el commit
- Revisa el status y los logs reduciendo la cantidad de información

Laboratorio – Explorando el History

- Uno puede ver los commits previos

```
$ git log --oneline
c40186a (HEAD -> master) Se edita mensaje.txt y se agregan 2 nuevos archivos
5178cef Seguimos aprendiendo
9e9e512 Se agrega otra línea al archivo mensaje.txt
6e45418 Iniciando notas del primer archivo en git
```

- Una manera sencilla es a través del ID del commit (al menos los primeros 4 caracteres) y el nombre del archivo
 - `git show <id_commit> <archivo>`
- Ahora sin querer queriendo, sobre escriban el archivo “mensaje.txt”
 - `echo “sin querer queriendo” > mensaje.txt`
- Valida el estado de git y el contenido de “mensaje.txt”

```
$ 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:   mensaje.txt

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

```
$ cat mensaje.txt
sin querer queriendo
```

Laboratorio – Explorando el History

- Para regresar a la versión anterior usaremos “git checkout <id_commit> <archivo>”

```
$ git log --oneline
c40186a (HEAD -> master) Se edita mensaje.txt y se agregan 2 nuevos archivos
5178cef seguimos aprendiendo
9e9e512 Se agrega otra linea al archivo mensaje.txt
6e45418 Iniciando notas del primer archivo en git
```

– git checkout c401 mensaje.txt

- Valida el status de git y el contenido de “mensaje.txt”
 - git status
 - cat mensaje.txt

Laboratorio – Deshaciendo un commit

- Sobre escribe el archivo “libro.txt” con un mensaje, agrégalo y haz un commit
 - echo “me equivoque” > libro.txt
 - git status
 - git add
 - git commit -m “Sigo agregando contenido”

```
$ git log --oneline
0941f5d (HEAD -> master) Sigo agregando contenido
c40186a Se edita mensaje.txt y se agregan 2 nuevos archivos
5178cef Seguimos aprendiendo
9e9e512 Se agrega otra linea al archivo mensaje.txt
6e45418 Iniciando notas del primer archivo en git
```

- Para regresar el estado anterior vamos hacer un checkout del HEAD -1
 - git checkout HEAD libro.txt
 - git checkout HEAD~1 libro.txt

```
$ git checkout HEAD libro.txt
Updated 0 paths from 66b5ee1
```

```
$ git checkout HEAD~1 libro.txt
Updated 1 path from 535505a
```

```
$ cat libro.txt
The Unicorn Project - Gene Kim
```

- Valida el estado de git
 - git status

```
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   libro.txt
```

- Restaura del “staging area”, muestra los logs y el estado de git
 - git restore --staged libro.txt
 - git log --oneline
 - git status

```
$ git log --oneline
0941f5d (HEAD -> master) Sigo agregando contenido
c40186a Se edita mensaje.txt y se agregan 2 nuevos archivos
5178cef Seguimos aprendiendo
9e9e512 Se agrega otra linea al archivo mensaje.txt
6e45418 Iniciando notas del primer archivo en git

leaudea@ubuntu:~/repositorio$ git restore --staged libro.txt
$ 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:   libro.txt

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

Laboratorio – Deshaciendo un commit

- Ve las diferencias del último commit para el archivo libro.txt
 - `git diff 0941 libro.txt`

```
$ git diff 0941
diff --git a/libro.txt b/libro.txt
index cf08b6d..d38aa50 100644
--- a/libro.txt
+++ b/libro.txt
@@ -1,2 @@
-me equivoque
+The Unicorn Project - Gene Kim
+
```

- Agrega libro.txt y haz un commit
 - `git add libro.txt`
 - `git commit -m "Regresando libro.txt al contenido original"`
 - `git status`
 - `git log --oneline`

```
$ git commit -m "Regresando libro.txt al contenido original"
[master 225701e] Regresando libro.txt al contenido original
1 file changed, 2 insertions(+), 1 deletion(-)

MINGW64 ~/repositorio (master)
$ git status
On branch master
nothing to commit, working tree clean

MINGW64 ~/repositorio (master)
$ git log --oneline
225701e (HEAD -> master) Regresando libro.txt al contenido original
0941f5d sigo agregando contenido
c40186a Se edita mensaje.txt y se agregan 2 nuevos archivos
5178cef Seguimos aprendiendo
9e9e512 Se agrega otra linea al archivo mensaje.txt
6e45418 Iniciando notas del primer archivo en git
```

Laboratorio – Creando un nuevo archivo antes de ir a GitHub

- Crea un archivo que se llame “README.md” y agrega brevemente algunas de tus habilidades técnicas.

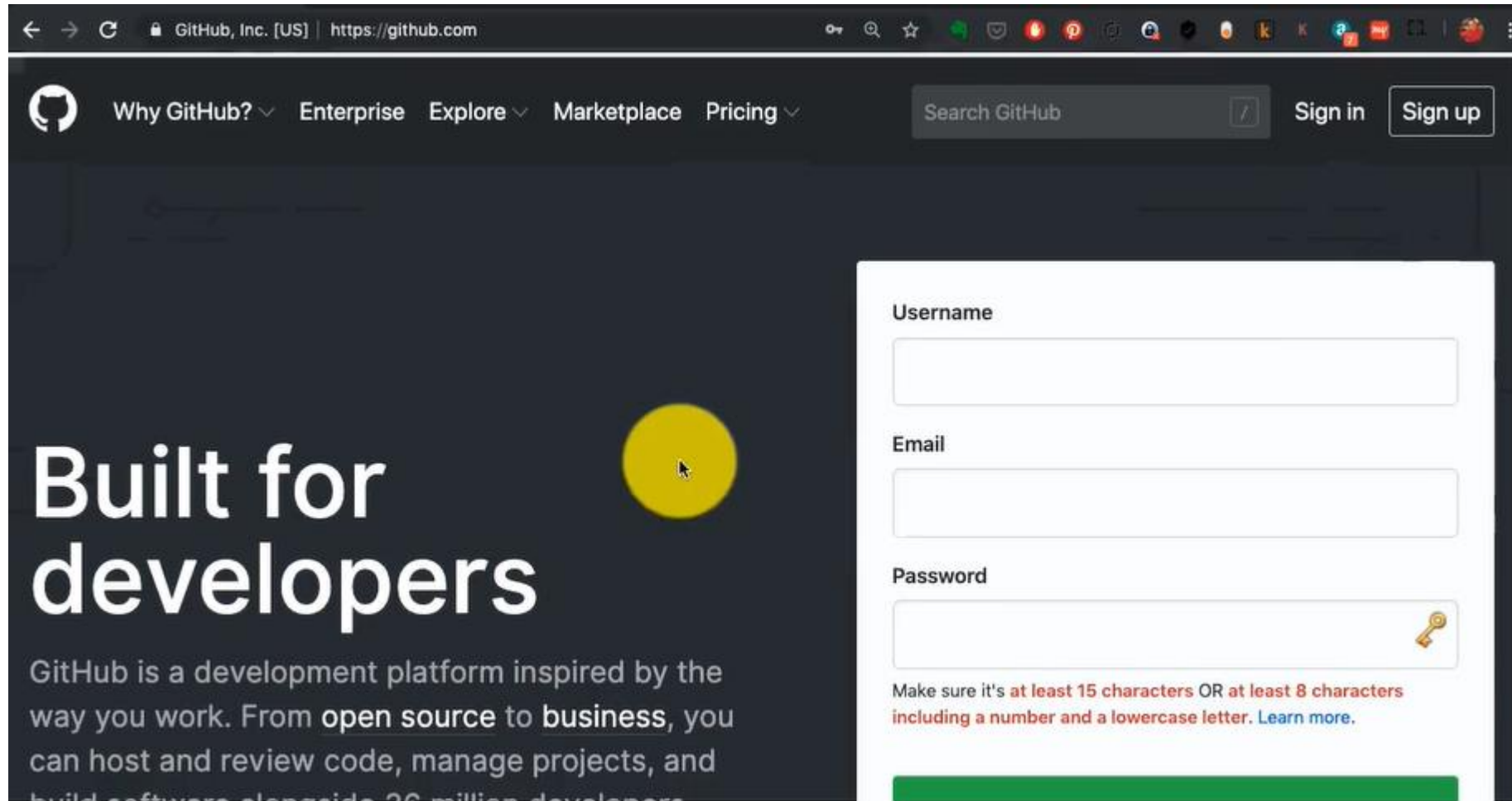
```
$ cat README.md
Habilidades Técnicas
- Programación: C/C++, Perl, Java, Python, R
- Routing & Switching
```

- Agrega el archivo, haz un commit y muestra los logs.

```
dcf737b (HEAD -> master) Agregando inventario de habilidades tecnicas
225701e Regresando libro.txt al contenido original
0941f5d Sigo agregando contenido
c40186a Se edita mensaje.txt y se agregan 2 nuevos archivos
5178cef Seguimos aprendiendo
9e9e512 Se agrega otra linea al archivo mensaje.txt
6e45418 Iniciando notas del primer archivo en git
```

Laboratorio – Conociendo GitHub

- Ve a <https://github.com/>, creen una cuenta usando su cuenta de correo personal.



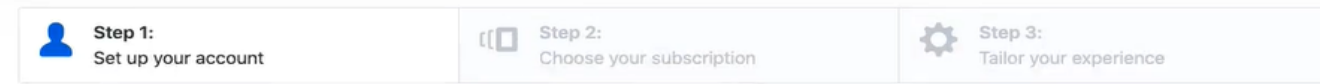
The screenshot shows the GitHub homepage in a web browser. The browser's address bar displays "GitHub, Inc. [US] | https://github.com". The navigation bar includes links for "Why GitHub?", "Enterprise", "Explore", "Marketplace", and "Pricing", along with a "Search GitHub" input field and "Sign in" and "Sign up" buttons. The main content area features the GitHub logo and the text "Built for developers". Below this, it states: "GitHub is a development platform inspired by the way you work. From open source to business, you can host and review code, manage projects, and build software alongside 26 million developers". On the right side, there is a sign-up form with the following fields and instructions:

- Username**: A text input field.
- Email**: A text input field.
- Password**: A text input field with a key icon on the right.
- Instructions**: "Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter. [Learn more.](#)"

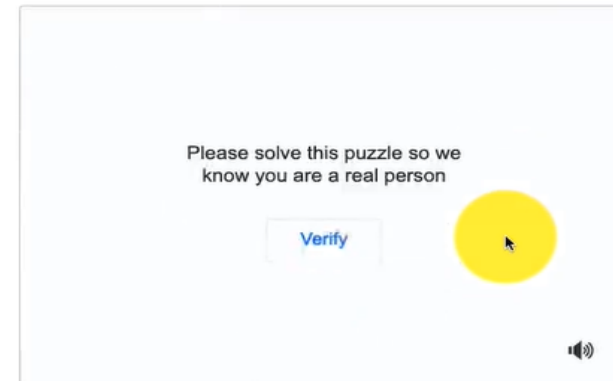
A green button is visible at the bottom of the sign-up form.

Laboratorio – Conociendo GitHub

- Verifica la cuenta, escogiendo la suscripción gratuita, contestando el survey y validando en tu correo.

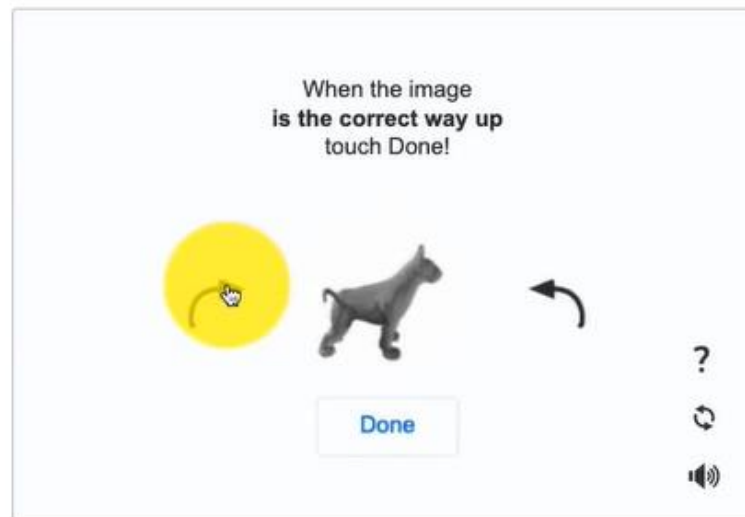


Verify account



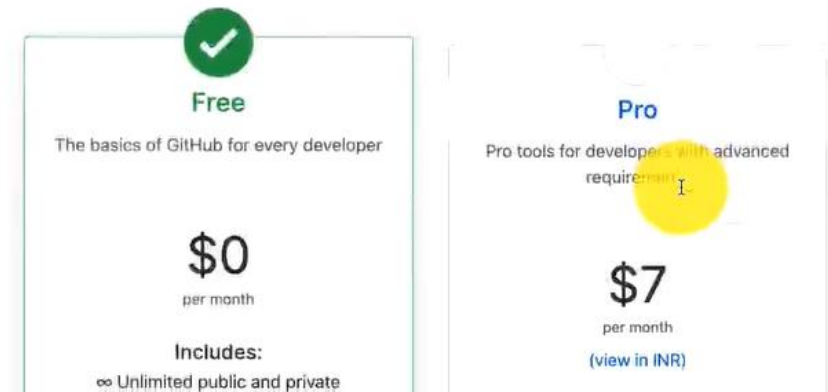
You'll love GitHub

- Unlimited public repositories
- Unlimited private repositories
- ✓ Limitless collaboration
- ✓ Frictionless development
- ✓ Open source community



Choose your subscription

With tools developers love and the world's largest open source community, there's no wrong choice.

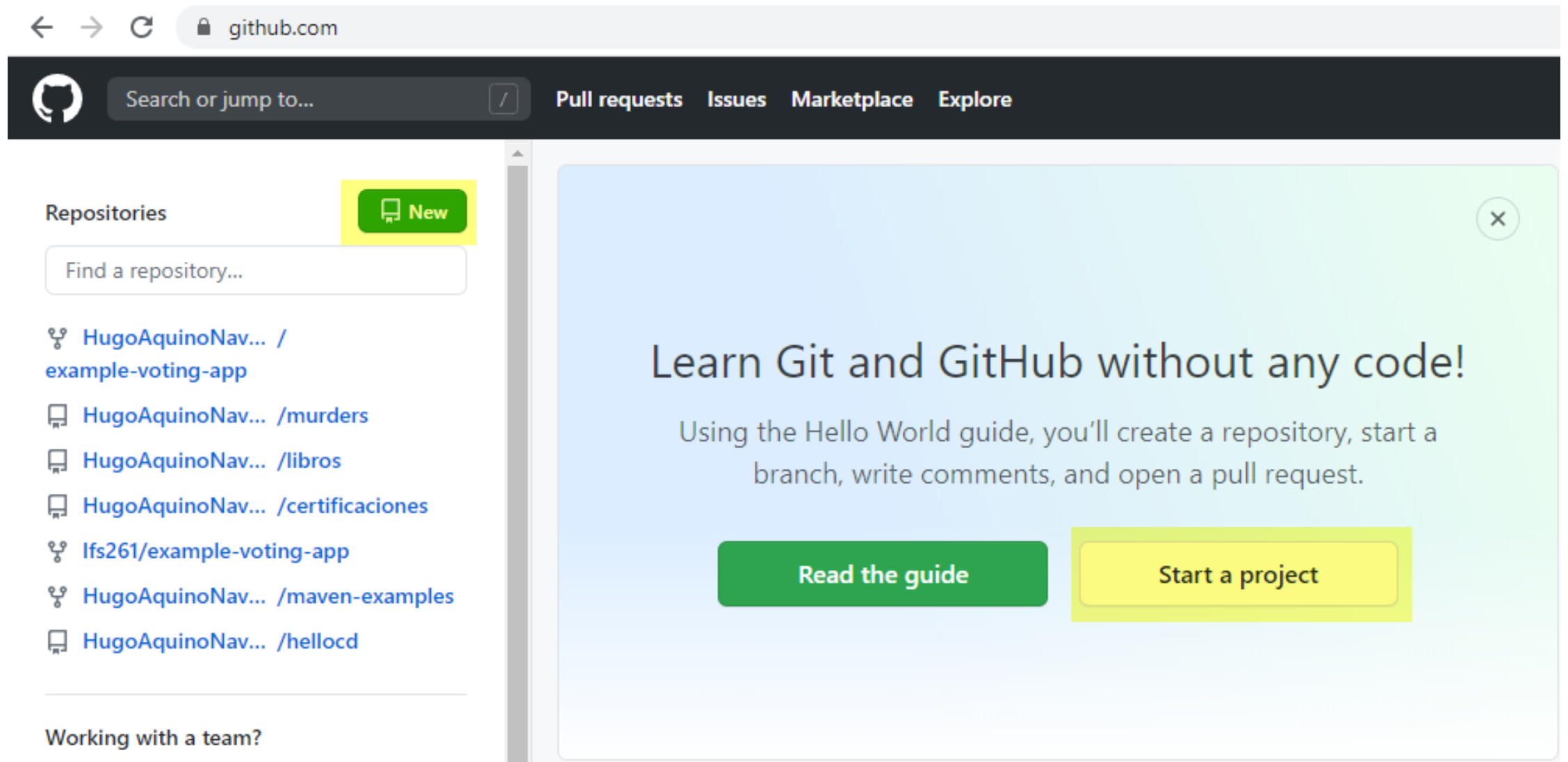


Please verify your email address

Before you can contribute on GitHub, we need you to verify your email address.

Laboratorio – Tu primer repositorio

- Una vez en GitHub, vamos a crear un repositorio



- Que se llame “repositorio”

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?

[Import a repository.](#)

Owner *



HugoAquinoNavarrete ▾

Repository name *

/ repositorio ✓

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

Description (optional)

Repositorio de prueba para laboratorios de Git



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

Laboratorio – Tu primer repositorio

- Vamos a sincronizarlo con el que tenemos en la laptop

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



Set up in Desktop

or

HTTPS

SSH

<https://github.com/HugoAquinoNavarrete/repositorio.git>



Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

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

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



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

```
git remote add origin https://github.com/HugoAquinoNavarrete/repositorio.git
git branch -M master
git push -u origin master
```



...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

Import code

Laboratorio – Tu primer repositorio

- Vamos a sincronizarlo con el que tenemos en la laptop
 - git remote show
 - git remote add origin <https://github.com/HugoAquinoNavarrete/repositorio.git>
 - git branch -M master
 - git push -u origin master
 - git remote show

```
$ git remote show
1 MINGW64 ~/repositorio (master)
$ git remote add origin https://github.com/HugoAquinoNavarrete/repositorio.git
1 MINGW64 ~/repositorio (master)
$ git branch -M master
1 MINGW64 ~/repositorio (master)
$ git push -u origin master
Enumerating objects: 21, done.
Counting objects: 100% (21/21), done.
Delta compression using up to 4 threads
Compressing objects: 100% (15/15), done.
Writing objects: 100% (21/21), 1.94 KiB | 110.00 KiB/s, done.
Total 21 (delta 5), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (5/5), done.
To https://github.com/HugoAquinoNavarrete/repositorio.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.

1 MINGW64 ~/repositorio (master)
$ git remote show
origin

1 MINGW64 ~/repositorio (master)
$ git remote show origin
* remote origin
Fetch URL: https://github.com/HugoAquinoNavarrete/repositorio.git
Push URL: https://github.com/HugoAquinoNavarrete/repositorio.git
HEAD branch: master
Remote branch:
  master tracked
Local branch configured for 'git pull':
  master merges with remote master
Local ref configured for 'git push':
  master pushes to master (up to date)
```



Laboratorio – Tu primer repositorio

- Felicidades has creado tu primer repositorio en Git sincronizado con GitHub

The screenshot shows a web browser window displaying a GitHub repository page. The address bar shows the URL `github.com/HugoAquinoNavarrete/repositorio`. The repository name is `HugoAquinoNavarrete / repositorio`. The navigation bar includes links for `Code`, `Issues`, `Pull requests`, `Actions`, `Projects`, `Wiki`, `Security`, `Insights`, and `Settings`. Below the navigation bar, there are buttons for `master` (selected), `1 branch`, and `0 tags`. To the right are buttons for `Go to file`, `Add file`, and `Code`. The commit history table shows the following entries:

| Commit | Message | Time |
|-------------|---|--------------|
| dcf737b | 1 hour ago | 7 commits |
| Netflix.txt | Se edita mensaje.txt y se agregan 2 nuevos archivos | 12 hours ago |
| README.md | Agregando inventario de habilidades tecnicas | 1 hour ago |
| libro.txt | Regresando libro.txt al contenido original | 1 hour ago |
| mensaje.txt | Se edita mensaje.txt y se agregan 2 nuevos archivos | 12 hours ago |

Below the commit history, the `README.md` file is displayed. The content of the README is:

```
Habilidades Técnicas
```

- Programación: C/C++, Perl, Java, Python, R
- Routing & Switching

Laboratorio – Vamos por más – Clonar repositorio

- Vamos a clonar un repositorio para mostrar el trabajo colaborativo en GitHub y Git.
- En la consola de Git en tu laptop, ve al directorio HOME, crea un directorio que se llame “projects”, clona el repositorio “git_concepts”, ve al directorio “git_concepts” y has un listado recursivo de su contenido.
 - `cd ~`
 - `mkdir projects`
 - `git clone https://github.com/HugoAquinoNavarrete/git_concepts.git`
 - `cd git_concepts`
 - `ls -lR`

```
MINGW64 ~/projects/git_concepts (master)
$ ls -lR
.:
total 5
drwxr-xr-x 1  1049089  0 Sep  5 14:17 cfg/
drwxr-xr-x 1  1049089  0 Sep  5 14:17 img/
-rw-r--r-- 1  1049089 405 Sep  5 14:17 README.md

./cfg:
total 1
-rw-r--r-- 1  1049089 41 Sep  5 14:17 comica.txt

./img:
total 256
-rw-r--r-- 1  1049089 104546 Sep  5 14:17 Branches.png
-rw-r--r-- 1  1049089  79765 Sep  5 14:17 Comandos_git.png
-rw-r--r-- 1  1049089  39841 Sep  5 14:17 GitHub.png
-rw-r--r-- 1  1049089  31182 Sep  5 14:17 Uso_colaborativo_de_Git.jpg
```



Laboratorio – Permisos para hacer commits en el master

- Ve al directorio “cfg” y editar el archivo “comica.txt” eliminando la segunda línea.
 - `cd cfg`
 - `nano comica.txt`
- Agrega el archivo y haz un commit
 - `cd cfg`
 - `nano comica.txt`
 - `git add comica.txt`
 - `git status`
 - `git commit -m “Eliminando segunda línea del archivo comica.txt”`
 - `git push origin master`

```
$ git commit -m "Eliminando segunda línea del archivo comica.txt"
[master 47b9cf1] Eliminando segunda línea del archivo comica.txt
1 file changed, 1 insertion(+), 1 deletion(-)

MINGW64 ~/projects/git_concepts/cfg (master)
$ git push origin master
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 403 bytes | 134.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
remote: error: GH006: Protected branch update failed for refs/heads/master.
remote: error: At least 1 approving review is required by reviewers with write access.
To https://github.com/HugoAquinoNavarrete/git_concepts.git
! [remote rejected] master -> master (protected branch hook declined)
error: failed to push some refs to 'https://github.com/HugoAquinoNavarrete/git_concepts.git'
```


Laboratorio – Permisos para hacer commits en el master

- Si ves los logs aparece que estamos defasados con respecto al Master branch
 - `git log --oneline`
 - `git status`

```
$ git log --oneline
47b9cf1 (HEAD -> master) Eliminando segunda linea del archivo comica.txt
0d26b06 (origin/master, origin/HEAD) Repositorio creado con README.me, imágenes y archivo para el laboratorio
```

```
$ git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
```

- Sincroniza con el remoto
 - `git reset --hard origin/master`
 - `git log --oneline`

```
$ git log --oneline
0d26b06 (HEAD -> master, origin/master, origin/HEAD) Repositorio creado con README.me, imágenes y archivo para el laboratorio
```

- Crea un Branch que se llame “test”
 - `git checkout -b test`

```
MINGW64 ~/projects/git_concepts/cfg (master)
$ git checkout -b test
Switched to a new branch 'test'

MINGW64 ~/projects/git_concepts/cfg (test)
```

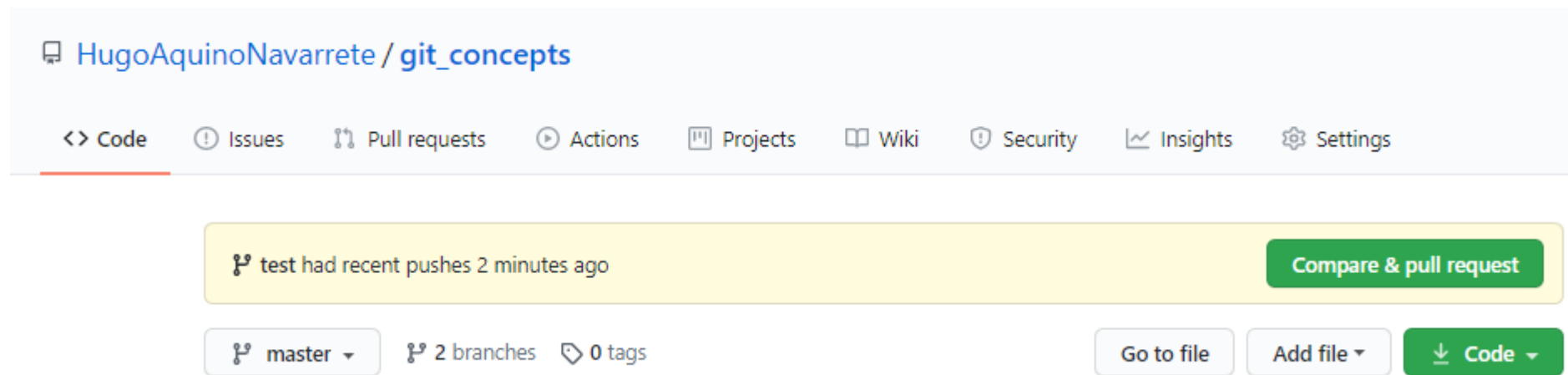
- `git branch`

```
$ git branch
master
* test
```

- Edita el archivo “comica.txt” y agrega la información que se solicita, agrégalo, haz un commit así como un push al repositorio en GitHub
 - `nano comica.txt`
 - `git commit -am "Agregando otra comica"`
 - `git push origin test`

```
$ git push origin test
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 420 bytes | 38.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'test' on GitHub by visiting:
remote:   https://github.com/HugoAquinoNavarrete/git_concepts/pull/new/test
remote:
To https://github.com/HugoAquinoNavarrete/git_concepts.git
 * [new branch]      test -> test
```

- Mira en GitHub lo que aparece



- El repositorio está configurado para permitir otros branches diferentes a Master

- Regla de protección del Branch

| | |
|---------------------|---|
| Options | Branch protection rule |
| Manage access | |
| Security & analysis | |
| Branches | Branch name pattern |
| Webhooks | master |
| Notifications | Applies to 1 branch |
| Integrations | master |
| Deploy keys | |
| Secrets | |
| Actions | |
| Moderation | Protect matching branches |
| Interaction limits | <input checked="" type="checkbox"/> Require pull request reviews before merging When enabled, all commits must be made to a non-protected branch and submitted via a pull request with the required number of approving reviews and no changes requested before it can be merged into a branch that matches this rule. Required approving reviews: 1 |
| | <input checked="" type="checkbox"/> Dismiss stale pull request approvals when new commits are pushed New reviewable commits pushed to a matching branch will dismiss pull request review approvals. |
| | <input type="checkbox"/> Require review from Code Owners Require an approved review in pull requests including files with a designated code owner. |
| | <input type="checkbox"/> Require status checks to pass before merging Choose which status checks must pass before branches can be merged into a branch that matches this rule. When enabled, commits must first be pushed to another branch, then merged or pushed directly to a branch that matches this rule after status checks have passed. |
| | <input type="checkbox"/> Require signed commits Commits pushed to matching branches must have verified signatures. |
| | <input type="checkbox"/> Require linear history Prevent merge commits from being pushed to matching branches. |
| | <input checked="" type="checkbox"/> Include administrators Enforce all configured restrictions above for administrators. |

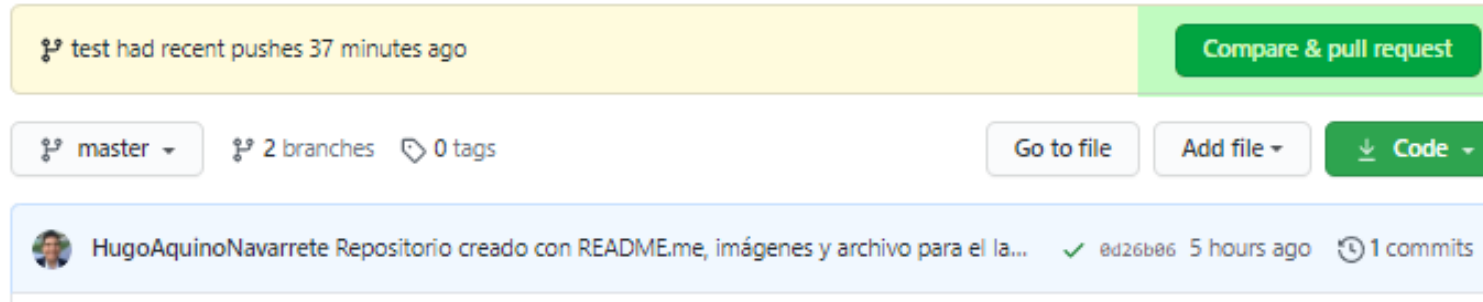
- Definir accesos

The screenshot shows the GitHub 'Manage access' interface. On the left is a sidebar with navigation links: Options, Manage access (highlighted), Security & analysis, Branches, Webhooks, Notifications, Integrations, Deploy keys, Secrets, Actions, Moderation, and Interaction limits. The main content area is titled 'Who has access' and shows two tabs: 'PUBLIC REPOSITORY' (selected) and 'DIRECT ACCESS'. The 'PUBLIC REPOSITORY' tab indicates the repository is public and visible to anyone, with a 'Manage' link. The 'DIRECT ACCESS' tab shows '1 has access to this repository. 1 invitation.' Below this is the 'Manage access' section, which includes a 'Select all' checkbox, a search bar 'Find a collaborator...', and a list of collaborators. One collaborator, 'HugoAquinoNavarrete-1977', is listed with a 'Pending Invite' status and a trash icon. A green button 'Invite a collaborator' is located at the top right of the 'Manage access' section.

The screenshot displays a GitHub invitation email and the corresponding invitation page. The email, from GitHub, states: '@HugoAquinoNavarrete has invited you to collaborate on the HugoAquinoNavarrete/git_concepts repository'. It provides instructions on how to accept or decline the invitation and includes a 'View invitation' button. The invitation page shows the GitHub logo, the repository name 'HugoAquinoNavarrete/git_concepts', and the invitation status 'HugoAquinoNavarrete invited you to collaborate'. It features 'Accept invitation' and 'Decline' buttons. Below these buttons, it lists the permissions granted to collaborators: 'Owners of git_concepts will be able to see: Your public profile information, Certain activity within this repository, Country of request origin, Your access level for this repository, and Your IP address'. At the bottom, there is a link to report spam or malicious content: 'Is this user sending spam or malicious content? Block HugoAquinoNavarrete'.

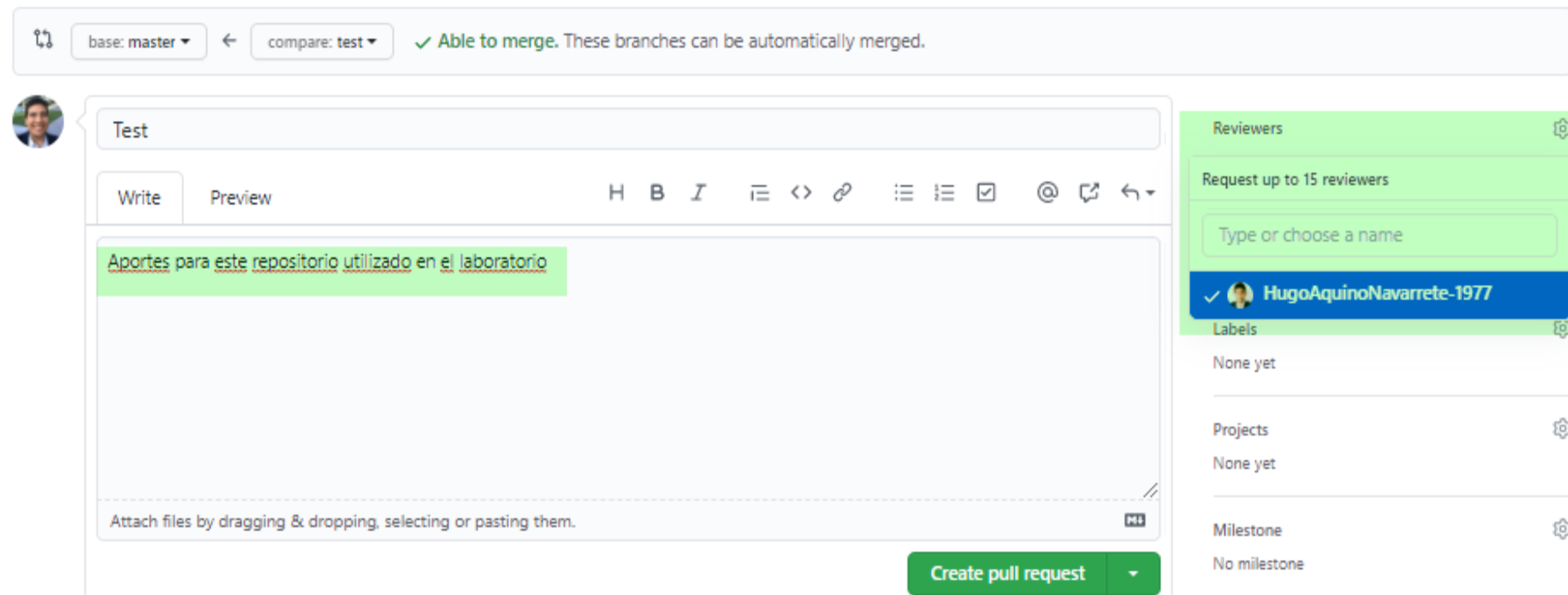
Laboratorio – Pull request

- Desde el Branch “Test” se creará un “Pull request”



Open a pull request


Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).



Laboratorio – Pull request


- Así se vería para el creador del repositorio clonado

[Open](#) **Test #1**
HugoAquinoNavarrete wants to merge 3 commits into `master` from `test`







HugoAquinoNavarrete commented now
Owner

Aportes para este repositorio utilizado en el laboratorio




HugoAquinoNavarrete added 3 commits 3 hours ago


-  Agregando otra comica 28e2f0d
-  Acomodando archivo README.md e2b45b4
-  Acomodando nuevamente el archivo README.md a2513cd





HugoAquinoNavarrete requested a review from **HugoAquinoNavarrete-1977** now


Add more commits by pushing to the `test` branch on **HugoAquinoNavarrete/git_concepts**.

**This branch has not been deployed**
No deployments



**Review required**
At least 1 approving review is required by reviewers with write access. [Learn more.](#) [Show all reviewers](#)


 1 pending reviewer

**Merging is blocked**
Merging can be performed automatically with 1 approving review.

As an administrator, you may still merge this pull request.

[Merge pull request](#) You can also [open this in GitHub Desktop](#) or view [command line instructions](#).

Reviewers

 **HugoAquinoNava...**

At least 1 approving review is required to merge this pull request.

Still in progress? [Convert to draft](#)

Assignees

No one—assign yourself

Labels

None yet

Projects

None yet

Milestone


No milestone

Linked issues

Successfully merging this pull request may close these issues.


None yet

Notifications [Customize](#)

 [Unsubscribe](#)

You're receiving notifications because you're watching this repository.

1 participant

Copa Airlines 

- Para aprobar los cambios

HugoAquinoNavarrete / git_concepts

Watch 1

Code Issues Pull requests 1 Actions Projects Wiki Security Insights

HugoAquinoNavarrete requested your review on this pull request.

Add your review

Test #1

Edit Open with

Open HugoAquinoNav... wants to merge 3 commits into master from test

Conversation 0 Commits 3 Checks 0 Files changed 2

+3 -1



HugoAquinoNavarrete commented 6 minutes ago

Owner

Aportes para este repositorio utilizado en el laboratorio



HugoAquinoNavarrete added 3 commits 3 hours ago



Agregando otra comica

28e2f8d



Acomodando archivo README.md

e2b45b4



Acomodando nuevamente el archivo README.md

a2513cd



HugoAquinoNavarrete requested a review from HugoAquinoNavarrete-1977 6 minutes ago

Reviewers



HugoAquinoNava...



At least 1 approving review is required to merge this pull request.

Still in progress? Convert to draft

Assignees



No one—assign yourself

Labels



None yet

Projects

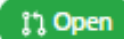


None yet

Add more commits by pushing to the test branch on HugoAquinoNavarrete/git_concepts.

- Para aprobar los cambios

Test #1



HugoAquinoNav... wants to merge 3 commits into `master` from `test`

Edit

Open with ▾



Conversation 0



Commits 3



Checks 0



Files changed 2

+3 -1

Changes from all commits ▾ File filter... ▾ Jump to... ▾ ⚙ ▾

0 / 2 files viewed ⓘ

Review changes ▾

3 README.md

```
... @@ -1,4 +1,5 @@
1 1 Repositorio del Laboratorio de Git
2 2 - En el directorio "img" están algunas imágenes de los coman
3 3 - En el directorio "doc" está el documento de la presentació
4 - - En el directorio "cfg" se encuentra el archivo "comica.txt
   - tenías cuando la veías>
4 + - En el directorio "cfg" se encuentra el archivo "comica.txt
5 + - nombre,cómica o caricatura preferida,edad que tenías c
```

1 cfg/comica.txt

```
... @@ -1,2 +1,3 @@
1 1 nombre,comica,edad
2 2 Hugo,Thundercats,10
3 + Hugo,El pajaro loco,8
```

Write

Preview

H B I ≡ <> 🔗 ≡ ≡ ☑ @ ↻ ▾

OK, aprobado

Attach files by dragging & dropping, selecting or pasting them. 📎

☐ Comment

Submit general feedback without explicit approval.

☒ Approve

Submit feedback and approve merging these changes.

☐ Request changes

Submit feedback that must be addressed before merging.

Submit review

- Para aprobar los cambios

The screenshot displays a GitHub Pull Request (PR) interface. At the top, a green 'Open' button is next to the title 'Test #1'. Below the title, it states 'HugoAquinoNavarrete wants to merge 3 commits into master from test'. A commit history shows two commits: 'Acomodando archivo README.md' (commit e2b45b4) and 'Acomodando nuevamente el archivo README.md' (commit a2513cd). A review section shows that 'HugoAquinoNavarrete-1977' approved the changes. A comment box shows a comment 'OK, aprobado' by 'HugoAquinoNavarrete-1977'. Below the review, a message states 'Add more commits by pushing to the test branch on HugoAquinoNavarrete/git_concepts.' A box indicates 'This branch has not been deployed' with 'No deployments'. A green box highlights the 'Changes approved' status, showing '1 approving review by reviewers with write access.' and '1 approval'. It also states 'This branch has no conflicts with the base branch' and 'Merging can be performed automatically.' A green button 'Merge pull request' is visible. At the bottom, a purple box indicates 'Pull request successfully merged and closed' and 'You're all set—the test branch can be safely deleted.' with a 'Delete branch' button.

Test #1
HugoAquinoNavarrete wants to merge 3 commits into master from test

Acomodando archivo README.md e2b45b4
Acomodando nuevamente el archivo README.md a2513cd

HugoAquinoNavarrete requested a review from HugoAquinoNavarrete-1977 7 minutes ago

HugoAquinoNavarrete-1977 approved these changes now [View changes](#)

HugoAquinoNavarrete-1977 left a comment [Collaborator](#) [🗨️](#) [⋮](#)

OK, aprobado

Add more commits by pushing to the test branch on HugoAquinoNavarrete/git_concepts.

[🚀](#) This branch has not been deployed
No deployments

[🔗](#) **Changes approved** [Show all reviewers](#)
1 approving review by reviewers with write access. [Learn more.](#)

✓ 1 approval

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

[Merge pull request](#) You can also open this in [GitHub Desktop](#) or [view command line instructions.](#)

[🔗](#) **Pull request successfully merged and closed** [Delete branch](#)
You're all set—the test branch can be safely deleted.

Laboratorio – Sincroniza los cambios

- En tu repositorio local, verifica en que Branch te encuentras
 - `git branch`

```
$ git branch
master
* test
```

- Elimina el repositorio “test”
 - `git branch -D test`

```
$ git branch -D test
Deleted branch test (was 56e86e9).
```

- Sincroniza el repositorio local con el branch “Master”
 - `git pull origin master`

```
MINGW64 ~/projects/git_concepts (master)
$ git pull origin master
From https://github.com/HugoAquinoNavarrete/git_concepts
 * branch                master          -> FETCH_HEAD
Updating 0d26b06..56e86e9
Fast-forward
 README.md               | 3 ++-
 cfg/comica.txt          | 1 +
 2 files changed, 3 insertions(+), 1 deletion(-)
```

- Tiempo para repasar jugando



Gracias

