

**Guía GitHub**

# Sistema de Control de Versiones

Te permiten llevar una organización adecuada de tus proyectos, permitiendo compartir con tu equipo de trabajo las versiones de tu código

- CVS
- SVN
- Mercurial
- **Github:**



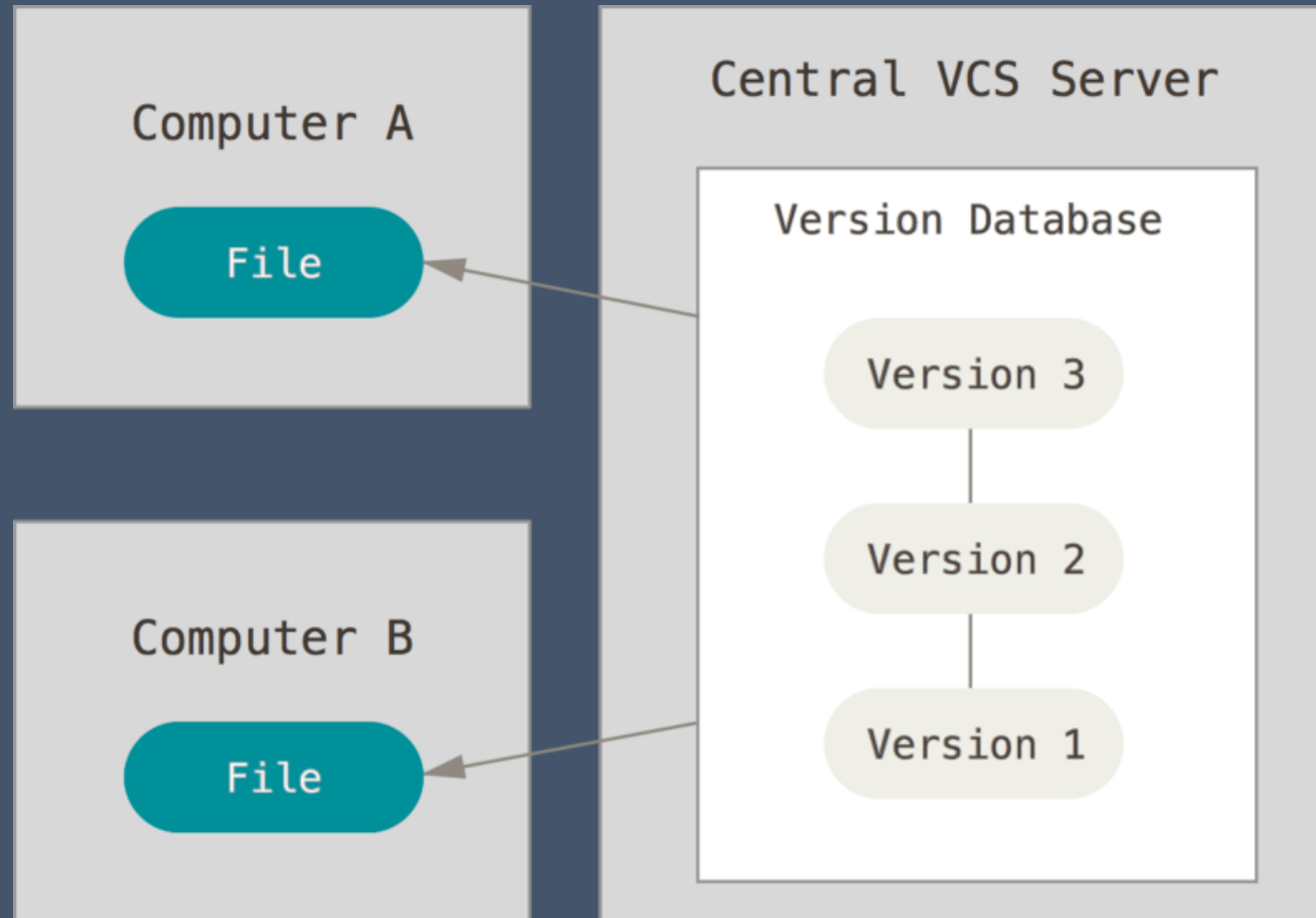
# Sistema de Control de Versiones

## Ejemplos

- Archivo01.c
- **Archivofinal01.c**
- **Archivofinalfinal2.c**
- **Archivofinalcasiterminado.c**



# Sistema de Control de Versiones



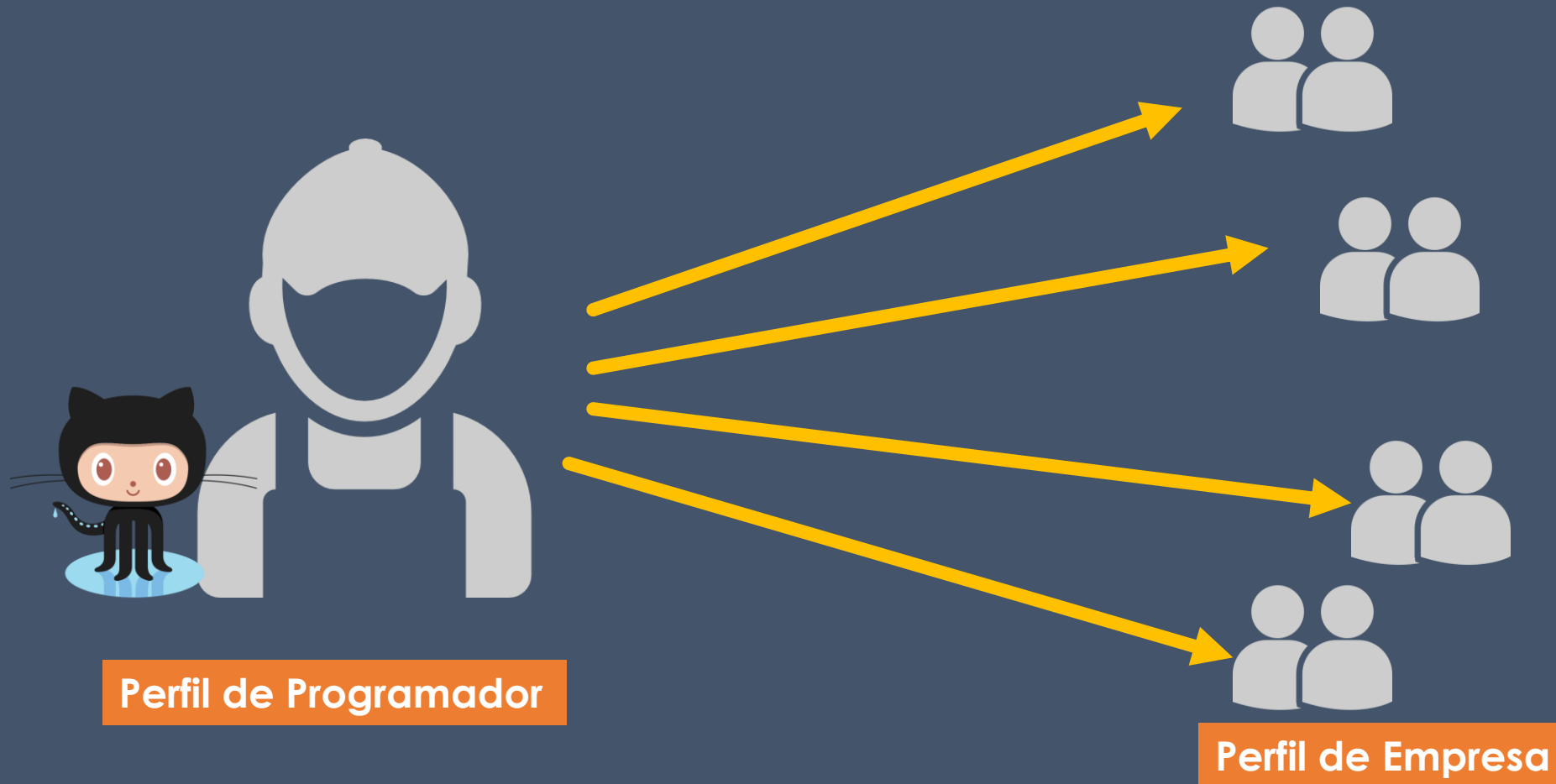
# Control de Versiones

- ❑ Versión de paga
- ❑ Versión gratuita

<https://github.com/>



# ¿github?



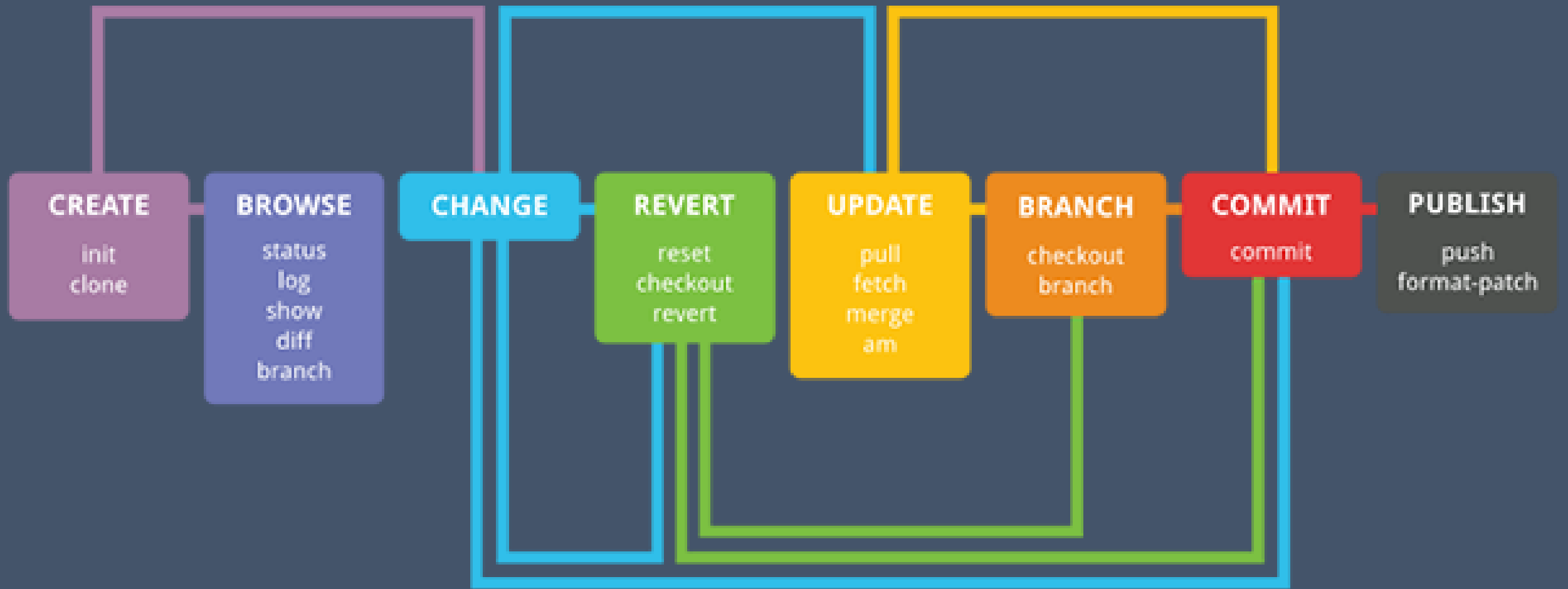
# Github: Conectar nuestra pc a github



- ❑ <https://git-scm.com/downloads>
- ❑ Abrir github bash



# Funcionamiento de general git

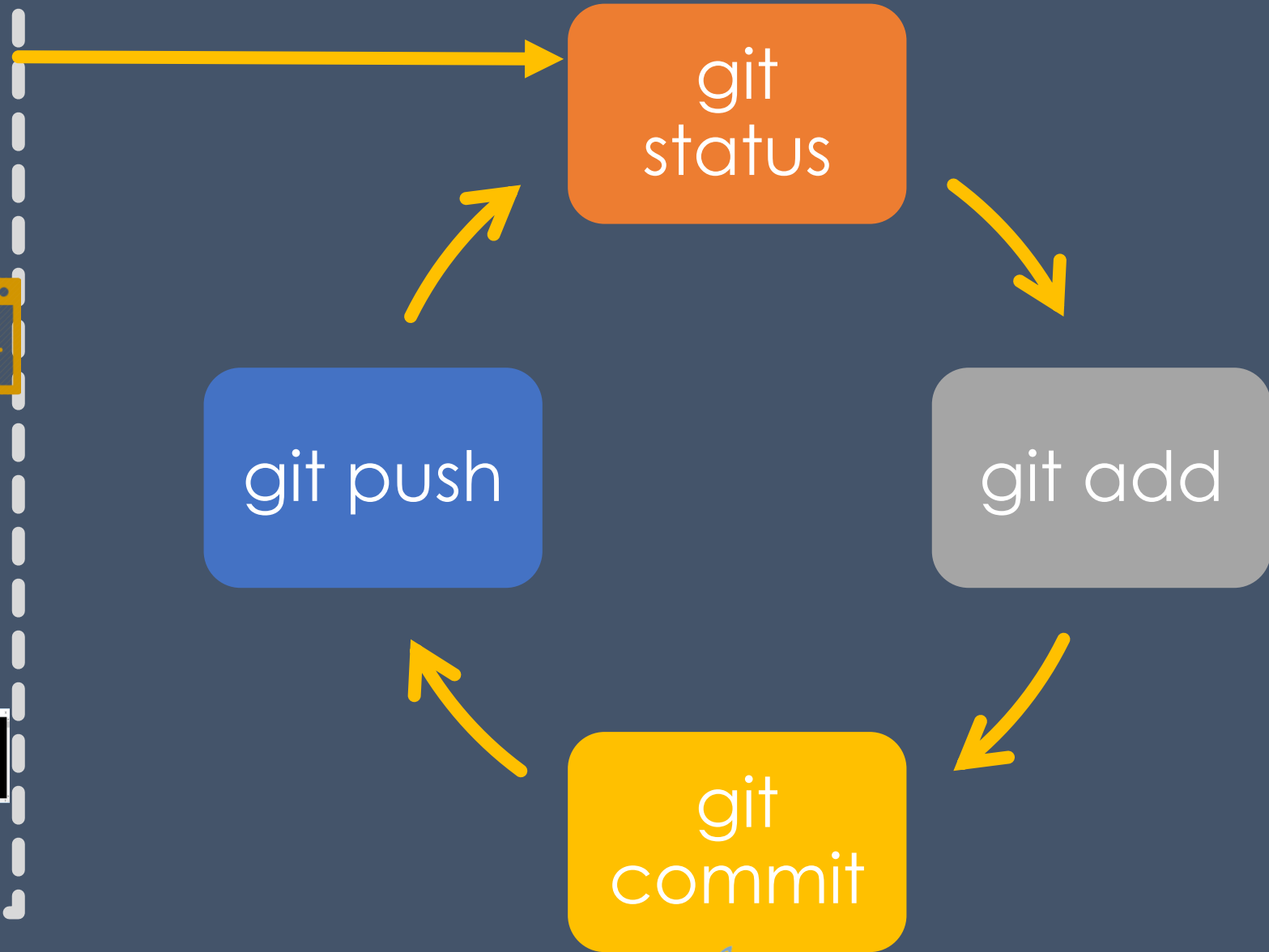
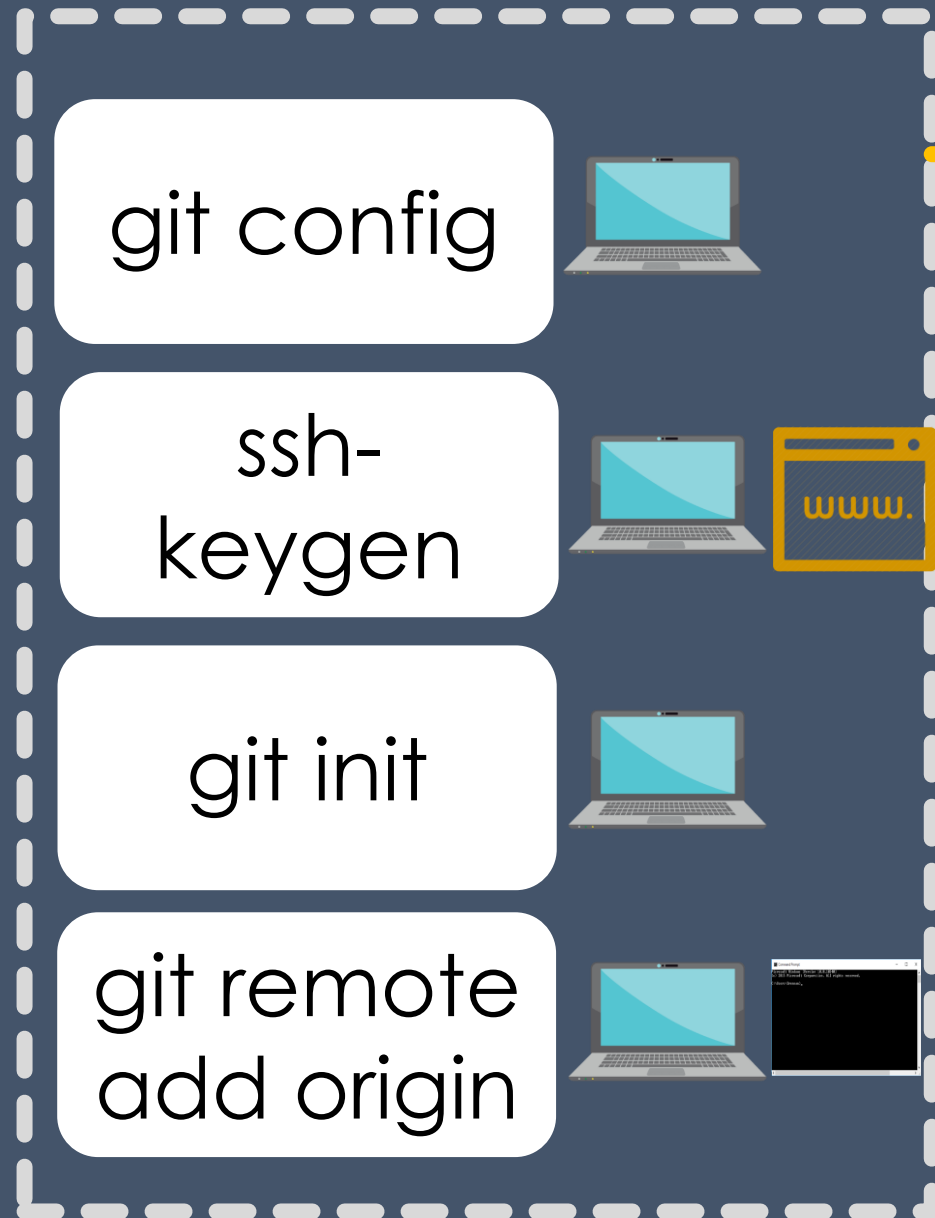


**Recurso Adicional:** [Libro en línea](#)



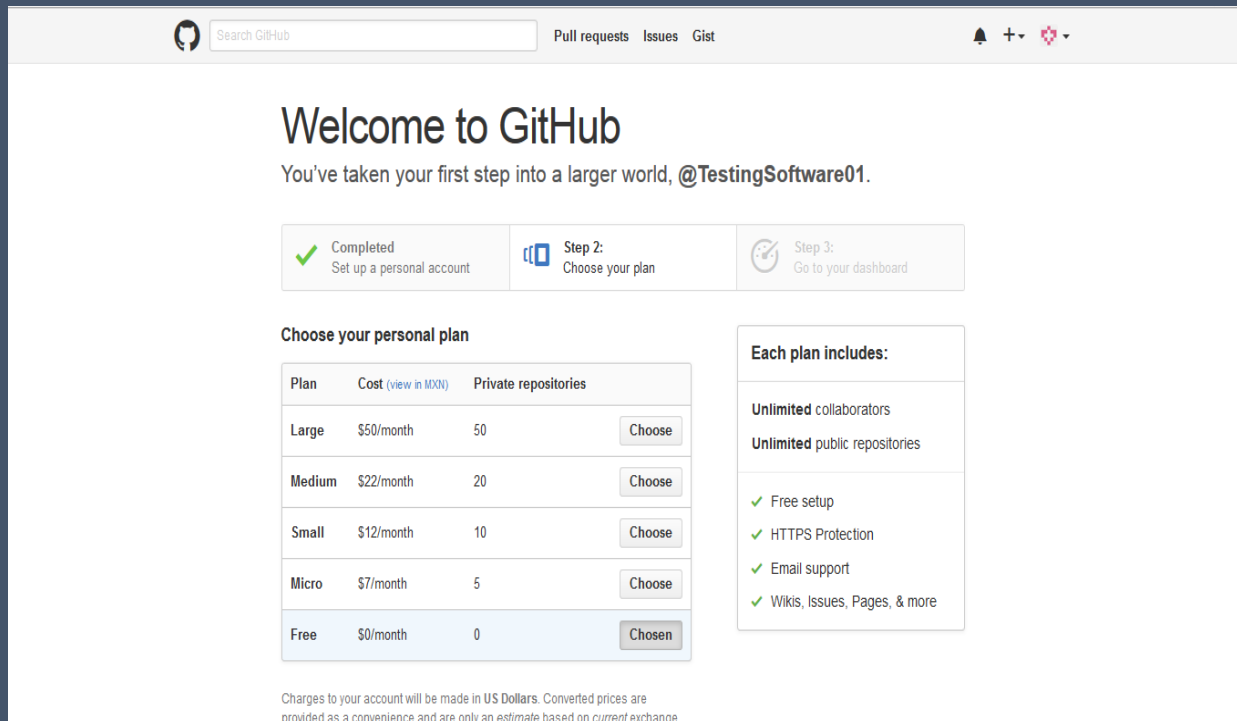


# Funcionamiento básico git



# Github:Cuenta

Crear una cuenta en el sitio de [github.com](https://github.com) y confirmar



The screenshot shows the GitHub sign-up process at the 'Choose your plan' step. The user is logged in as @TestingSoftware01. The page displays a table of plans and a list of features included in each plan.

**Welcome to GitHub**  
You've taken your first step into a larger world, @TestingSoftware01.

Progress: Step 1: Completed (Set up a personal account) | **Step 2: Choose your plan** | Step 3: Go to your dashboard

**Choose your personal plan**

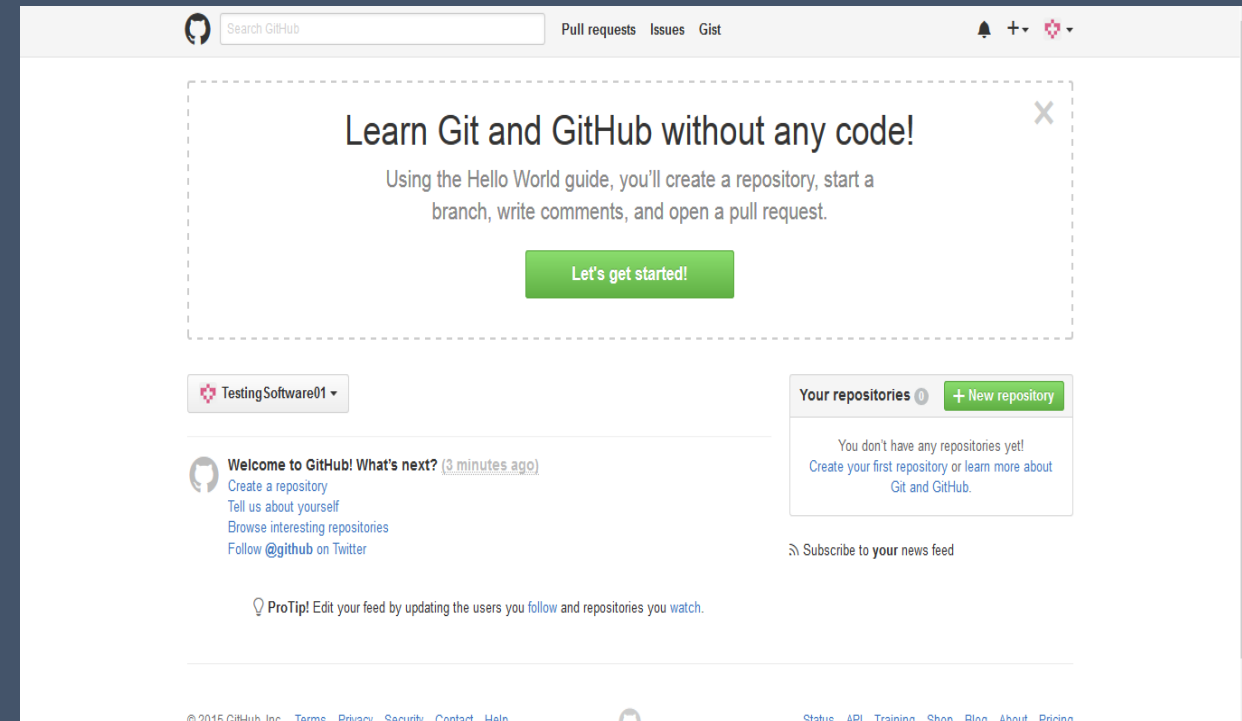
Plan	Cost (view in MDN)	Private repositories	
Large	\$50/month	50	Choose
Medium	\$22/month	20	Choose
Small	\$12/month	10	Choose
Micro	\$7/month	5	Choose
Free	\$0/month	0	Chosen

**Each plan includes:**

- Unlimited collaborators
- Unlimited public repositories
- Free setup
- HTTPS Protection
- Email support
- Wikis, Issues, Pages, & more

Charges to your account will be made in US Dollars. Converted prices are provided as a convenience and are only an estimate based on current exchange

Seleccionar el plan y dar en sign up



The screenshot shows the GitHub dashboard for user @TestingSoftware01. It includes a welcome message, a 'Let's get started!' button, and a section for repositories.

**Learn Git and GitHub without any code!**  
Using the Hello World guide, you'll create a repository, start a branch, write comments, and open a pull request.  
[Let's get started!](#)

**TestingSoftware01**

**Welcome to GitHub! What's next? (3 minutes ago)**  
[Create a repository](#)  
[Tell us about yourself](#)  
[Browse interesting repositories](#)  
[Follow @github on Twitter](#)

**Your repositories** [+ New repository](#)

You don't have any repositories yet!  
[Create your first repository or learn more about Git and GitHub.](#)

[Subscribe to your news feed](#)

**ProTip!** Edit your feed by updating the users you [follow](#) and repositories you [watch](#).

© 2015 GitHub, Inc. [Terms](#) [Privacy](#) [Security](#) [Contact](#) [Help](#) [Status](#) [API](#) [Training](#) [Shop](#) [Blog](#) [About](#) [Pricing](#)


Panel de control principal





# Github: Configurar

## Crear organización o empresa

### Create an organization

 **Completed**  
Set up a personal account

 **Step 2:**  
Set up the organization

 **Step 3:**  
Invite team members

#### Set up the organization

##### Organization name

codingtheworld ✓

The organization will live at <https://github.com/codingtheworld>

##### Billing email

testingsoftware@hotmail.com

Receipts will be sent here

#### Choose the organization's plan



Plan	Cost (view in MXN)	Private repos	
Diamond	\$450/month	300	<button>Choose</button>
Platinum	\$200/month	125	<button>Choose</button>
Gold	\$100/month	50	<button>Choose</button>

#### Organizations


- ✓ Repository management
- ✓ Fine-grained permissions
- ✓ Focused dashboard


The credit card and plan you choose on this screen will be billed to the organization — not your user account (TestingSoftware01).


#### Managed by owners

On the next screen you'll be able to grant administrative access to other GitHub users. These people will be able to manage every aspect of the organization (billing, repositories, teams, etc).

### Invite team members

 **Completed**  
Set up a personal account

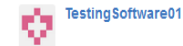
 **Completed**  
Set up the organization

 **Step 3:**  
Invite team members

#### Invite people to the codingtheworld Owners Team

Search by username, full name or email address

lmonge@uabc.edu.mx



Finish

#### Owners

- ✓ Create repositories
- ✓ Organize into teams
- ✓ Review code
- ✓ Communicate via @mentions

Owners have full access to all of the organization's repositories and have **admin rights** to the organization.

Owners can change billing info and [cancel organization plans](#).

[Learn more](#)



luis.monge@uabc.edu.mx



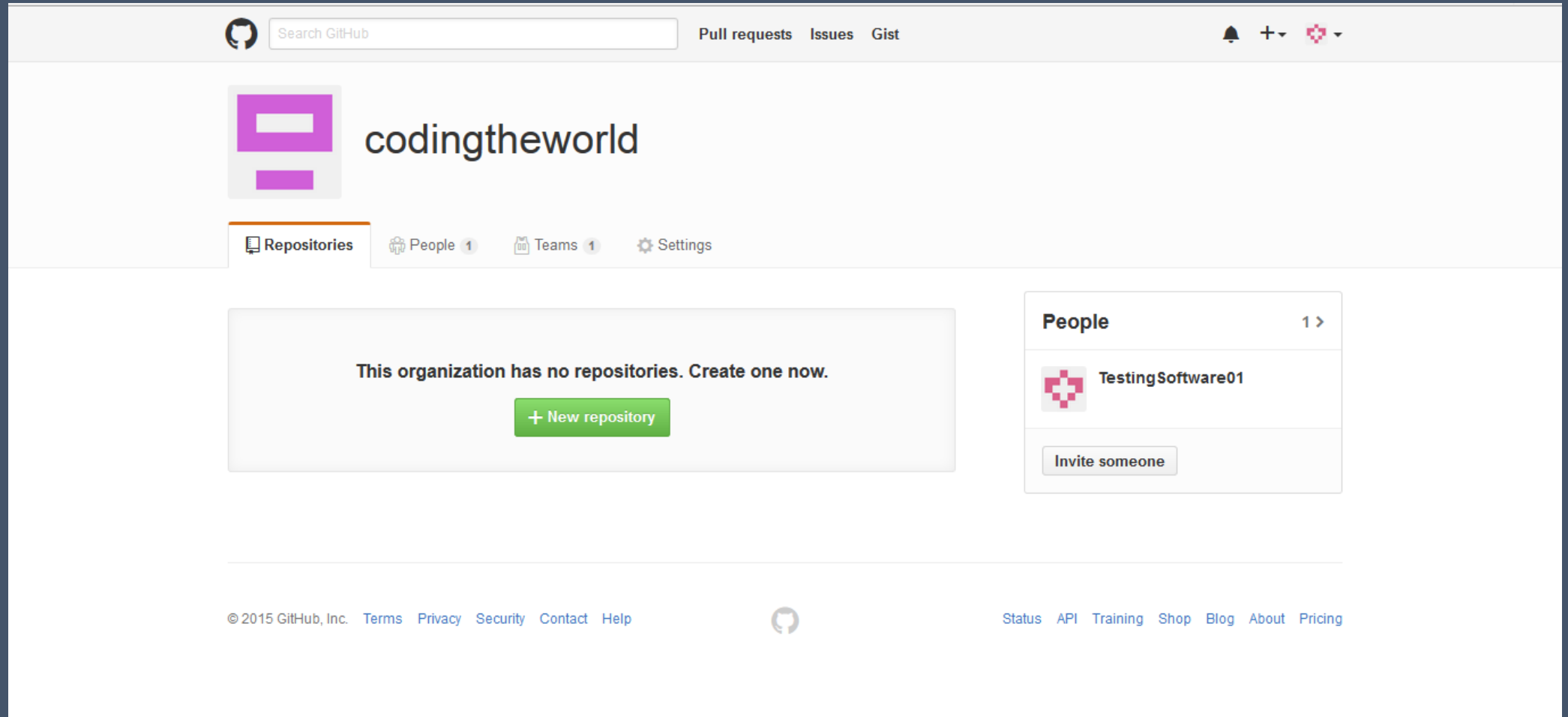
<http://www.luismonge.com.mx>



<http://www.ingenioteka.com>

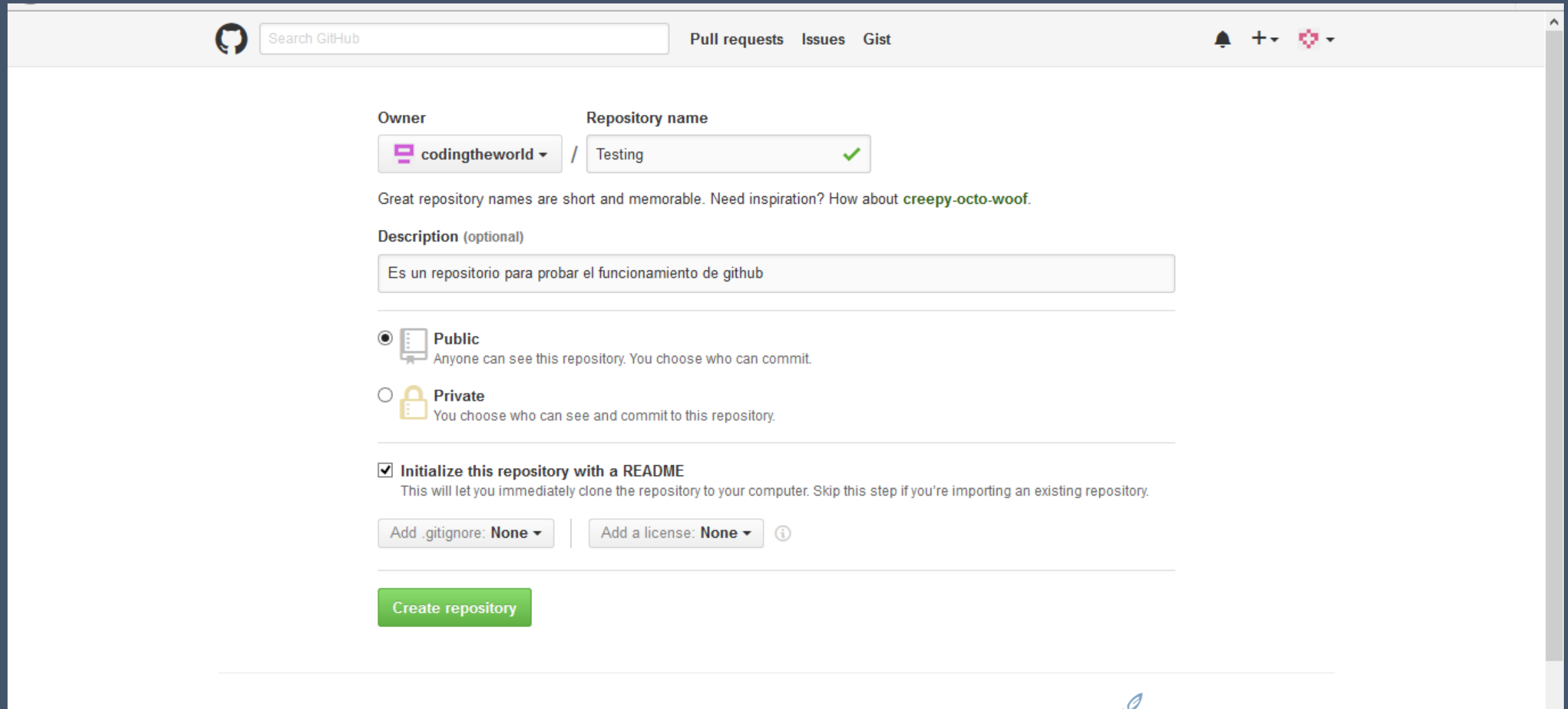
# Github: Configurar

Ahora tenemos un perfil empresarial para un equipo



# Github: Repositorio

Crear y configurar  
repositorio ¿Para qué perfil?



The screenshot shows the GitHub 'Create new repository' page. At the top, there's a navigation bar with the GitHub logo, a search bar, and links for 'Pull requests', 'Issues', and 'Gist'. On the right, there are icons for notifications, a dropdown menu, and a repository icon. The main form has two columns: 'Owner' and 'Repository name'. The 'Owner' dropdown is set to 'codingtheworld'. The 'Repository name' field contains 'Testing' with a green checkmark. Below this, a message says: 'Great repository names are short and memorable. Need inspiration? How about **creepy-octo-woof**.' The 'Description (optional)' field contains the text 'Es un repositorio para probar el funcionamiento de github'. Under the 'Visibility' section, 'Public' is selected with a radio button, and 'Private' is unselected. Below this, the 'Initialize this repository with a README' checkbox is checked. At the bottom, there are dropdowns for '.gitignore' (set to 'None') and 'Add a license' (set to 'None'). A green 'Create repository' button is at the bottom.

Owner: codingtheworld / Repository name: Testing ✓

Great repository names are short and memorable. Need inspiration? How about **creepy-octo-woof**.

Description (optional): Es un repositorio para probar el funcionamiento de github

☒ **Public**  
Anyone can see this repository. You choose who can commit.

☐ **Private**  
You choose who can see and commit to this repository.

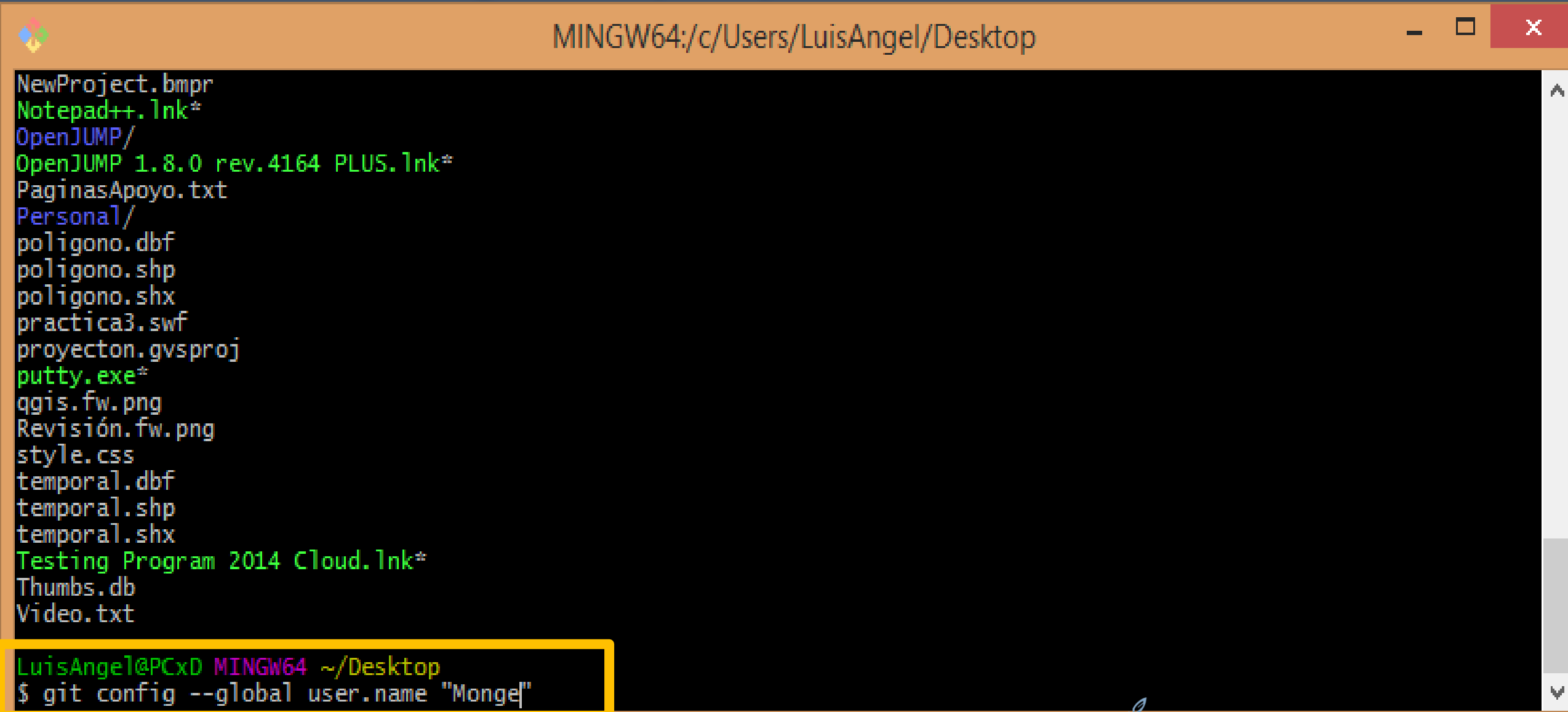
☒ **Initialize this repository with a README**  
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: None | Add a license: None ⓘ

**Create repository**



# Github: Configurar propietario




```
MINGW64:/c/Users/LuisAngel/Desktop

NewProject.bmpr
Notepad++.lnk*
OpenJUMP/
OpenJUMP 1.8.0 rev.4164 PLUS.lnk*
PaginasApoyo.txt
Personal/
poligono.dbf
poligono.shp
poligono.shx
practica3.swf
proyector.gvsproj
putty.exe*
qgis.fw.png
Revisión.fw.png
style.css
temporal.dbf
temporal.shp
temporal.shx
Testing Program 2014 Cloud.lnk*
Thumbs.db
Video.txt

LuisAngel@PCxD MINGW64 ~/Desktop
$ git config --global user.name "Monge"
```



# Github: Configurar correo



A terminal window titled "MINGW64:/c/Users/LuisAngel/Desktop" with standard window controls. The terminal displays a directory listing of files on the desktop, including GIS files, a PDF, a CSS file, a database, a SWF file, a project file, a text file, and a video file. Below the listing, the user runs two git configuration commands to set their global user name and email. The second command is highlighted with a yellow box. The prompt is "LuisAngel@PCxD MINGW64 ~/Desktop".

```
poligono.dbf
poligono.shp
poligono.shx
practica3.swf
proyector.gvsproj
putty.exe*
qgis.fw.png
Revisión.fw.png
style.css
temporal.dbf
temporal.shp
temporal.shx
Testing Program 2014 Cloud.lnk*
Thumbs.db
Video.txt

LuisAngel@PCxD MINGW64 ~/Desktop
$ git config --global user.name "Monge"

LuisAngel@PCxD MINGW64 ~/Desktop
$ git config --global user.email "testingsoftware@hotmail.com"

LuisAngel@PCxD MINGW64 ~/Desktop
$
```



# Github: Generar llave única

```
MINGW64:/c/Users/LuisAngel/Desktop
qgis.fw.png
Revisión.fw.png
style.css
temporal.dbf
temporal.shp
temporal.shx
Testing Program 2014 Cloud.lnk*
Thumbs.db
Video.txt

LuisAngel@PCxD MINGW64 ~/Desktop
$ git config --global user.name "Monge"

LuisAngel@PCxD MINGW64 ~/Desktop
$ git config --global user.email "testingsoftware@hotmail.com"

LuisAngel@PCxD MINGW64 ~/Desktop
$ ssh-keygen
Generating public/private rsa key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/LuisAngel/.ssh/id_rsa.
Your public key has been saved in /c/Users/LuisAngel/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:CsLRRcFgOFPGuJq69G55gUleugDPSnf5FpBB9Gmp5bs LuisAngel@PCxD
The key's randomart image is:
+---[RSA 2048]---+
|  o+o          |
|  . =. o       |
|  . = +*       |
| +* +=o        |
| *+B.O.. S     |
| *o@ *.        |
| ooo *.o       |
| = .+ ..       |
| =ooo.E        |
+---[SHA256]-----+

LuisAngel@PCxD MINGW64 ~/Desktop
$
```





# Github: Generar llave única

```
MINGW64:/c/Users/LuisAngel/Desktop
OpenJUMP/
OpenJUMP 1.8.0 rev.4164 PLUS.lnk*
PaginasApoyo.txt
Personal/
poligono.dbf
poligono.shp
poligono.shx
practica3.swf
proyectoron.gvsproj
putty.exe*
qgis.fw.png
Revisión.fw.png
style.css
temporal.dbf
temporal.shp
temporal.shx
Testing Program 2014 Cloud.lnk*
Thumbs.db
Video.txt

LuisAngel@PCxD MINGW64 ~/Desktop
$ git config --global user.name "Monge"

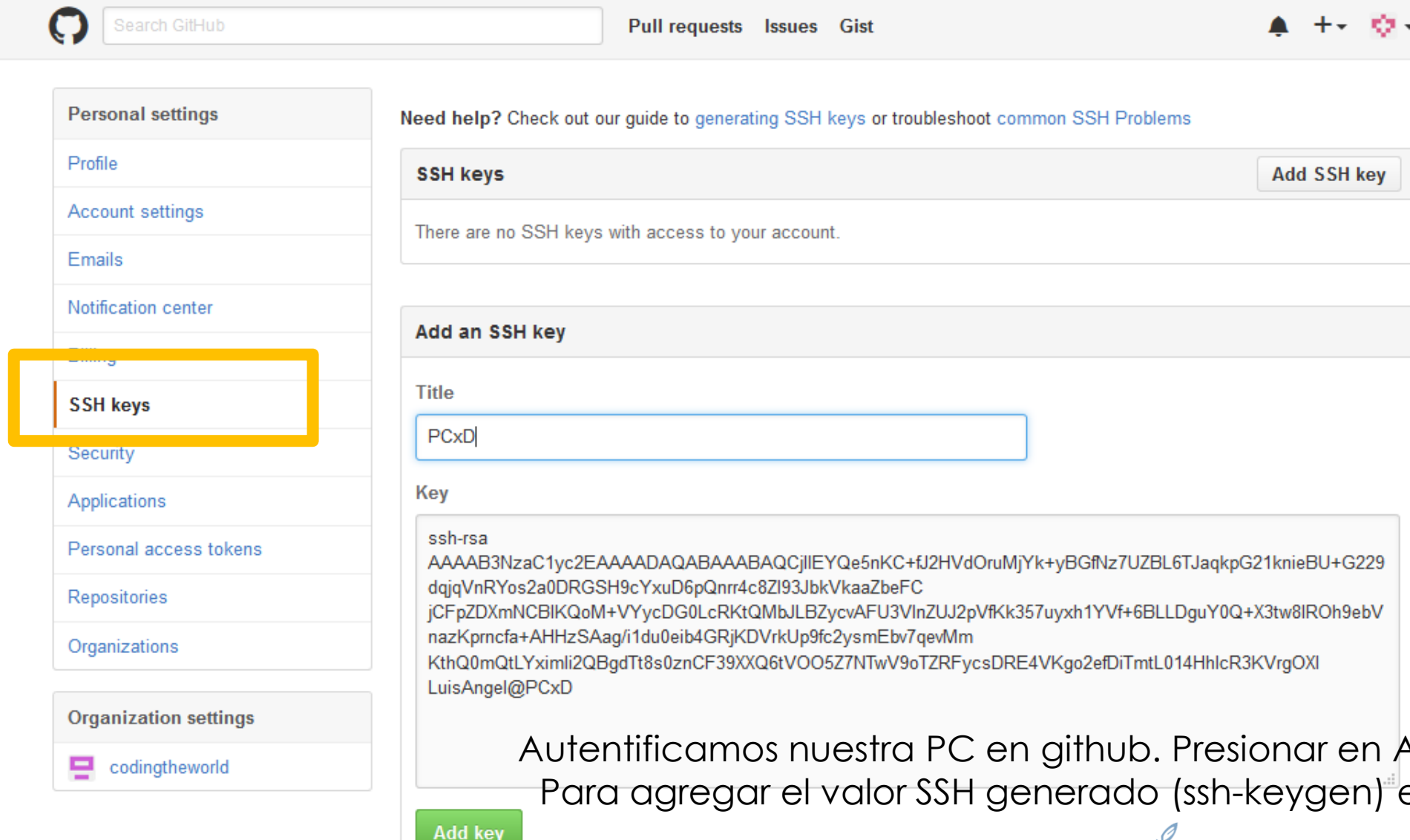
LuisAngel@PCxD MINGW64 ~/Desktop
$ git config --global user.email "testingsoftware@hotmail.com"

LuisAngel@PCxD MINGW64 ~/Desktop
$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/LuisAngel/.ssh/id_rsa):
Created directory '/c/Users/LuisAngel/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/LuisAngel/.ssh/id_rsa.
Your public key has been saved in /c/Users/LuisAngel/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:CsLRrCfG0FPgujg69G55gUleugDPSnf5FpBB9Gmp5bs LuisAngel@PCxD
The key's randomart image is:
+---[RSA 2048]-----+
|  o+o |
| . =. o |
| . = +* |
| +* +=o |
|*+B.O.. S |
| *o@ *.. |
|ooo *.o |
| = .+ .. |
| =ooo.E |
+---[SHA256]-----+

LuisAngel@PCxD MINGW64 ~/Desktop
$ cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCAj1IEYQe5nKC+fJ2Hvd0ruMjYk+yBGfNz7UZB6TJaqpG21knieBU+G229dqjqVnRYos2a0DRG5H9cYxuD6pQnrr4c8ZI93JbkVkaaZbeFCjCFpZDXmNCB1KQoM+VYycDG0LCrKtQMbJLBZycvAFU3V1nZUJ2pVfKk357uyxh1YVf+6BLLDguYQ+X3tw81R0h9ebVnazKprncfa+AHHzSAag/i1du0eib4GRjKDvrkUp9Fc2ysmEbv7gevMmsthQ0mQtLYximIi2QBgdTt8s0znCF39XXQ6tV005Z7NTwV9oTZRFycsDRE4VKgo2eFDiTmtL014h1cR3KVrg0X1 LuisAngel@PCxD
```



# Github: Settings/SSH Keys



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

**SSH keys** [Add SSH key](#)

There are no SSH keys with access to your account.

**Add an SSH key**

Title

PCxD

Key

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQACjllEYQe5nKC+fJ2HVdOruMjYk+yBGfNz7UZBL6TJaQkpG21knieBU+G229
dqjqVnRYos2a0DRGSH9cYxuD6pQnrr4c8ZI93JbkVkaaZbeFC
jCFpZDXmNCBIKQoM+VYycDG0LcRKtQMbjLBZycvAFU3VInZUJ2pVfKk357uyxh1YVf+6BLLDguY0Q+X3tw8IROh9ebV
nazKprncfa+AHHzSAag/i1du0eib4GRjKDVrkUp9fc2ysmEbv7qevMm
KthQ0mQtLYximli2QBgdTt8s0znCF39XXQ6tVOO5Z7NTwV9oTZRFycsDRE4VKgo2efDiTmtL014HhlcR3KVrgOXI
LuisAngel@PCxD
```

[Add key](#)

Autenticamos nuestra PC en github. Presionar en Add SSH Key  
Para agregar el valor SSH generado (ssh-keygen) en consola



# Github: Settings/SSH Keys

GitHub, Inc. (US) | <https://github.com/settings/ssh>

Search GitHub

Pull requests Issues Gist

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

**SSH keys** [Add SSH key](#)

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

	<b>PCxD</b> a9:50:0a:64:64:07:df:5a:83:af:e5:43:45:40:ac:1f	<a href="#">Delete</a>
--	--	------------------------

**Personal settings**

- Profile
- Account settings
- Emails
- Notification center
- Billing
- SSH keys**
- Security
- Applications
- Personal access tokens
- Repositories
- Organizations

**Organization settings**

- codingtheworld

Computadora autenticada!



# Github: Repositorio Local

```
MINGW64:/c/Users/LuisAngel/Desktop/miespacio

LuisAngel@PCxD MINGW64 ~/Desktop
$ ls
All/  Cursos/  desktop.ini  miespacio/  Organizer/  putty.exe*  Thumbs.db

LuisAngel@PCxD MINGW64 ~/Desktop
$ cd miespacio

LuisAngel@PCxD MINGW64 ~/Desktop/miespacio
$ git init
Initialized empty Git repository in C:/Users/LuisAngel/Desktop/miespacio/.git/

LuisAngel@PCxD MINGW64 ~/Desktop/miespacio (master)
$ |
```

Creamos un directorio y damos git init



# Github: Repositorio Local-Remoto. Remoto-Local

MINGW64:/c/Users/LuisAngel/Desktop/miespacio

```
LuisAngel@PCxD MINGW64 ~/Desktop
$ ls
All/  Cursos/  desktop.ini  miespacio/  Organizer/  putty.exe*  Thumbs.db

LuisAngel@PCxD MINGW64 ~/Desktop
$ cd miespacio

LuisAngel@PCxD MINGW64 ~/Desktop/miespacio
$ git init
Initialized empty Git repository in C:/Users/LuisAngel/Desktop/miespacio/.git/

LuisAngel@PCxD MINGW64 ~/Desktop/miespacio (master)
$ touch README_LOCAL

LuisAngel@PCxD MINGW64 ~/Desktop/miespacio (master)
$ git add README_LOCAL
```

Añadimos un archivo lectura a nuestro repositorio local



# Github: Repositorio Local-Remoto. Remoto Local

MINGW64:/c/Users/LuisAngel/Desktop/miespacio

```
LuisAngel@PCxD MINGW64 ~/Desktop
$ ls
All/  Cursos/  desktop.ini  miespacio/  Organizer/  putty.exe*  Thumbs.db

LuisAngel@PCxD MINGW64 ~/Desktop
$ cd miespacio

LuisAngel@PCxD MINGW64 ~/Desktop/miespacio
$ git init
Initialized empty Git repository in C:/Users/LuisAngel/Desktop/miespacio/.git/

LuisAngel@PCxD MINGW64 ~/Desktop/miespacio (master)
$ touch README_LOCAL

LuisAngel@PCxD MINGW64 ~/Desktop/miespacio (master)
$ git add README_LOCAL

LuisAngel@PCxD MINGW64 ~/Desktop/miespacio (master)
$ git commit -m "mi primer archivo local a enviarse al servidor github"
[master (root-commit) 6f68f01] mi primer archivo local a enviarse al servidor github
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 README_LOCAL

LuisAngel@PCxD MINGW64 ~/Desktop/miespacio (master)
$ |
```

Añadir un comentario. Preparando para enviarlo



# Github: Repositorio Local-Remoto. Remoto-Local

Es un repositorio para probar el funcionamiento de github — Edit

1 commit 1 branch 0 releases 1 contributor

Branch: master Testing / +

Initial commit  
TestingSoftware01 authored an hour ago latest commit 58be8ce4b3  
README.md Initial commit an hour ago

README.md  
**Testing**  
Es un repositorio para probar el funcionamiento de github

Code

Issues 0

Pull requests 0

Wiki

Pulse

Graphs

Settings

SSH clone URL

git@github.com:codin

You can clone with HTTPS, SSH, or Subversion.

Clone in Desktop

Download ZIP

Enlazar repositorio remoto con local. Copiar dirección SSH



# Github: Repositorio Local-Remoto. Remoto-Local

Es un repositorio para probar el funcionamiento de github — Edit

1 commit

1 branch

0 releases

1 contributor

Branch: master Testing / +

Initial commit

TestingSoftware01 authored an hour ago

latest commit 58be8ce4b3

README.md

Initial commit

an hour ago

README.md

## Testing

Es un repositorio para probar el funcionamiento de github

Code

Issues

Pull requests

Wiki

Pulse

Graphs

Settings

SSH clone URL

git@github.com:codi

You can clone with HTTPS,  
or Subversion.

Clone in Desktop

Download ZIP

MINGW64:/c/Users/LuisAngel/Desktop

```
LuisAngel@PCxD MINGW64 ~/Desktop
$ ls
All/ Cursos/ desktop.ini miespacio/ Organizer/ putty.exe* Thumbs.db

LuisAngel@PCxD MINGW64 ~/Desktop
$ cd miespacio

LuisAngel@PCxD MINGW64 ~/Desktop/miespacio
$ git init
Initialized empty Git repository in C:/Users/LuisAngel/Desktop/miespacio/.git/

LuisAngel@PCxD MINGW64 ~/Desktop/miespacio (master)
$ touch README_LOCAL

LuisAngel@PCxD MINGW64 ~/Desktop/miespacio (master)
$ git add README_LOCAL

LuisAngel@PCxD MINGW64 ~/Desktop/miespacio (master)
$ git commit -m "mi primer archivo local a enviarse al servidor github"
[master (root-commit) 6f68f01] mi primer archivo local a enviarse al servidor github
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 README_LOCAL
```

```
LuisAngel@PCxD MINGW64 ~/Desktop/miespacio (master)
$ git remote add origin git@github.com:codingtheworld/Testing.git

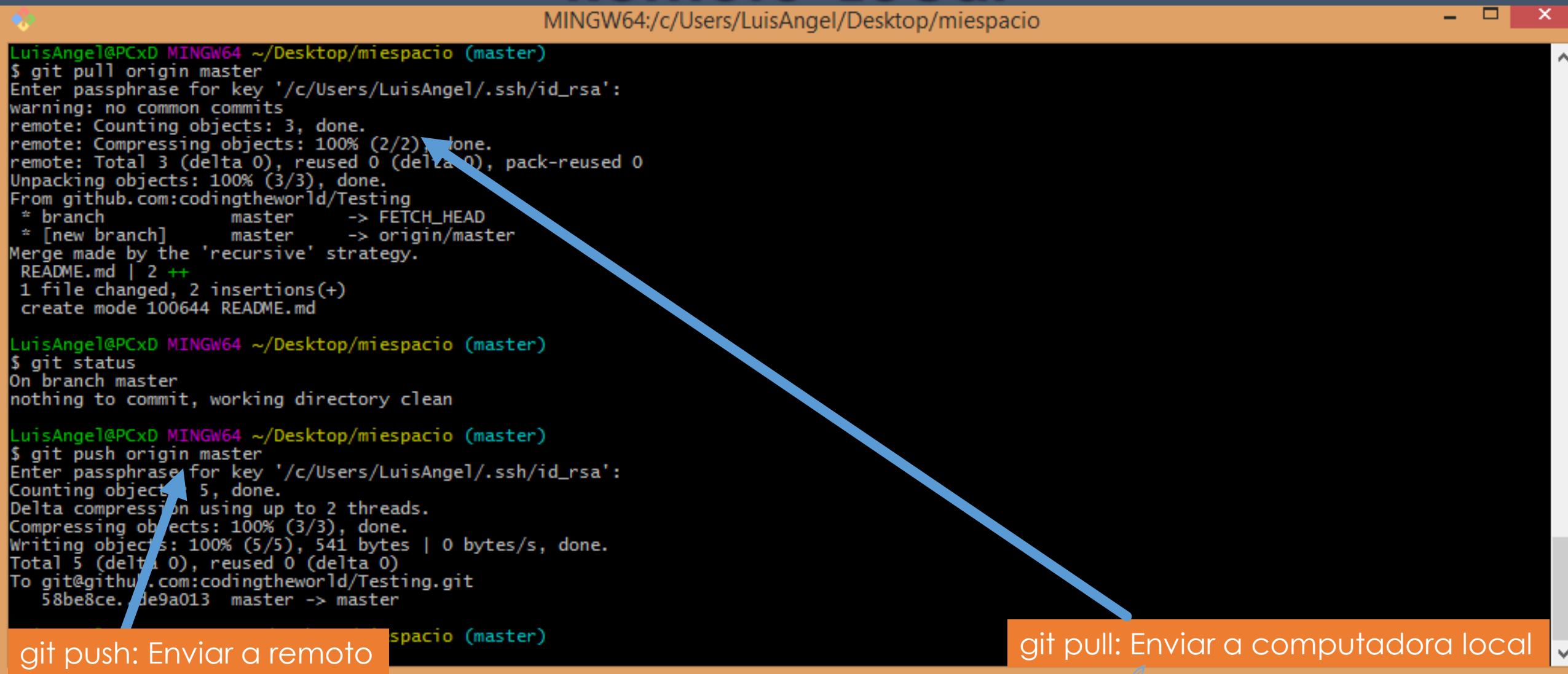
LuisAngel@PCxD MINGW64 ~/Desktop/miespacio (master)
```

Pegar dirección en consola





# Github: Repositorio Local-Remoto. Remoto-Local



```
MINGW64:/c/Users/LuisAngel/Desktop/miespacio

LuisAngel@PCxD MINGW64 ~/Desktop/miespacio (master)
$ git pull origin master
Enter passphrase for key '/c/Users/LuisAngel/.ssh/id_rsa':
warning: no common commits
remote: Counting objects: 3, done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
From github.com:codingtheworld/Testing
 * branch      master      -> FETCH_HEAD
 * [new branch] master      -> origin/master
Merge made by the 'recursive' strategy.
 README.md | 2 ++
1 file changed, 2 insertions(+)
create mode 100644 README.md

LuisAngel@PCxD MINGW64 ~/Desktop/miespacio (master)
$ git status
On branch master
nothing to commit, working directory clean

LuisAngel@PCxD MINGW64 ~/Desktop/miespacio (master)
$ git push origin master
Enter passphrase for key '/c/Users/LuisAngel/.ssh/id_rsa':
Counting objects: 5, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (5/5), 541 bytes | 0 bytes/s, done.
Total 5 (delta 0), reused 0 (delta 0)
To git@github.com:codingtheworld/Testing.git
 58be8ce..de9a013 master -> master
```

git push: Enviar a remoto

git pull: Enviar a computadora local



# Github: Repositorio Local-Remoto. Remoto-Local

```
LuisAngel@PCxD MINGW64 ~/Desktop/miespacio (master)
$ git pull origin master
Enter passphrase for key '/c/Users/LuisAngel/.ssh/id_rsa':
warning: no common commits
remote: Counting objects: 3, done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
From github.com:codingtheworld/Testing
 * branch            master      -> FETCH_HEAD
 * [new branch]      master      -> origin/master
Merge made by the 'recursive' strategy.
 README.md | 2 ++
 1 file changed, 2 insertions(+)
 create mode 100644 README.md

LuisAngel@PCxD MINGW64 ~/Desktop/miespacio (master)
$ git status
On branch master
nothing to commit, working directory clean

LuisAngel@PCxD MINGW64 ~/Desktop/miespacio (master)
$ git push origin master
Enter passphrase for key '/c/Users/LuisAngel/.ssh/id_rsa':
Delta compression using up to 2 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (5/5), 541 bytes | 0 bytes/s, done.
Total 5 (delta 0), reused 0 (delta 0)
To git@github.com:codingtheworld/Testing.git
 58be8ce..de9a013  master -> master

LuisAngel@PCxD MINGW64 ~/Desktop/miespacio (master)
$ |
```

The screenshot shows the GitHub web interface for the repository 'codingtheworld / Testing'. At the top, there's a search bar and navigation links for 'Pull requests', 'Issues', and 'Gist'. Below the repository name, there are buttons for 'Unwatch', 'Star' (0), and 'Fork' (0). The main content area shows the repository description: 'Es un repositorio para probar el funcionamiento de github — Edit'. It displays statistics: 3 commits, 1 branch, 0 releases, and 1 contributor. A list of branches is shown, with 'master' selected and a 'Testing / +' button. Below this, a commit history table lists two commits: 'Initial commit' by 'TestingSoftware01' 2 hours ago, and 'mi primer archivo local a enviarse al servidor github' 21 minutes ago. The 'README.md' file is highlighted. The right sidebar contains links for 'Code', 'Issues' (0), 'Pull requests' (0), 'Wiki', 'Pulse', 'Graphs', and 'Settings'. At the bottom, the 'SSH clone URL' is provided as 'git@github.com:codir', and a 'Clone in Desktop' button is visible.



# Github: Ramas (Branches)

MINGW64:/c/Users/LuisAngel/Desktop/miespacio

```
Delta compression using up to 2 threads.  
Compressing objects: 100% (3/3), done.  
Writing objects: 100% (5/5), 541 bytes | 0 bytes/s, done.  
Total 5 (delta 0), reused 0 (delta 0)  
To git@github.com:codingtheworld/Testing.git  
58be8ce..de9a013 master -> master
```

```
LuisAngel@PCxD MINGW64 ~/Desktop/miespacio (master)  
$ git branch desarrollo
```

```
LuisAngel@PCxD MINGW64 ~/Desktop/miespacio (master)  
$ git checkout desarrollo  
Switched to branch 'desarrollo'
```

```
LuisAngel@PCxD MINGW64 ~/Desktop/miespacio (desarrollo)  
$ git checkout master  
Switched to branch 'master'
```

git branch nombre  
git branch -d nombre

Crear y cambiar de ramas

git checkout nombre\_rama



# Github: Ramas (Branches)

GitHub interface for repository **codingtheworld / Testing**.

Navigation: Pull requests, Issues, Gist. Search: This repository Search.

Actions: Unwatch (1), Star (0), Fork (0).

Branches: Overview, Yours, Active, Stale, All branches. Search branches...

**Default branch**

master Updated 28 minutes ago by TestingSoftw... Default Change default branch

**Your branches**

desarrollo Updated 18 minutes ago by TestingSoftw... 2 | 1 New pull request

**Active branches**

desarrollo Updated 18 minutes ago by TestingSoftw... 2 | 1 New pull request

```
LuisAngel@PCxD MINGW64 ~/Desktop/miespacio (desarrollo)
$ git push origin desarrollo
Enter passphrase for key '/c/Users/LuisAngel/.ssh/id_rsa':
Counting objects: 4, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 1.76 KiB | 0 bytes/s, done.
Total 4 (delta 0), reused 0 (delta 0)
To git@github.com:codingtheworld/Testing.git
* [new branch]      desarrollo -> desarrollo
```

Actualizar branch en github

© 2015 GitHub, Inc. Terms Privacy Security



luis.monge@uabc.edu.mx



<http://www.luismonge.com.mx>



<http://www.ingenioteka.com>

# Github: Ramas (Branches)

MINGW64:/c/Users/LuisAngel/Desktop/miespacio

```
LuisAngel@PCxD MINGW64 ~/Desktop/miespacio (desarrollo)
$ git push origin desarrollo
Enter passphrase for key '/c/Users/LuisAngel/.ssh/id_rsa':
Counting objects: 4, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 1.76 KiB | 0 bytes/s, done.
Total 4 (delta 0), reused 0 (delta 0)
To git@github.com:codingtheworld/Testing.git
 * [new branch]      desarrollo -> desarrollo

LuisAngel@PCxD MINGW64 ~/Desktop/miespacio (desarrollo)
$ ls
index.html  README.md  README_LOCAL  style.css

LuisAngel@PCxD MINGW64 ~/Desktop/miespacio (desarrollo)
$ git status
On branch desarrollo
nothing to commit, working directory clean

LuisAngel@PCxD MINGW64 ~/Desktop/miespacio (desarrollo)
$ git status
On branch desarrollo
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   style.css

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

LuisAngel@PCxD MINGW64 ~/Desktop/miespacio (desarrollo)
$ git commit -m "este es un cambio para mi segundo branch"
On branch desarrollo
Changes not staged for commit:
  modified:   style.css

no changes added to commit

LuisAngel@PCxD MINGW64 ~/Desktop/miespacio (desarrollo)
$ git push origin desarrollo
Enter passphrase for key '/c/Users/LuisAngel/.ssh/id_rsa':
Everything up-to-date

LuisAngel@PCxD MINGW64 ~/Desktop/miespacio (desarrollo)
$ |
```

Actualizar branch en github



# Comandos adicionales

- ❑ **Borra una rama:** `git branch -d nombre_rama`
- ❑ **Crea un proyecto o reinicia:** `git init`
- ❑ **Crea una copia localmente:**  
`git clone git://github.com/tu_ruta`
- ❑ **Fusiona Ramas**  
`git merge nombre_rama`



# Comandos adicionales

## ❑ Moverse entre versiones

git log o git log --decorate

```
commit f04d938eaec19c31e7185389a51da03339e83d1d (HEAD, master)  
Author: Luis <lmonge@uabc.edu.com>  
Date: Sun Sept 02 01:24:35 2015 +0100
```

git checkout `Agrega_el_id_commit`





## Si tuviste algún problema al lograr el ejercicio, sigue estos pasos

1. Crea un nuevo repositorio en tu cuenta personal de **github**
2. Crea un nuevo directorio en tu computadora
3. Por medio de la consola **git bash** ingresa al directorio. Dentro del directorio aplica el comando **git init**, para indicar que será un directorio con control de versiones
4. Coloca algún archivo dentro del directorio creado.
5. Escribe el comando **git status** para verificar si detecto el archivo colocado en el directorio
6. En la pagina de github.com copia la ruta SSH del repositorio creado y ahora dirígete a la consola git bash y escribe **git remote add origin** y la ruta SSH obtenida de github.com
7. Escribe el comando **git status**, **git add nombreadarchivo.txt**, **git commit -m "mensaje"** y **git push origin master**.
8. Ahora verifica en github si el archivo fue enviado





## M.I Luis Angel Monge



<http://www.luismonge.com.mx>



[luis.monge@uabc.edu.mx](mailto:luis.monge@uabc.edu.mx)



[profesor@luismonge.com.mx](mailto:profesor@luismonge.com.mx)



[/ing.luismonge](https://www.facebook.com/ing.luismonge)



[/in/luisangelmonge](https://www.linkedin.com/in/luisangelmonge)



<http://www.ingenioteka.com>