

Breve descripción de Git.



Git es un sistema de control de versiones distribuido [gratuito y de código abierto](#) diseñado para manejar todo, desde proyectos pequeños hasta muy grandes, con rapidez y eficiencia.

Git es [fácil de aprender](#) y tiene una [huella pequeña con un rendimiento increíblemente rápido](#). Supera a las herramientas SCM (*Software configuration management*) como Subversion, CVS, Perforce y ClearCase con características como [ramificación local barata](#), [áreas de preparación](#) convenientes y [múltiples flujos de trabajo](#).



¿Quién lo diseñó y por qué?

Diseñado por [Linus Torvalds](#), pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando éstas tienen un gran número de archivos de [código fuente](#).

Una breve historia de Git

<https://git-scm.com/book/es/v1/Empezando-Una-breve-historia-de-Git>

Como muchas de las grandes cosas en esta vida, Git comenzó con un poco de destrucción creativa y encendida polémica. El núcleo de Linux es un proyecto de software de código abierto con un alcance bastante grande. Durante la mayor parte del mantenimiento del núcleo de Linux (1991-2002), los cambios en el software se pasaron en forma de parches y archivos. En 2002, el proyecto del núcleo de Linux empezó a usar un DVCS propietario llamado BitKeeper.

En 2005, la relación entre la comunidad que desarrollaba el núcleo de Linux y la compañía que desarrollaba BitKeeper se vino abajo, y la herramienta dejó de ser ofrecida gratuitamente. Esto impulsó a la comunidad de desarrollo de Linux (y en particular a Linus Torvalds, el creador de Linux) a desarrollar su propia herramienta basada en algunas de las lecciones que aprendieron durante el uso de BitKeeper. Algunos de los objetivos del nuevo sistema fueron los siguientes:

- Velocidad
- Diseño sencillo
- Fuerte apoyo al desarrollo no lineal (miles de ramas paralelas)
- Completamente distribuido
- Capaz de manejar grandes proyectos (como el núcleo de Linux) de manera eficiente (velocidad y tamaño de los datos)

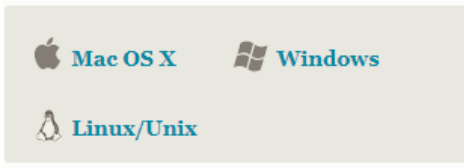
Desde su nacimiento en 2005, Git ha evolucionado y madurado para ser fácil de usar y aún conservar estas cualidades iniciales. Es tremendamente rápido, muy eficiente con grandes proyectos, y tiene un increíble sistema de ramificación (branching) para desarrollo no lineal

Sitio web.

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

Plataforma de descarga.

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



Instalación de Git en Linux.

<https://git-scm.com/download/linux>

Debian/Ubuntu

For the latest stable version for your release of Debian/Ubuntu

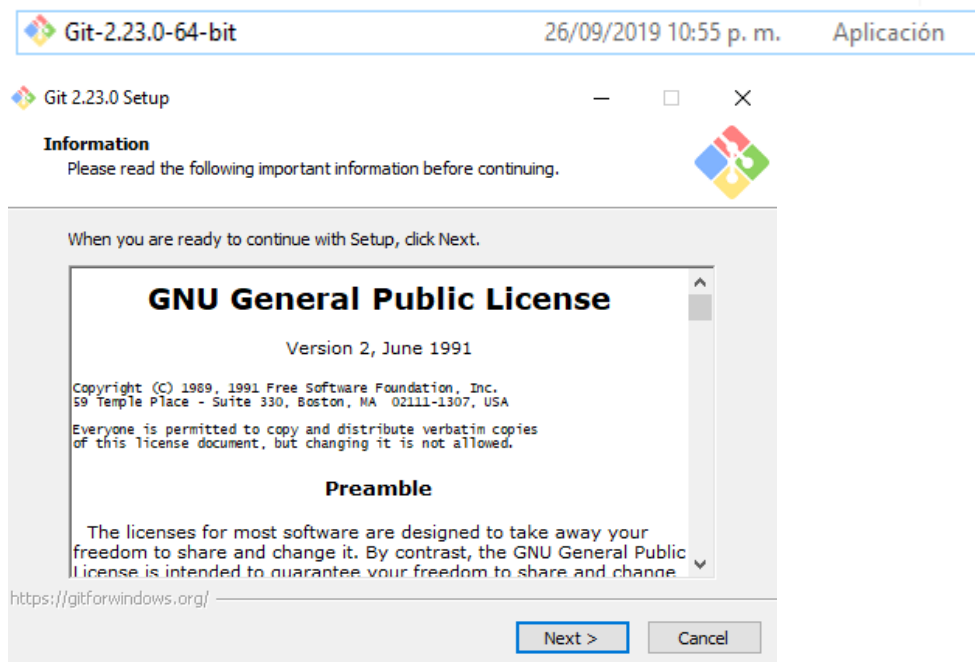
```
# sudo apt-get install git
```

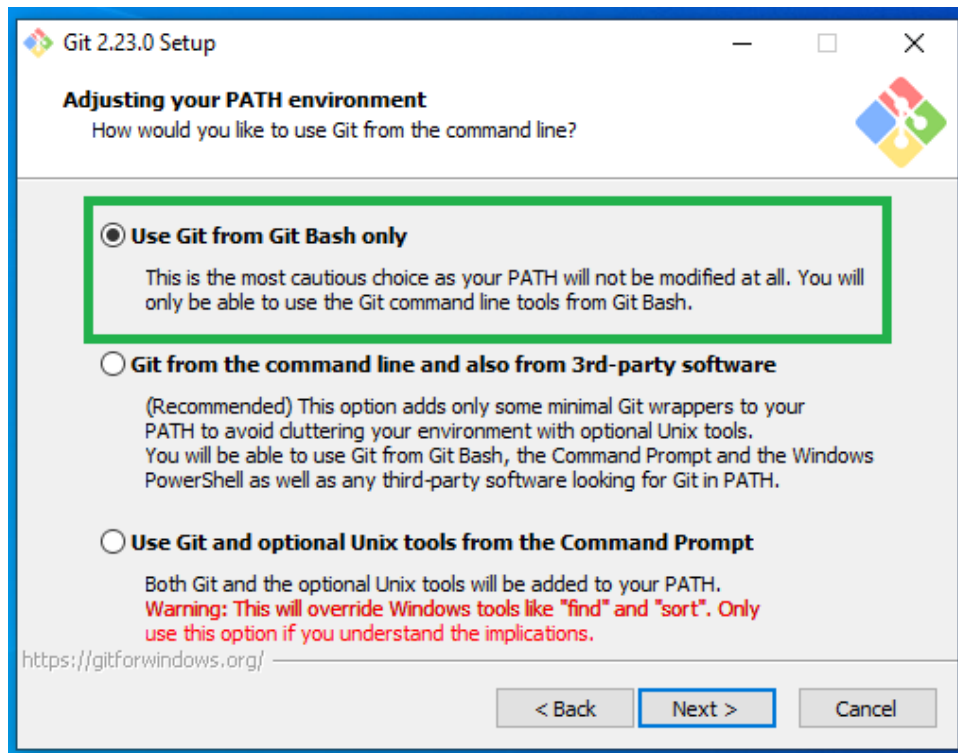
Revisar con la siguiente instrucción, si se instaló correctamente git y su versión.

```
# git --version
```

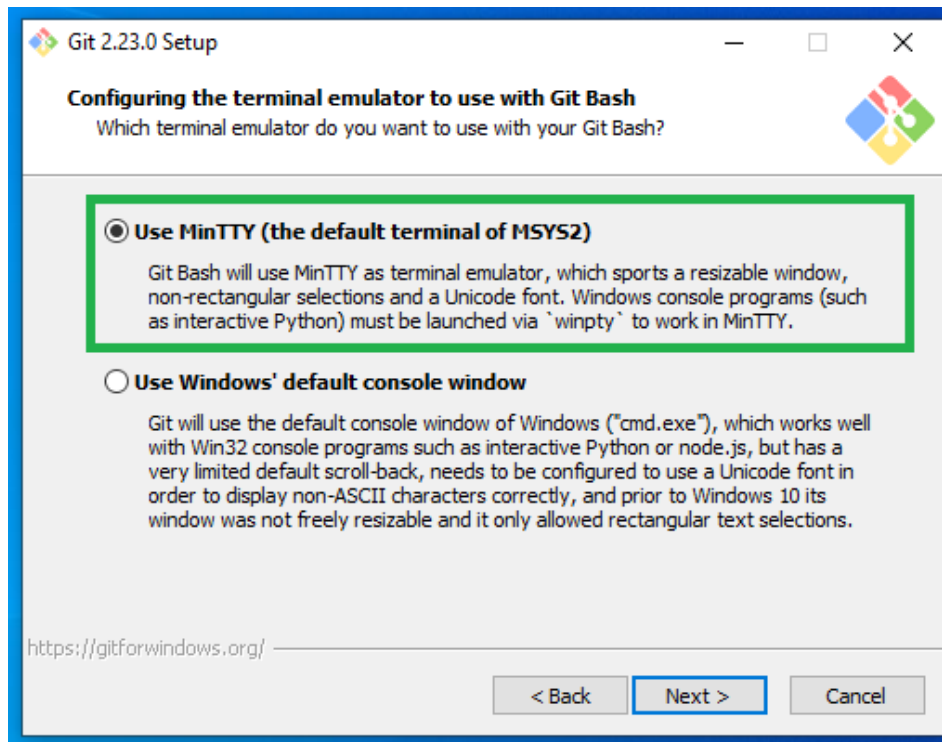
Instalación de Git en Windows.

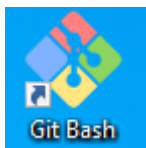
Descargar el software de git del sitio web oficial



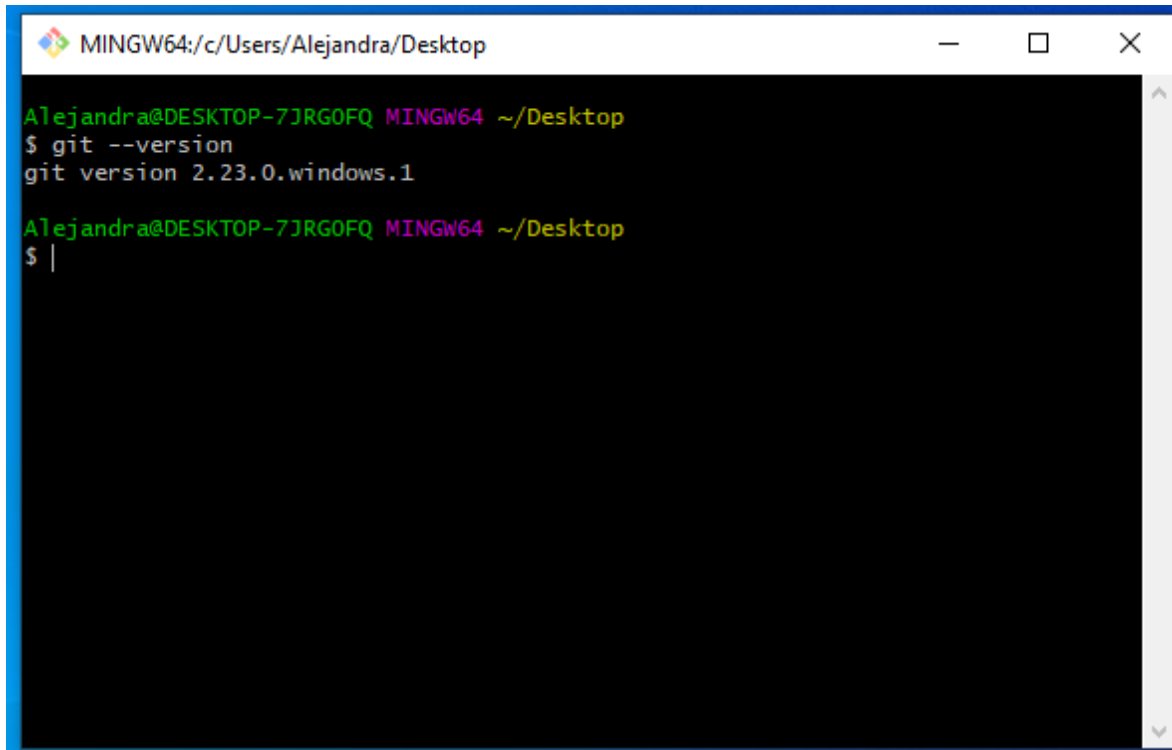
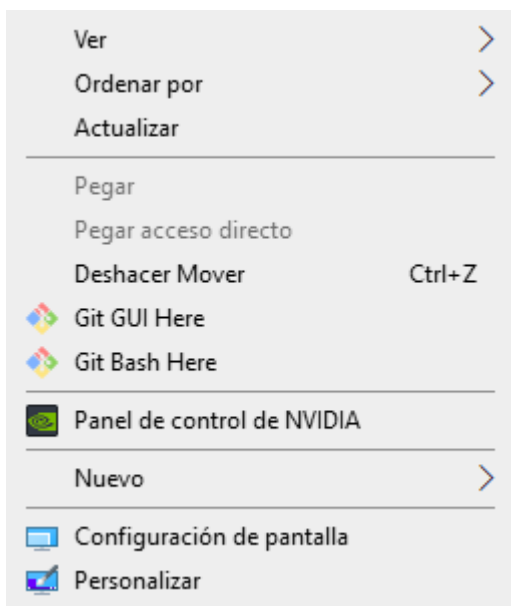


Esta opción es muy importante, ya que utiliza git desde el git bash (en específico git bash, es lo más parecido a una terminal de Unix).





Menú contextual después de instalar Git

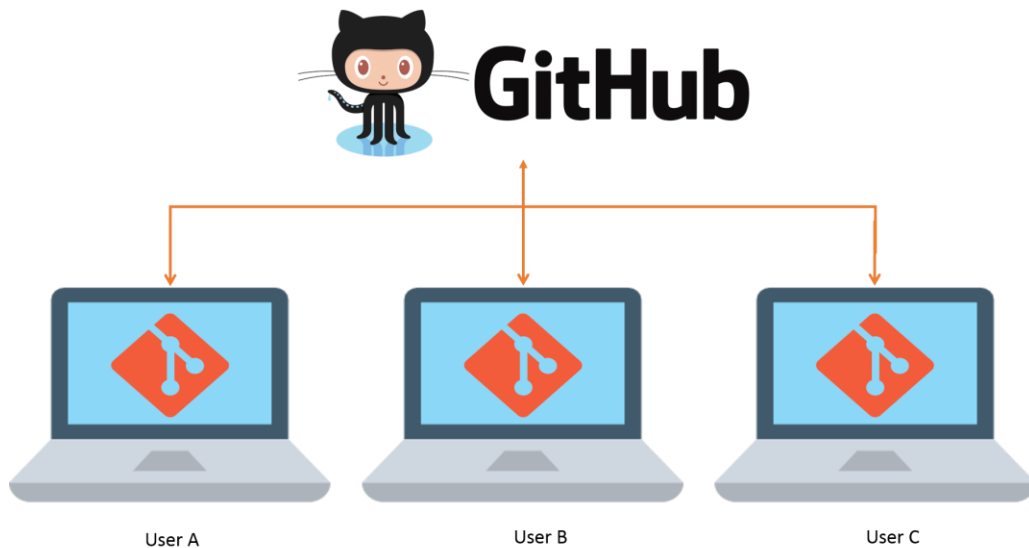
A screenshot of a Git Bash terminal window. The title bar shows 'MINGW64:/c/Users/Alejandra/Desktop'. The terminal output is:

```
Alejandra@DESKTOP-7JRG0FQ MINGW64 ~/Desktop
$ git --version
git version 2.23.0.windows.1

Alejandra@DESKTOP-7JRG0FQ MINGW64 ~/Desktop
$ |
```

GitHub.

<https://github.com/github>



Creado para desarrolladores

GitHub es una plataforma de desarrollo inspirada en tu forma de trabajar. Desde [código abierto](#) hasta [negocios](#), puede alojar y revisar código, administrar proyectos y crear software junto con 40 millones de desarrolladores.

¿Qué es GitHub?

Es una plataforma de desarrollo colaborativo, para alojar proyectos utilizando el sistema de control de versiones Git. Se utiliza principalmente para la creación de código fuente de programas de computadora. El software que opera GitHub fue escrito en Ruby on Rails.

Git vs GitHub



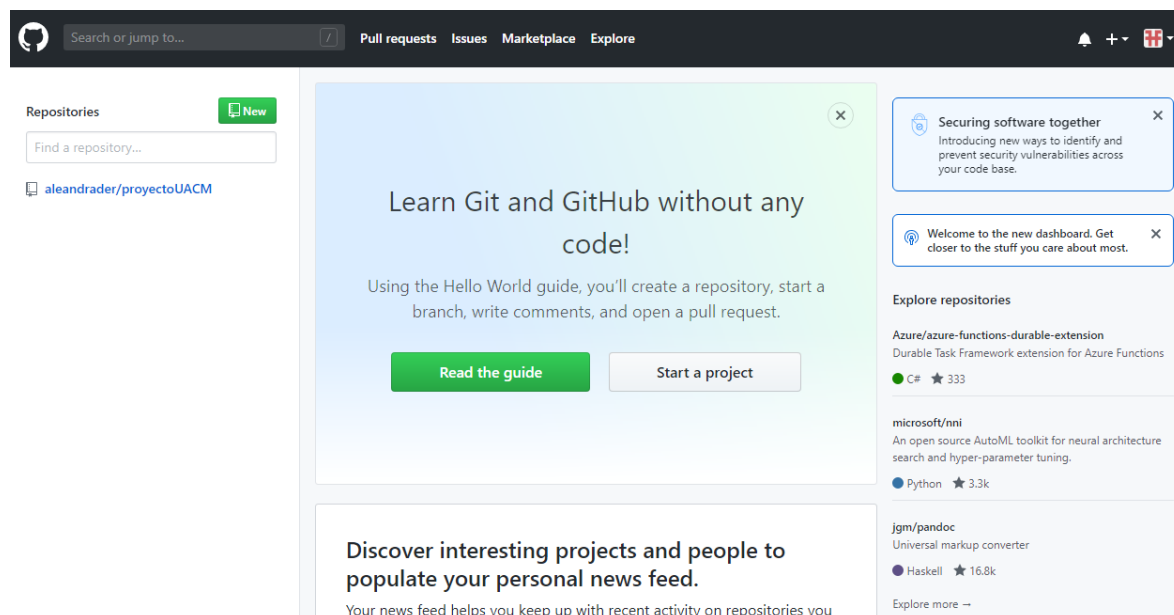
Git es el **sistema de control de versiones**, que es un software permite tener un control de cambios sobre el código fuente de una aplicación. Fue inventado por Linus Torwalds, el mismo ingeniero que comenzó el desarrollo del Kernel de Linux.

GitHub es **una página que ofrece un grupo de servicios que facilitan el uso de Git**, como por ejemplo hosting de proyectos, facilidades de colaboración, reviews de código, perfiles personales, pull requests, issues, etc.

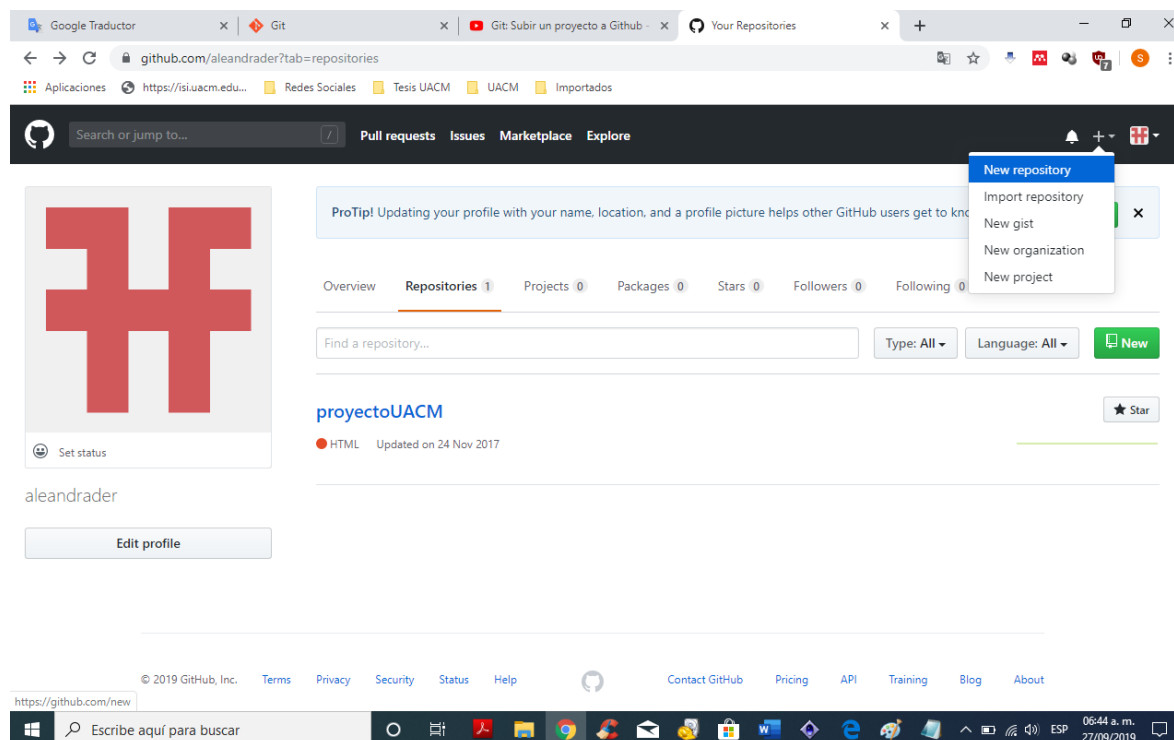
GitHub

Para acceder a github, sólo necesitas crear un usuario y una contraseña.

Página principal de GitHub



¿Cómo crear un nuevo repositorio en GitHub?



Create a new repository

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

Owner



aleandrader ▾



Repository name *

ProyectoACS



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

Description (optional)

Este repositorio es creado para alojar el proyecto de la materia Aseguramiento de la Calidad de Software que s



Public

Anyone can see this repository. You choose who can commit.



Private

You choose who can see and commit to this repository.

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



Initialize this repository with a README

This will let you immediately clone the repository to your computer.

Add .gitignore: None ▾

Add a license: None ▾



Create repository

aleandrader / ProyectoACS

Unwatch ▾

1

★ Star

0

🍴 Fork

0

Code

Issues 0

Pull requests 0

Projects 0

Wiki

Security

Insights

Settings

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

Set up in Desktop or HTTPS SSH `https://github.com/aleandrader/ProyectoACS.git`

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

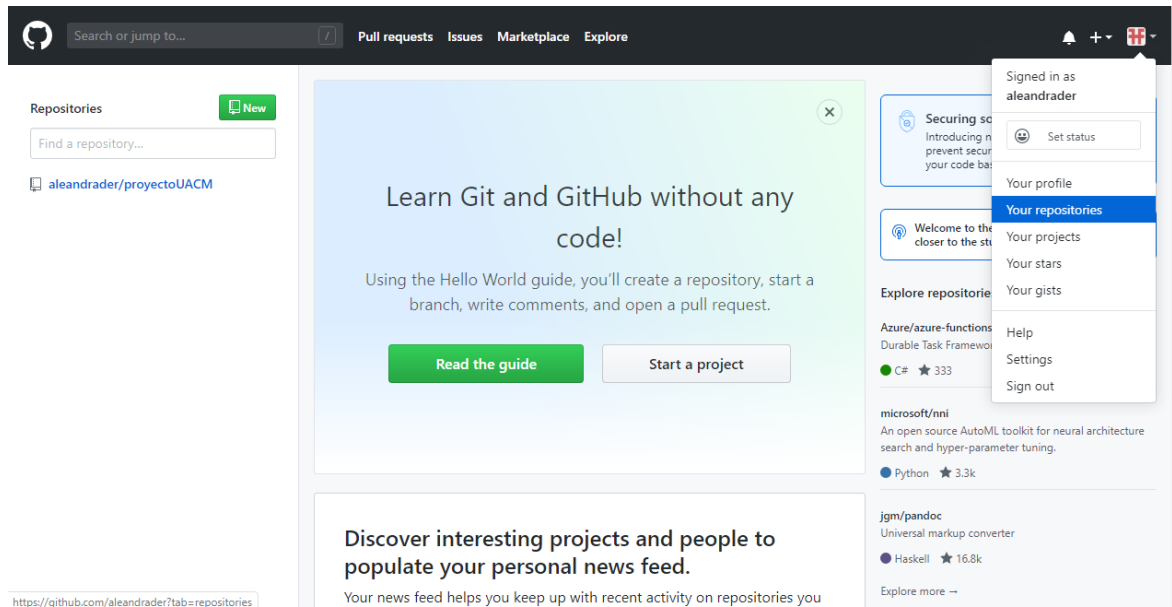
...or create a new repository on the command line

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

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

```
git remote add origin https://github.com/aleandrader/ProyectoACS.git
git push -u origin master
```

¿Cómo ver tus repositorios?



Guía de comandos básicos de git.

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

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

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

```
git remote add origin https://github.com/aleandrader/ProyectoACS.git
git push -u origin master
```

Comando para obtener ayuda de los comandos de git en la terminal.

\$git help

```
MINGW64:/c/Users/Alejandra/Desktop
$ git config --global user.name "<aleandrader@gmail.com.mx>"

Alejandra@DESKTOP-7JRG0FQ MINGW64 ~/Desktop
$ git help
usage: git [--version] [--help] [-C <path>] [-c <name>=<value>]
      [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
      [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--bare]
      [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
      <command> [<args>]

These are common Git commands used in various situations:

start a working area (see also: git help tutorial)
  clone      Clone a repository into a new directory
  init       Create an empty Git repository or reinitialize an existing one

work on the current change (see also: git help everyday)
  add        Add file contents to the index
  mv         Move or rename a file, a directory, or a symlink
  restore    Restore working tree files
  rm         Remove files from the working tree and from the index

examine the history and state (see also: git help revisions)
  bisect     Use binary search to find the commit that introduced a bug
  diff       Show changes between commits, commit and working tree, etc
  grep       Print lines matching a pattern
  log        Show commit logs
  show       Show various types of objects
  status     Show the working tree status

grow, mark and tweak your common history
  branch     List, create, or delete branches
  commit     Record changes to the repository
  merge      Join two or more development histories together
  rebase     Reapply commits on top of another base tip
  reset      Reset current HEAD to the specified state
  switch     Switch branches
  tag        Create, list, delete or verify a tag object signed with GPG

collaborate (see also: git help workflows)
  fetch      Download objects and refs from another repository
  pull       Fetch from and integrate with another repository or a local branch
  push       Update remote refs along with associated objects

'git help -a' and 'git help -g' list available subcommands and some
concept guides. See 'git help <command>' or 'git help <concept>'
to read about a specific subcommand or concept.
See 'git help git' for an overview of the system.

Alejandra@DESKTOP-7JRG0FQ MINGW64 ~/Desktop
$ |
```

Práctica paso a paso.

Instrucciones Generales.

1. Instala git en Windows y en Linux, obtén tu captura de pantalla para tener la evidencia de su instalación final (no captures todas las pantallas).
2. Obtener una cuenta de github.
3. Crear un repositorio en GitHub.
4. Subir un proyecto a GitHub, desde git utilizando la consola de bashgit en Windows o utilizando una terminal en Linux.

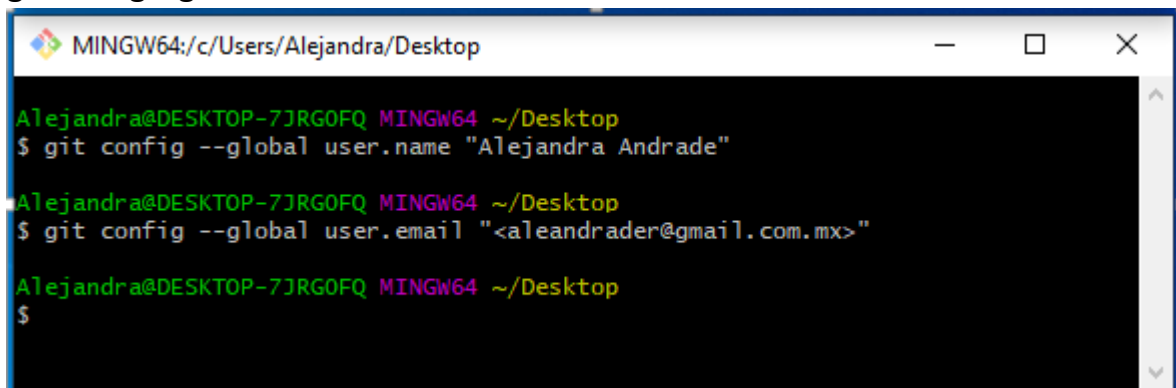
Subiendo un proyecto a GitHub.

1. Crear en Github un nuevo repositorio público, donde almacenaras el código de un proyecto.
2. Realizar paso a paso el upload o la carga utilizando git.
 - a. Configuración inicial de git

Esto es muy importante, porque el nombre y correo identifican específicamente que persona o autor que realiza los cambios.

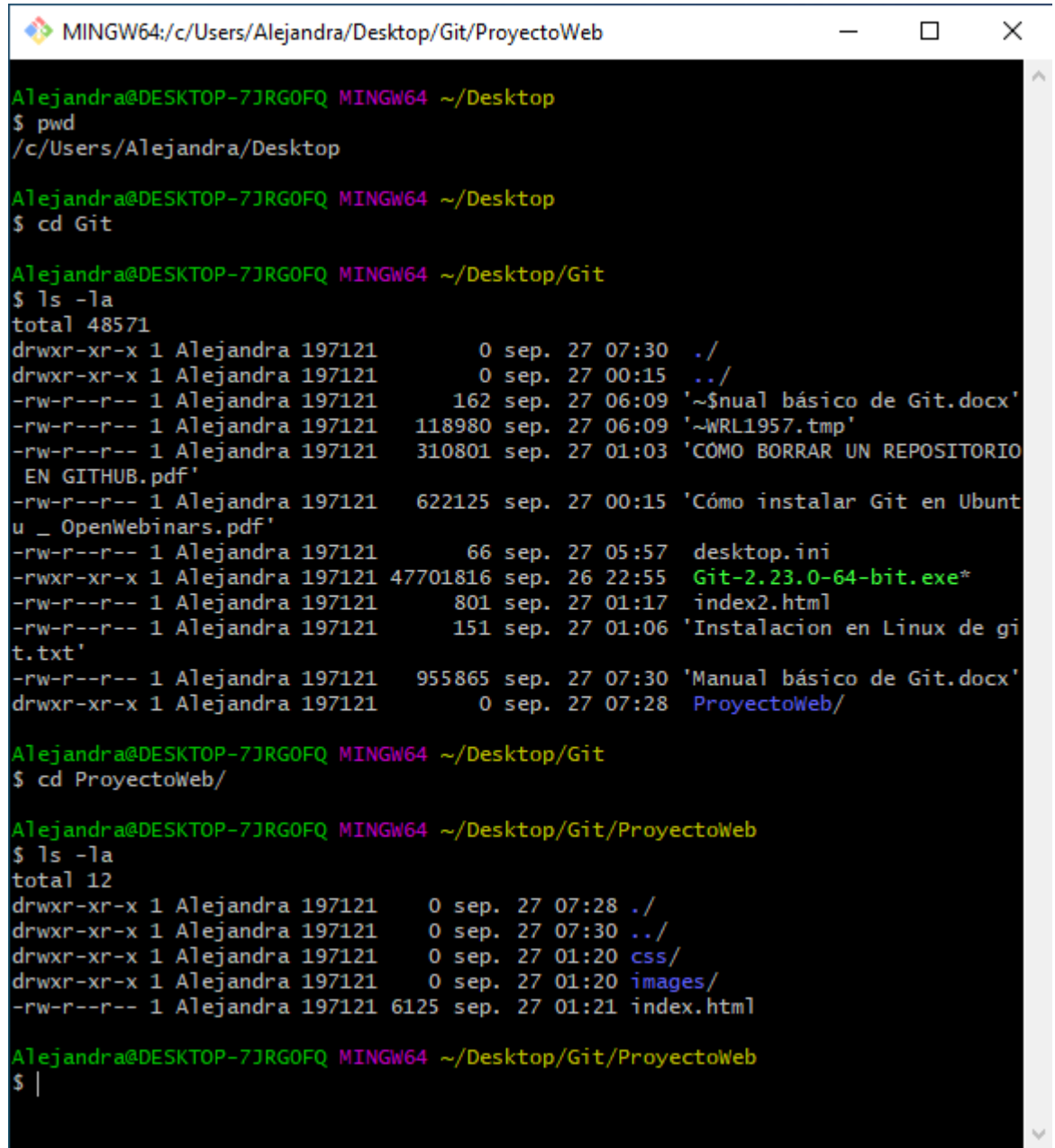
`git config --global user.name "Nombre del desarrollador"`

`git config --global user.email "<xx@uacm.edu.mx>"`



The screenshot shows a Windows command prompt window titled "MINGW64; c:/Users/Alejandra/Desktop". The prompt is "Alejandra@DESKTOP-7JRG0FQ MINGW64 ~/Desktop". The user enters the command `$ git config --global user.name "Alejandra Andrade"`, and the prompt returns. Then the user enters `$ git config --global user.email "<aleandrader@gmail.com.mx>"`, and the prompt returns. Finally, the user enters `$` and the prompt returns.

- b. Ingresa a tu directorio en donde se encuentra tu proyecto.



```
MINGW64:/c/Users/Alejandra/Desktop/Git/ProyectoWeb

Alejandra@DESKTOP-7JRG0FQ MINGW64 ~/Desktop
$ pwd
/c/Users/Alejandra/Desktop

Alejandra@DESKTOP-7JRG0FQ MINGW64 ~/Desktop
$ cd Git

Alejandra@DESKTOP-7JRG0FQ MINGW64 ~/Desktop/Git
$ ls -la
total 48571
drwxr-xr-x 1 Alejandra 197121      0 sep. 27 07:30 ./
drwxr-xr-x 1 Alejandra 197121      0 sep. 27 00:15 ../
-rw-r--r-- 1 Alejandra 197121    162 sep. 27 06:09 '~$nual básico de Git.docx'
-rw-r--r-- 1 Alejandra 197121  118980 sep. 27 06:09 '~WRL1957.tmp'
-rw-r--r-- 1 Alejandra 197121  310801 sep. 27 01:03 'COMO BORRAR UN REPOSITORIO
EN GITHUB.pdf'
-rw-r--r-- 1 Alejandra 197121  622125 sep. 27 00:15 'Cómo instalar Git en Ubuntu
u _ OpenWebinars.pdf'
-rw-r--r-- 1 Alejandra 197121     66 sep. 27 05:57 desktop.ini
-rwxr-xr-x 1 Alejandra 197121 47701816 sep. 26 22:55 Git-2.23.0-64-bit.exe*
-rw-r--r-- 1 Alejandra 197121    801 sep. 27 01:17 index2.html
-rw-r--r-- 1 Alejandra 197121    151 sep. 27 01:06 'Instalacion en Linux de git.txt'
-rw-r--r-- 1 Alejandra 197121  955865 sep. 27 07:30 'Manual básico de Git.docx'
drwxr-xr-x 1 Alejandra 197121      0 sep. 27 07:28 ProyectoWeb/

Alejandra@DESKTOP-7JRG0FQ MINGW64 ~/Desktop/Git
$ cd ProyectoWeb/

Alejandra@DESKTOP-7JRG0FQ MINGW64 ~/Desktop/Git/ProyectoWeb
$ ls -la
total 12
drwxr-xr-x 1 Alejandra 197121      0 sep. 27 07:28 ./
drwxr-xr-x 1 Alejandra 197121      0 sep. 27 07:30 ../
drwxr-xr-x 1 Alejandra 197121      0 sep. 27 01:20 css/
drwxr-xr-x 1 Alejandra 197121      0 sep. 27 01:20 images/
-rw-r--r-- 1 Alejandra 197121   6125 sep. 27 01:21 index.html

Alejandra@DESKTOP-7JRG0FQ MINGW64 ~/Desktop/Git/ProyectoWeb
$ |
```

Resumen

```
Alejandra@DESKTOP-7JRG0FQ MINGW64 ~/Desktop/Git/ProyectoWeb
$ pwd
/c/Users/Alejandra/Desktop/Git/ProyectoWeb
Alejandra@DESKTOP-7JRG0FQ MINGW64 ~/Desktop/Git/ProyectoWeb
$ ls -la
total 12
drwxr-xr-x 1 Alejandra 197121  0 sep. 27 07:28 ./
drwxr-xr-x 1 Alejandra 197121  0 sep. 27 07:30 ../
drwxr-xr-x 1 Alejandra 197121  0 sep. 27 01:20 css/
drwxr-xr-x 1 Alejandra 197121  0 sep. 27 01:20 images/
-rw-r--r-- 1 Alejandra 197121 6125 sep. 27 01:21 index.html
```

- c. Crear un repositorio vacío de git e inicializarlo.

Una vez creado el repositorio, es necesario inicializarlo con el comando:

```
$ git init
```

```
Alejandra@DESKTOP-7JRG0FQ MINGW64 ~/Desktop/Git/ProyectoWeb
$ pwd
/c/Users/Alejandra/Desktop/Git/ProyectoWeb
Alejandra@DESKTOP-7JRG0FQ MINGW64 ~/Desktop/Git/ProyectoWeb
$ ls -la
total 12
drwxr-xr-x 1 Alejandra 197121  0 sep. 27 07:28 ./
drwxr-xr-x 1 Alejandra 197121  0 sep. 27 07:30 ../
drwxr-xr-x 1 Alejandra 197121  0 sep. 27 01:20 css/
drwxr-xr-x 1 Alejandra 197121  0 sep. 27 01:20 images/
-rw-r--r-- 1 Alejandra 197121 6125 sep. 27 01:21 index.html
Alejandra@DESKTOP-7JRG0FQ MINGW64 ~/Desktop/Git/ProyectoWeb
$ git init
Initialized empty Git repository in C:/Users/Alejandra/Desktop/Git/ProyectoWeb/.git/
Alejandra@DESKTOP-7JRG0FQ MINGW64 ~/Desktop/Git/ProyectoWeb (master)
$ |
```

Observa que has inicializado el repositorio localmente en tu ordenador. El repositorio es llamado ProyectoWeb y ya es un repositorio vacío de git.

Nota: La carpeta .git almacena toda la información del proyecto y repositorio. Es conveniente no trabajar con ella.

Hasta este punto, aún no hemos accedido al repositorio distribuido al que todos los desarrolladores depositaran sus contribuciones,

por supuesto el repositorio distribuido es ProyectoACS, el cual se creo en el punto 3 de las instrucciones generales.

- d. Enviar el código del proyecto local, al repositorio distribuido de github.

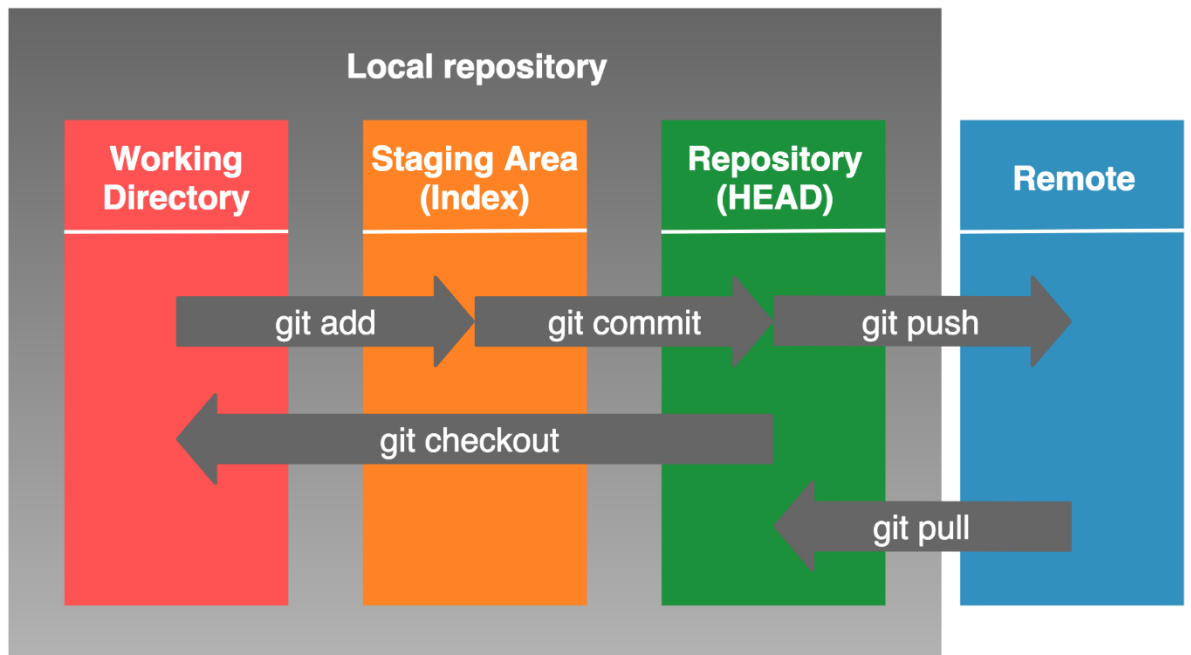
Para añadir un archivo al staging index se ejecuta
`git add nombreArchivo.ext`

Otra opción, es añadir todos los archivos modificados si escribimos el comando
`git add .`

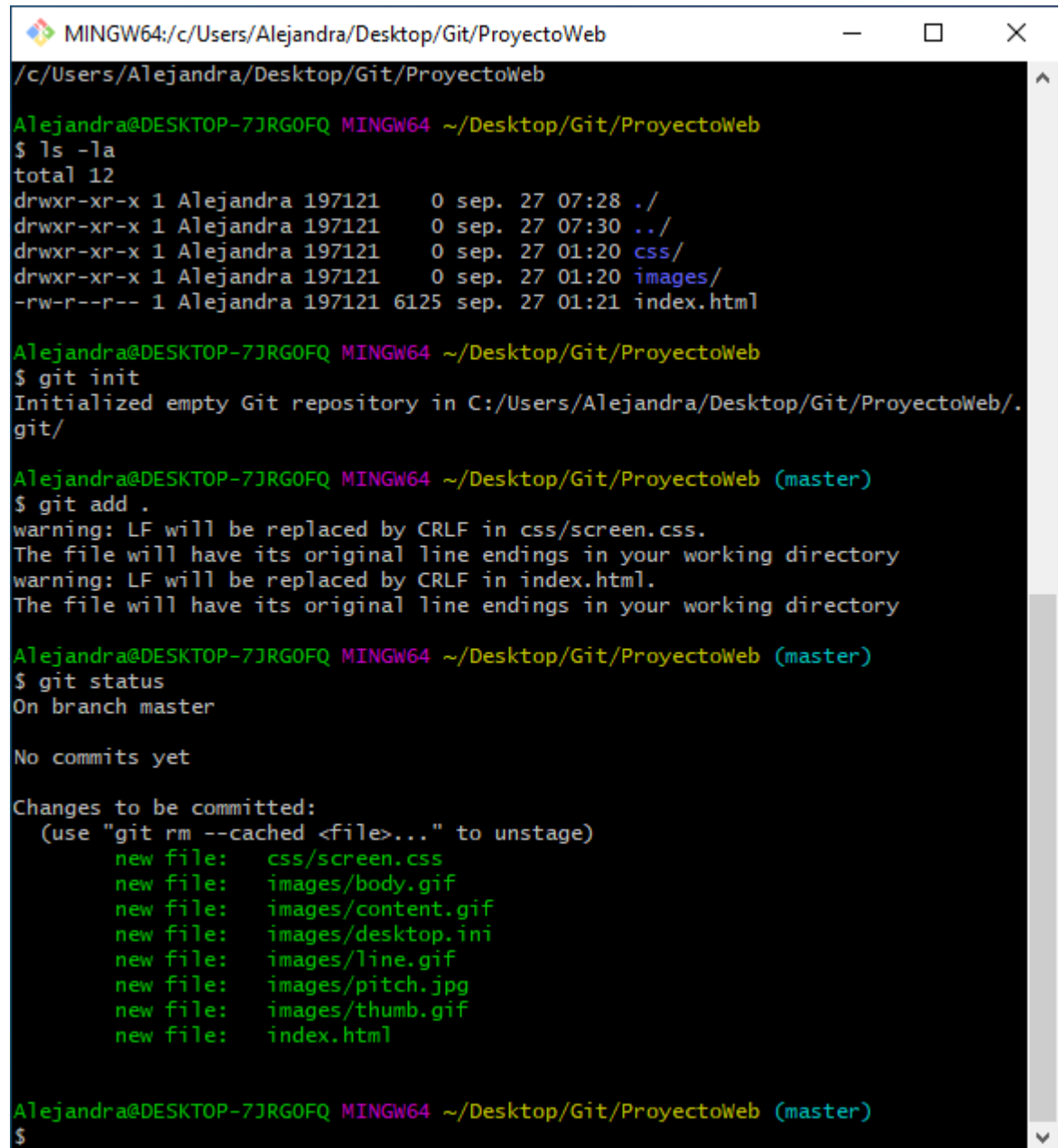
El comando

`$git add NombreArchivo.extensión`

Agrega los archivos a una parte intermedia.



Ejecución en terminal.



```
MINGW64:/c/Users/Alejandra/Desktop/Git/ProyectoWeb
/c/Users/Alejandra/Desktop/Git/ProyectoWeb

Alejandra@DESKTOP-7JRG0FQ MINGW64 ~/Desktop/Git/ProyectoWeb
$ ls -la
total 12
drwxr-xr-x 1 Alejandra 197121  0 sep. 27 07:28 ./
drwxr-xr-x 1 Alejandra 197121  0 sep. 27 07:30 ../
drwxr-xr-x 1 Alejandra 197121  0 sep. 27 01:20 css/
drwxr-xr-x 1 Alejandra 197121  0 sep. 27 01:20 images/
-rw-r--r-- 1 Alejandra 197121 6125 sep. 27 01:21 index.html

Alejandra@DESKTOP-7JRG0FQ MINGW64 ~/Desktop/Git/ProyectoWeb
$ git init
Initialized empty Git repository in C:/Users/Alejandra/Desktop/Git/ProyectoWeb/.git/

Alejandra@DESKTOP-7JRG0FQ MINGW64 ~/Desktop/Git/ProyectoWeb (master)
$ git add .
warning: LF will be replaced by CRLF in css/screen.css.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in index.html.
The file will have its original line endings in your working directory

Alejandra@DESKTOP-7JRG0FQ MINGW64 ~/Desktop/Git/ProyectoWeb (master)
$ git status
On branch master

No commits yet

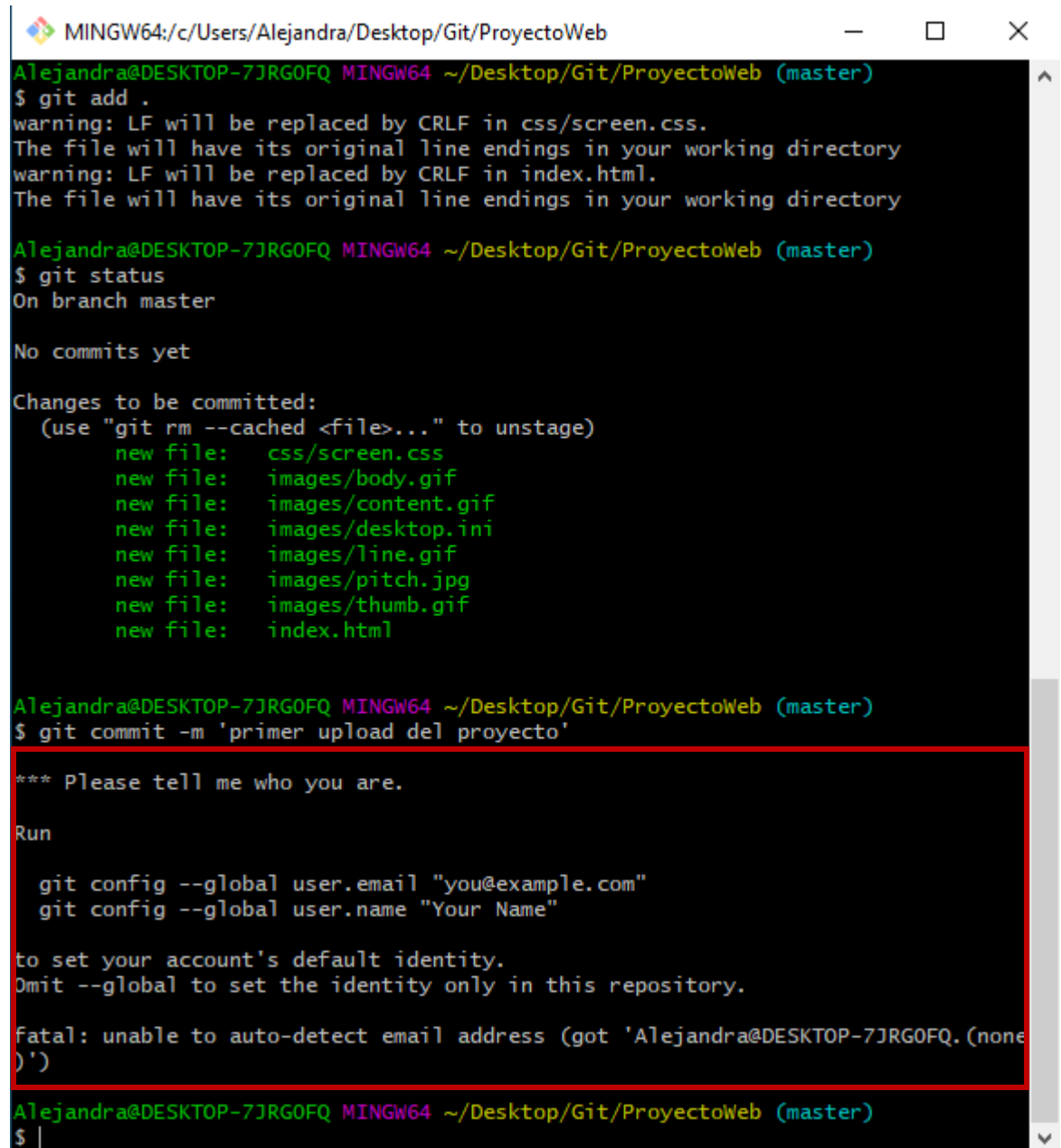
Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   css/screen.css
    new file:   images/body.gif
    new file:   images/content.gif
    new file:   images/desktop.ini
    new file:   images/line.gif
    new file:   images/pitch.jpg
    new file:   images/thumb.gif
    new file:   index.html

Alejandra@DESKTOP-7JRG0FQ MINGW64 ~/Desktop/Git/ProyectoWeb (master)
$
```

- e. Para agregar el proyecto directamente al repositorio git es necesario agregar el comando
\$git commit -m 'primer commit'

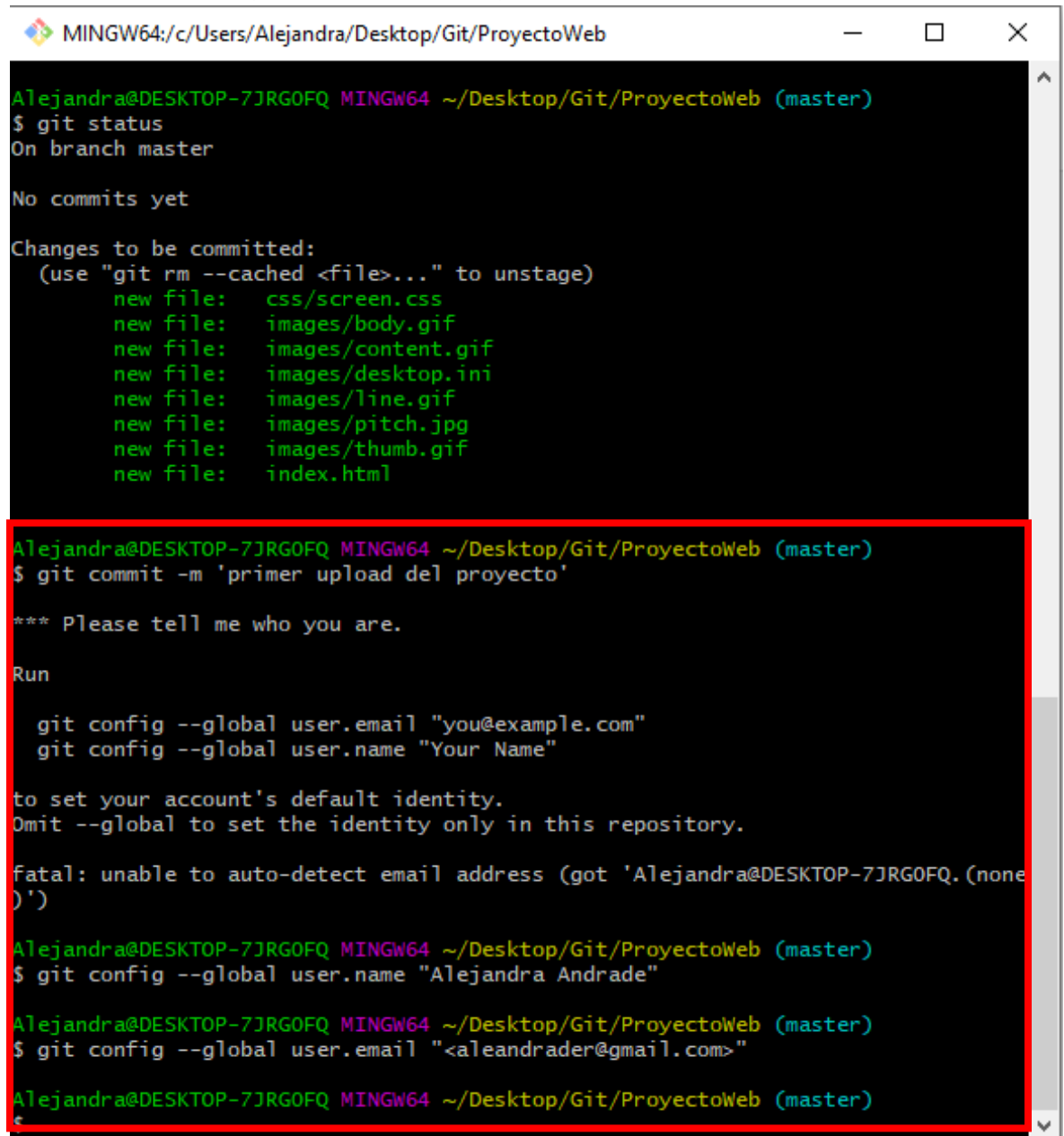
Ojo es necesario que le hayas dicho a git quien eres, configurando el repositorio con el comando


```
$git config --global user.name  
$git config --global user.email
```



```
MINGW64:/c/Users/Alejandra/Desktop/Git/ProyectoWeb  
Alejandra@DESKTOP-7JRG0FQ MINGW64 ~/Desktop/Git/ProyectoWeb (master)  
$ git add .  
warning: LF will be replaced by CRLF in css/screen.css.  
The file will have its original line endings in your working directory  
warning: LF will be replaced by CRLF in index.html.  
The file will have its original line endings in your working directory  
  
Alejandra@DESKTOP-7JRG0FQ MINGW64 ~/Desktop/Git/ProyectoWeb (master)  
$ git status  
On branch master  
  
No commits yet  
  
Changes to be committed:  
  (use "git rm --cached <file>..." to unstage)  
    new file:   css/screen.css  
    new file:   images/body.gif  
    new file:   images/content.gif  
    new file:   images/desktop.ini  
    new file:   images/line.gif  
    new file:   images/pitch.jpg  
    new file:   images/thumb.gif  
    new file:   index.html  
  
Alejandra@DESKTOP-7JRG0FQ MINGW64 ~/Desktop/Git/ProyectoWeb (master)  
$ git commit -m 'primer upload del proyecto'  
  
*** 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 'Alejandra@DESKTOP-7JRG0FQ.(none)')  
  
Alejandra@DESKTOP-7JRG0FQ MINGW64 ~/Desktop/Git/ProyectoWeb (master)  
$ |
```

Salida de la configuración del repositorio, identificado el usuario y correo del desarrollador de software que va a realizar una contribución al repositorio local de git.



```
MINGW64:/c/Users/Alejandra/Desktop/Git/ProyectoWeb

Alejandra@DESKTOP-7JRG0FQ MINGW64 ~/Desktop/Git/ProyectoWeb (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   css/screen.css
    new file:   images/body.gif
    new file:   images/content.gif
    new file:   images/desktop.ini
    new file:   images/line.gif
    new file:   images/pitch.jpg
    new file:   images/thumb.gif
    new file:   index.html

Alejandra@DESKTOP-7JRG0FQ MINGW64 ~/Desktop/Git/ProyectoWeb (master)
$ git commit -m 'primer upload del proyecto'

*** 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 'Alejandra@DESKTOP-7JRG0FQ.(none)')

Alejandra@DESKTOP-7JRG0FQ MINGW64 ~/Desktop/Git/ProyectoWeb (master)
$ git config --global user.name "Alejandra Andrade"

Alejandra@DESKTOP-7JRG0FQ MINGW64 ~/Desktop/Git/ProyectoWeb (master)
$ git config --global user.email "<aleandrader@gmail.com>"

Alejandra@DESKTOP-7JRG0FQ MINGW64 ~/Desktop/Git/ProyectoWeb (master)
```

Agregando el proyecto directamente al repositorio git, localmente en el ordenador.

\$git commit -m 'primer commit'

```
Alejandra@DESKTOP-7JRG0FQ MINGW64 ~/Desktop/Git/ProyectoWeb (master)
$ git commit -m 'primer upload del proyecto'
[master (root-commit) 53c6af7] primer upload del proyecto
8 files changed, 174 insertions(+)
create mode 100644 css/screen.css
create mode 100644 images/body.gif
create mode 100644 images/content.gif
create mode 100644 images/desktop.ini
create mode 100644 images/line.gif
create mode 100644 images/pitch.jpg
create mode 100644 images/thumb.gif
create mode 100644 index.html

Alejandra@DESKTOP-7JRG0FQ MINGW64 ~/Desktop/Git/ProyectoWeb (master)
$ |
```

En resumen.

Los archivos que se encontraban en el directorio ProyectoWeb, se alojaban en el working directory.

Con el comando

\$git add .

Los archivos que se encontraban en el working directory, pasaron al Staging Area (index)

Con el comando

\$git commit -m

Los archivos que estaban en el Staging Area (index), pasaron al Repository (head).

Ahora vamos a enviar los archivos al repositorio remoto, es decir github.

Colocando el origen remoto.

```
$git remote add origin https://github.com/aleandrader/ProyectoACS.git
```

Empujando los archivos en local, al repositorio remoto.

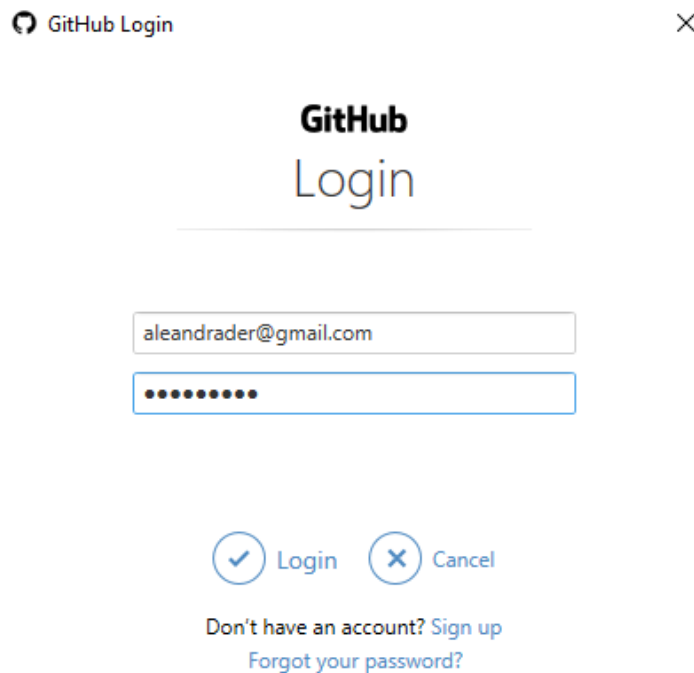
```
$git push -u origin master
```

```
$ git commit -m 'primer upload del proyecto'
[master (root-commit) 53c6af7] primer upload del proyecto
 8 files changed, 174 insertions(+)
 create mode 100644 css/screen.css
 create mode 100644 images/body.gif
 create mode 100644 images/content.gif
 create mode 100644 images/desktop.ini
 create mode 100644 images/line.gif
 create mode 100644 images/pitch.jpg
 create mode 100644 images/thumb.gif
 create mode 100644 index.html

Alejandra@DESKTOP-7JRG0FQ MINGW64 ~/Desktop/Git/ProyectoWeb (master)
$ git remote add origin https://github.com/aleandrader/ProyectoACS.git

Alejandra@DESKTOP-7JRG0FQ MINGW64 ~/Desktop/Git/ProyectoWeb (master)
$ git push -u origin master
```

Una vez ingresadas las instrucciones que identifican que un desarrollador quiere subir sus contribuciones a un repositorio remoto, se abre una ventana de git hub.



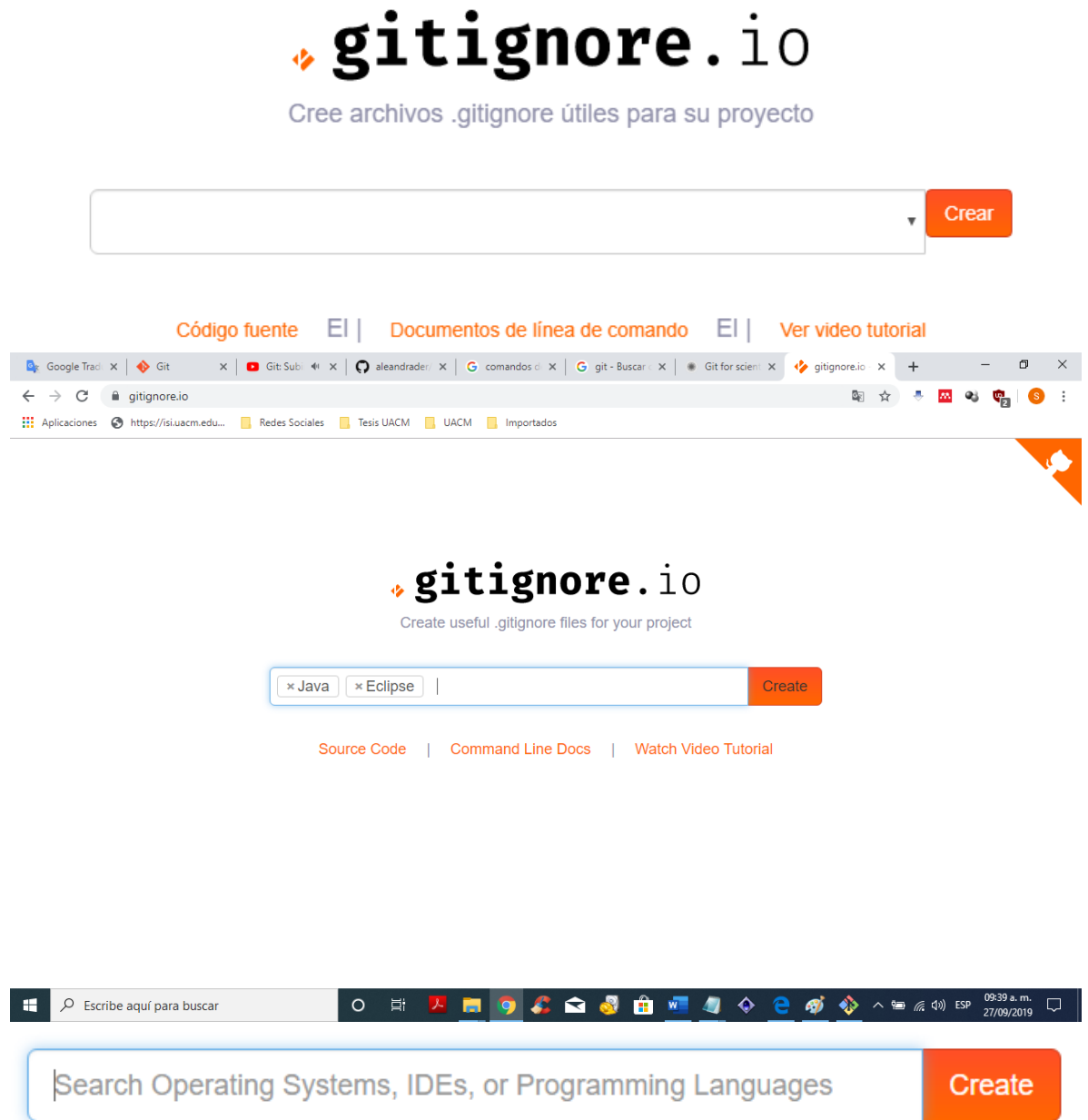
```
$ git remote add origin https://github.com/aleandrader/ProyectoACS.git  
Alejandra@DESKTOP-7JRG0FQ MINGW64 ~/Desktop/Git/ProyectoWeb (master)  
$ git push -u origin master  
Enumerating objects: 12, done.  
Counting objects: 100% (12/12), done.  
Delta compression using up to 2 threads  
Compressing objects: 100% (10/10), done.  
Writing objects: 100% (12/12), 56.17 KiB | 1.75 MiB/s, done.  
Total 12 (delta 1), reused 0 (delta 0)  
remote: Resolving deltas: 100% (1/1), done.  
To https://github.com/aleandrader/ProyectoACS.git  
* [new branch]      master -> master  
Branch 'master' set up to track remote branch 'master' from 'origin'.  
Alejandra@DESKTOP-7JRG0FQ MINGW64 ~/Desktop/Git/ProyectoWeb (master)  
$ |
```

Vamos a revisar ahora la salida de github

The screenshot shows the GitHub interface for the repository 'aleandrader / ProyectoACS'. At the top, there are navigation links for 'Code', 'Issues', 'Pull requests', 'Projects', 'Wiki', 'Security', 'Insights', and 'Settings'. Below this, a message states: 'Este repositorio es creado para alojar el proyecto de la materia Aseguramiento de la Calidad de Software que se imparte en el sexto semestre de la LIS en la UACM.' followed by an 'Edit' button. A progress bar shows '1 commit', '1 branch', '0 releases', and '1 contributor'. Below the progress bar, there are buttons for 'Branch: master', 'New pull request', 'Create new file', 'Upload files', 'Find File', and 'Clone or download'. A list of files is shown: 'css', 'images', and 'index.html', each with a note 'primer upload del proyecto' and a timestamp '18 minutes ago'. At the bottom, there is a prompt to 'Add a README'.

Tu repositorio ya tiene una nueva contribución de un desarrollador.

Si utilizas dependencias puedes crear un archivo, para generar un archivo git ignore



Agrega lenguaje, entornos de desarrollo

Creando el código del archivo git ignore.

```
# Created by https://www.gitignore.io/api/java,eclipse
# Edit at https://www.gitignore.io/?templates=java,eclipse

### Eclipse ###
.metadata
bin/
tmp/
*.tmp
```

```
*.bak
*.swp
*~.nib
local.properties
.settings/
.loadpath
.recommenders

# External tool builders
.externalToolBuilders/

# Locally stored "Eclipse launch configurations"
*.launch

# PyDev specific (Python IDE for Eclipse)
*.pydevproject

# CDT-specific (C/C++ Development Tooling)
.cproject

# CDT- autotools
.autotools

# Java annotation processor (APT)
.factorypath

# PDT-specific (PHP Development Tools)
.buildpath

# sbteclipse plugin
.target

# Tern plugin
.tern-project

# TeXlipse plugin
.texlipse

# STS (Spring Tool Suite)
.springBeans

# Code Recommenders
.recommenders/

# Annotation Processing
.appt_generated/

# Scala IDE specific (Scala & Java development for Eclipse)
.cache-main
.scala_dependencies
.worksheet

### Eclipse Patch ###
# Eclipse Core
.project

# JDT-specific (Eclipse Java Development Tools)
```

```
.classpath

# Annotation Processing
.appt_generated

.sts4-cache/

### Java ###
# Compiled class file
*.class

# Log file
*.log

# BlueJ files
*.ctxt

# Mobile Tools for Java (J2ME)
.mtj.tmp/

# Package Files #
*.jar
*.war
*.nar
*.ear
*.zip
*.tar.gz
*.rar

# virtual machine crash logs, see
http://www.java.com/en/download/help/error_hotspot.xml
hs_err_pid*

# End of https://www.gitignore.io/api/java,eclipse
```

Dentro de tu directorio de trabajo o work directory genera un archivo .gitignore

Clonar un directorio remoto de gitHub.

\$git clone <https://github.com/aleandrader/ProyectoACS.git>

UACM, Aseguramiento de la Calidad del Software

Dra. Alejandra Andrade

Este repositorio es creado para alojar el proyecto de la materia Aseguramiento de la Calidad de Software que se imparte en el sexto semestre de la LIS en la UACM.

Edit

[Manage topics](#)

1 commit

1 branch

0 releases

1 contributor

Branch: master

New pull request

Create new file

Upload files

Find File

Clone or download

aleandrader

primer upload del proyecto

css

primer upload del proyecto

images

primer upload del proyecto

index.html

primer upload del proyecto

Clone with HTTPS

Use Git or checkout with SVN using the web URL.

https://github.com/aleandrader/ProyectoA

Open in Desktop

Download ZIP

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

Add a README

Video Tutorial Recomendado.

<https://www.youtube.com/watch?v=RRegIKu-z3k>

Ejercicio clonar un directorio

Paso 1. Situarse en la carpeta que se va a clonar el proyecto.

```
Alejandra@DESKTOP-7JRG0FQ MINGW64 ~/Desktop
$ pwd
/c/Users/Alejandra/Desktop

Alejandra@DESKTOP-7JRG0FQ MINGW64 ~/Desktop
$ cd prueba\ clona/

Alejandra@DESKTOP-7JRG0FQ MINGW64 ~/Desktop/prueba clona
$ ls
```

```
Alejandra@DESKTOP-7JRG0FQ MINGW64 ~/Desktop/prueba clona
$ git clone https://github.com/aleandrader/ProyectoACS.git
Cloning into 'ProyectoACS'...
remote: Enumerating objects: 12, done.
remote: Counting objects: 100% (12/12), done.
remote: Compressing objects: 100% (9/9), done.
remote: Total 12 (delta 1), reused 12 (delta 1), pack-reused 0
Unpacking objects: 100% (12/12), done.

Alejandra@DESKTOP-7JRG0FQ MINGW64 ~/Desktop/prueba clona
$
```