



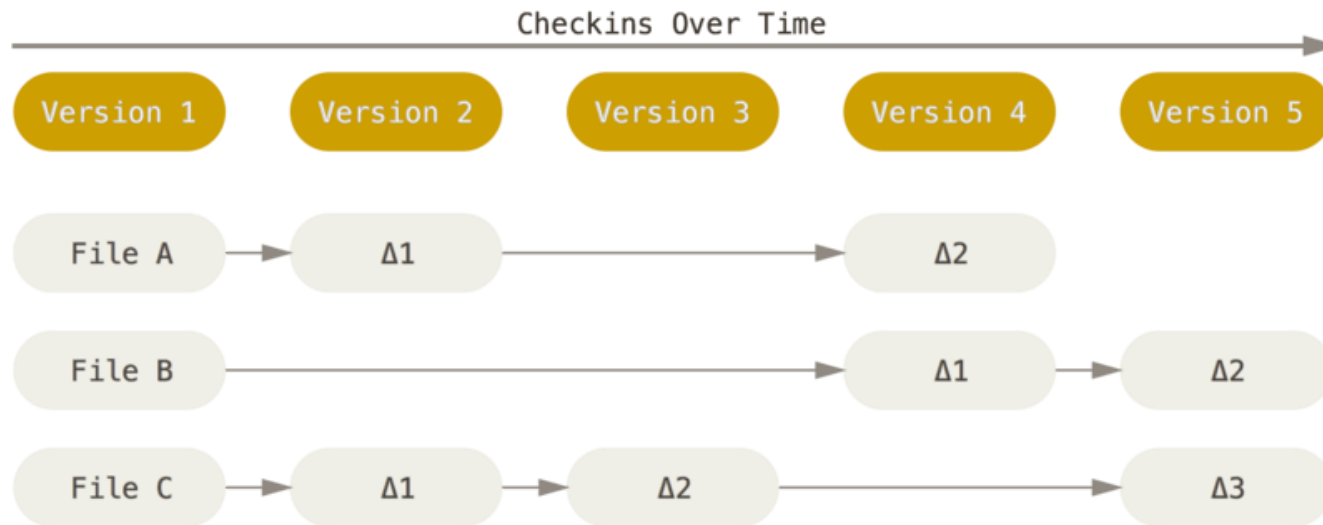
A horizontal bar chart with a single grey bar. The bar is positioned in the middle of the slide and extends across most of the width. To the right of the bar, the word "GIT" is written in white capital letters.

GIT

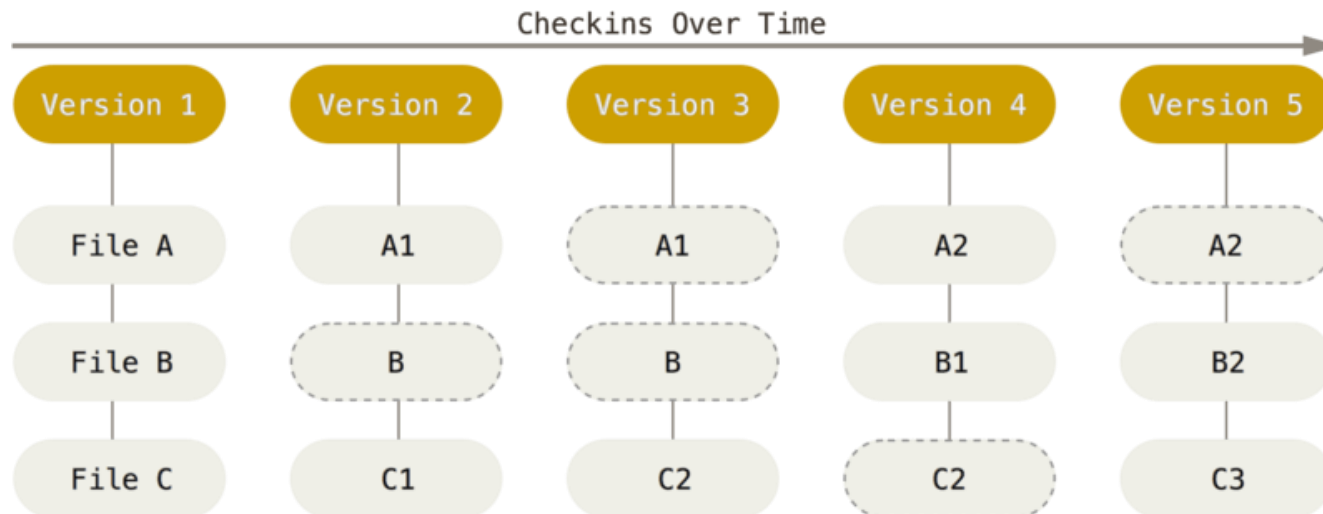
# CONTENIDOS

1. Fundamentos de Git
2. GitHub

# Copias instantáneas vs diferencias



CVS, Subversion, Perforce,...  
información como un  
conjunto de archivos  
y modificaciones en  
el tiempo



Git  
conjunto de copias  
instantáneas de un  
sistema de archivos  
miniatura

# Operaciones locales con integridad

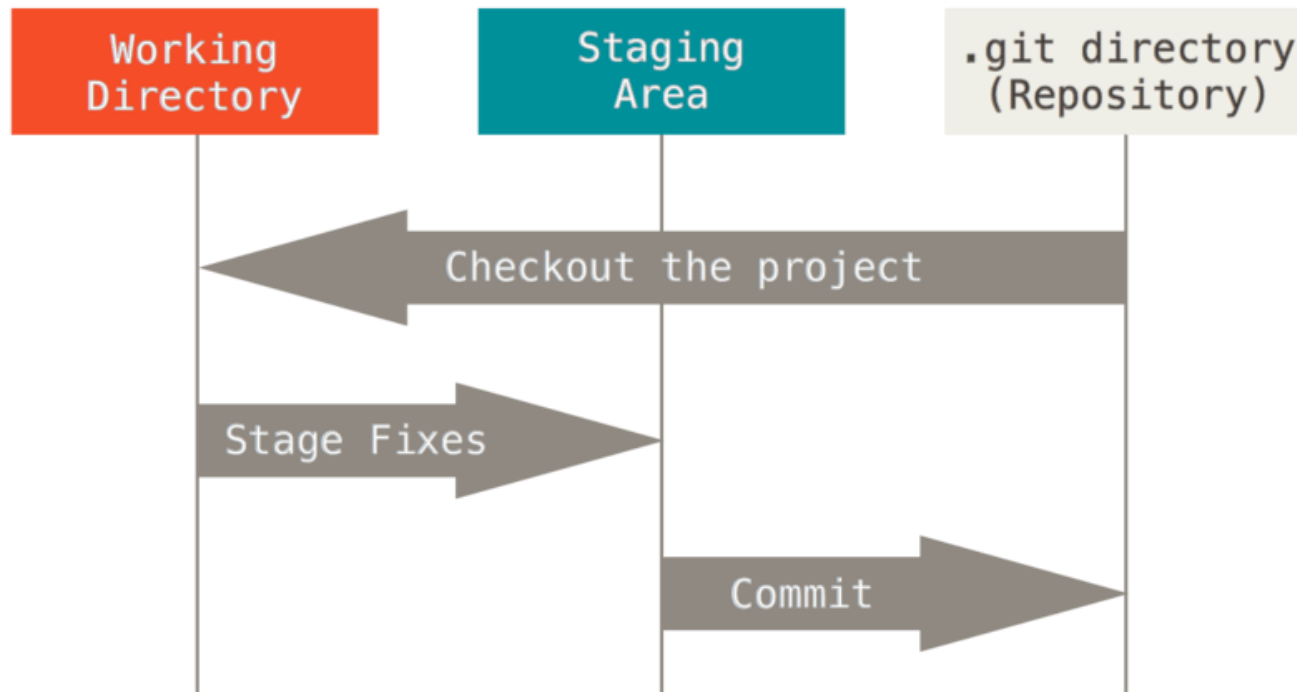
- Se trabaja en la copia local, acceso inmediato
- Permite trabajar sin conexión al repositorio
- Uso de sumas de chequeo\* (*checksum*) para comprobar la integridad de los datos.
  - *\*función hash que detecta cambios en una secuencia de datos entre la verificación inicial y la final.*

# Tres estados de los ficheros

- **confirmado** (committed) significa que los datos están almacenados de manera segura en tu base de datos local.
- **modificado** (modified) significa que has modificado el archivo pero todavía no lo has confirmado a tu base de datos.
- **preparado** (staged) significa que has marcado un archivo modificado en su versión actual para que vaya en tu próxima confirmación.

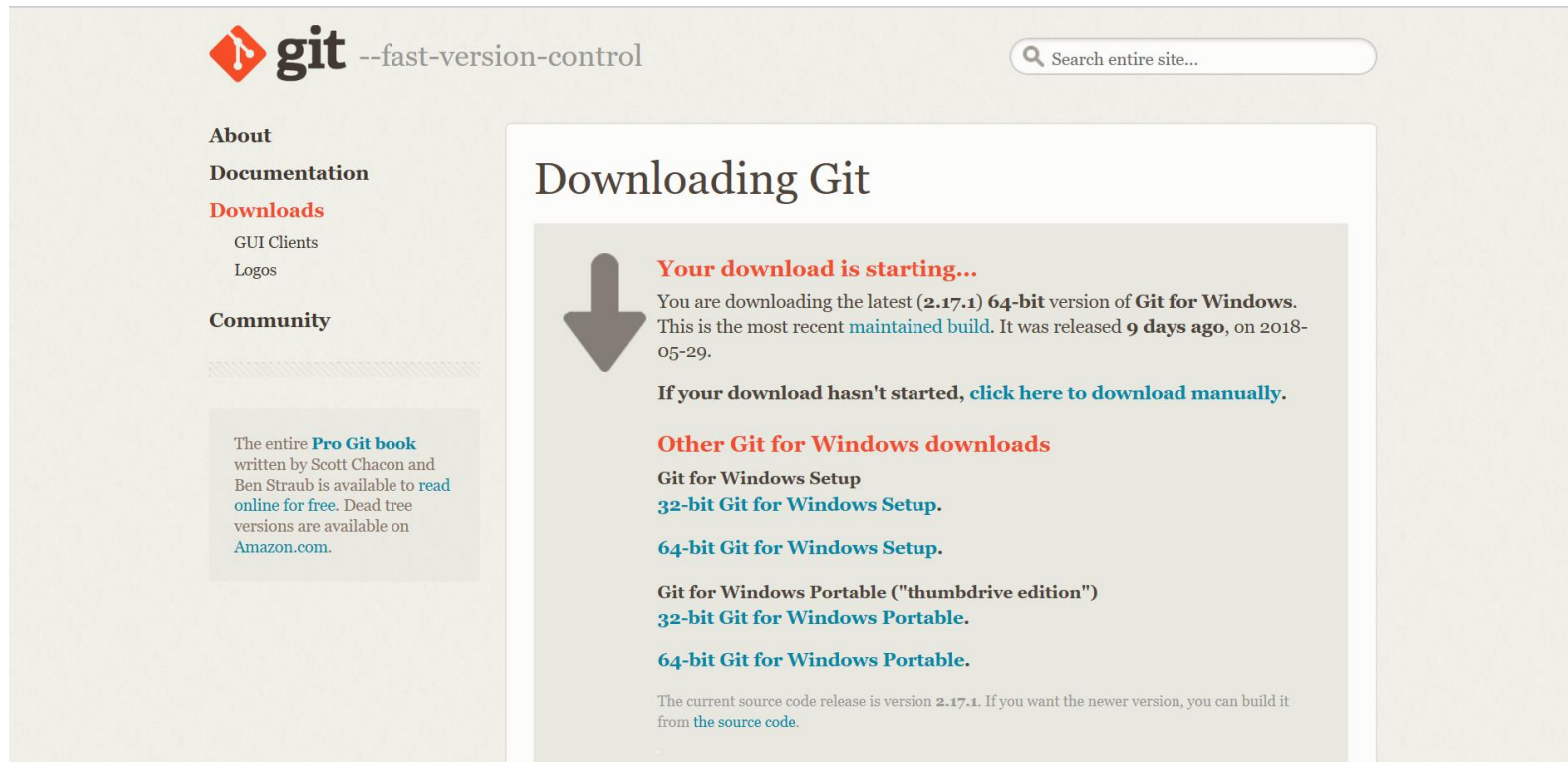
# Tres secciones

- El directorio de Git (Git directory): almacena metadatos y bd de objetos para el proyecto, es lo que se copia al clonar un repositorio
- El directorio de trabajo (working directory) copia de trabajo.
- El área de preparación (staging area) almacena información de lo que irá en la próxima confirmación (*index*)



# Instalación de Git en Windows

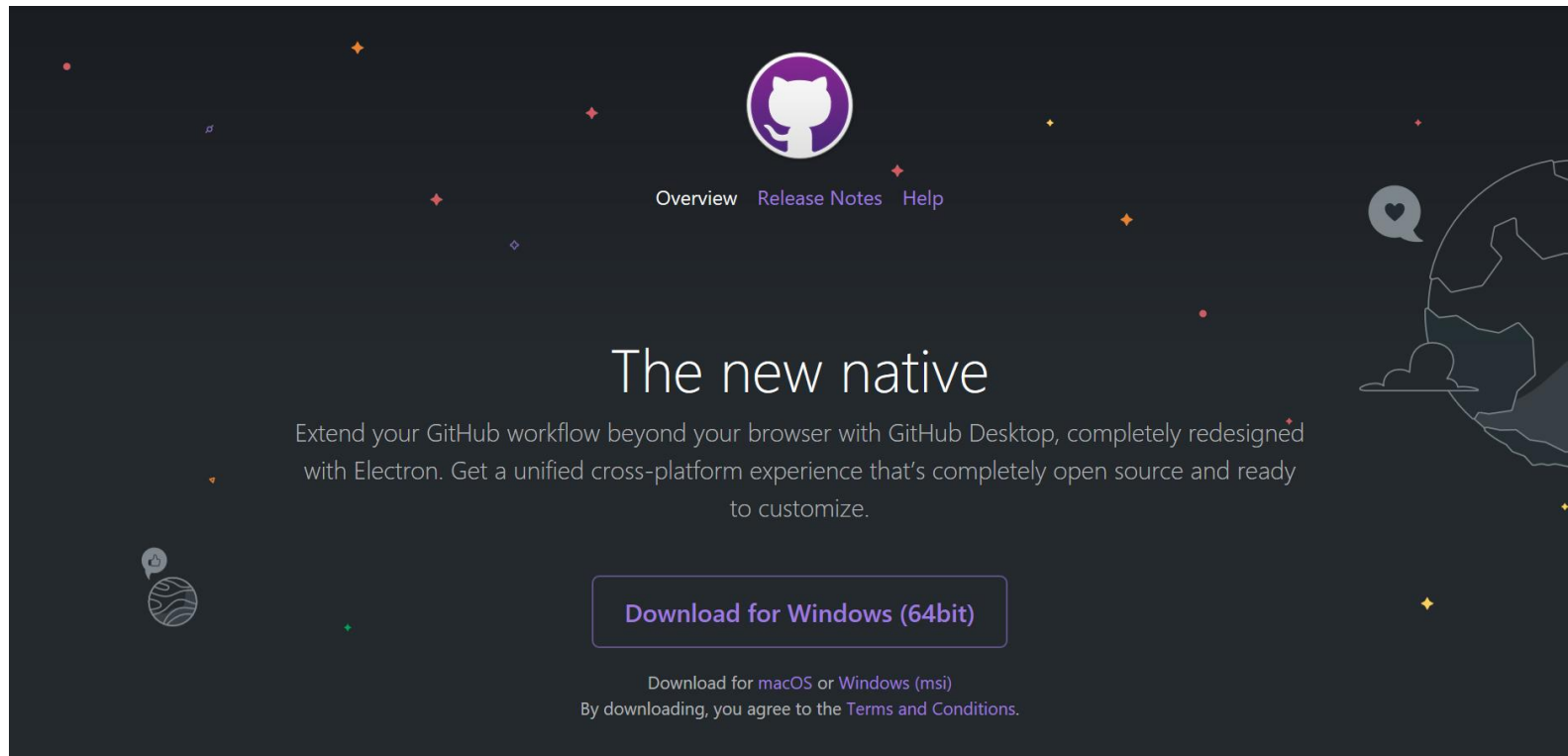
- Descargar el instalador del sitio oficial de git para Windows en <http://git-scm.com/download/win>.



The screenshot shows the Git website's 'Downloading Git' page. The header features the Git logo and the tagline '--fast-version-control'. A search bar is located in the top right corner. The left sidebar contains navigation links: 'About', 'Documentation', 'Downloads' (highlighted in red), 'GUI Clients', 'Logos', and 'Community'. Below these links is a box mentioning the 'Pro Git book' by Scott Chacon and Ben Straub, available online for free. The main content area is titled 'Downloading Git' and features a large downward arrow icon. The text states: 'Your download is starting... You are downloading the latest (2.17.1) 64-bit version of Git for Windows. This is the most recent maintained build. It was released 9 days ago, on 2018-05-29.' Below this, it says: 'If your download hasn't started, click here to download manually.' A section titled 'Other Git for Windows downloads' lists: 'Git for Windows Setup', '32-bit Git for Windows Setup.', '64-bit Git for Windows Setup.', 'Git for Windows Portable ("thumbdrive edition")', '32-bit Git for Windows Portable.', and '64-bit Git for Windows Portable.'. At the bottom, it notes: 'The current source code release is version 2.17.1. If you want the newer version, you can build it from the source code.'

# Instalación de github en Windows

- Puedes descargar este instalador del sitio web de GitHub para Windows en <http://windows.github.com>.





# Configuración de git

- Descargar el instalador del sitio oficial de git para Windows en <http://git-scm.com/download/win>
- Seleccionar todas las opciones que nos ofrece por defecto
- Abrir el editor git\_bash y configurar el nombre de usuario y el email.  
Importante para la información que se envía de los *commits*:

```
$ git config --global user.name "John Doe"

$ git config --global user.email johndoe@example.com
```

- Con esto se modifica el archivo de configuración: **.gitconfig**

# Obtener un repositorio Git

- Tomar un proyecto o directorio existente e importarlo en Git
  - Situarse en el directorio del proyecto y ejecutar:

```
$ git init
```

- Se genera un subdirectorio `.git` que contiene el esqueleto del repositorio.

- Controlar versiones de archivos existentes :

```
$ git add *.php  
$ git add LICENSE  
$ git commit -m 'initial project version'
```

# Obtener un repositorio Git

- Clonar un repositorio de git:

- Ej.: 

```
$ git clone https://github.com/libgit2/libgit2
```

# Guardando cambios en el Repositorio

- Dentro del repositorio hay archivos rastreados (añadidos al repositorio) o sin rastrear (el resto de archivos)
- Para saber el estado de los archivos:

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

```
rglep@RGL-LAP MINGW64 /c/xampp/htdocs/proyectos/login_restful (master)
$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    new file:   Rest.php

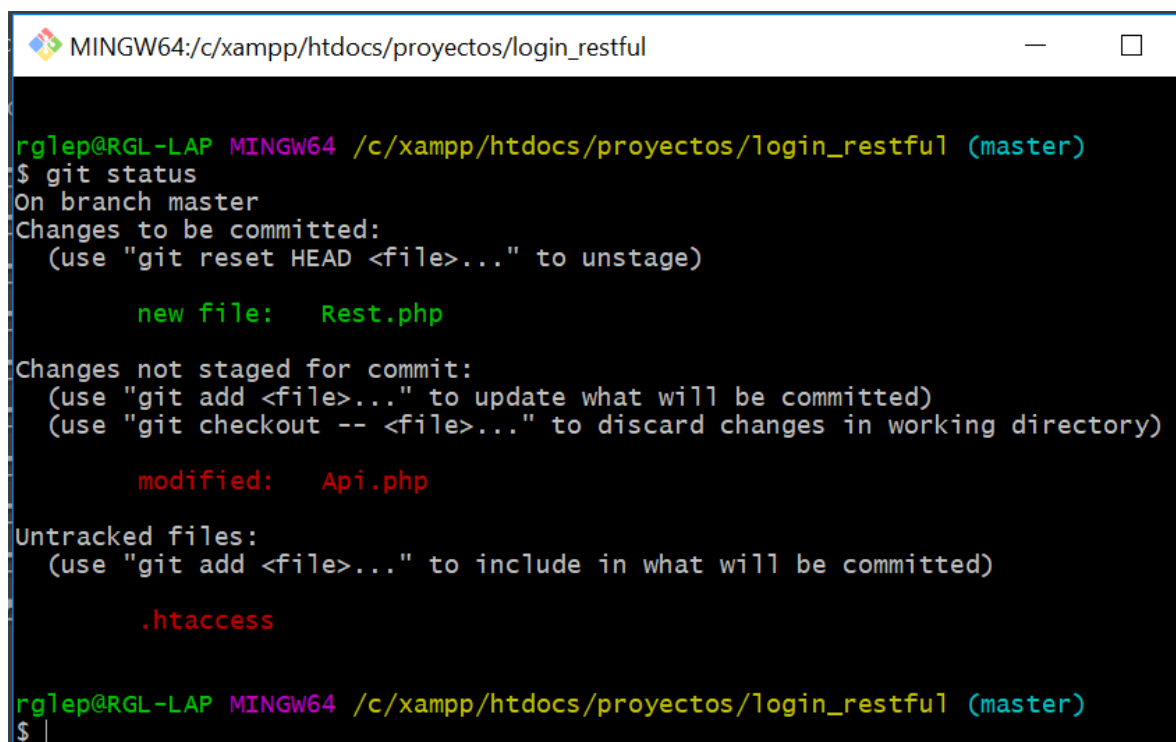
Untracked files:
  (use "git add <file>..." to include in what will be committed)

    .htaccess
```

# Para rastrear nuevos archivos o preparar cambios

```
$ git add .htaccess
```

- Modificar algo del archivo y volver a ejecutar \$git status:



```
MINGW64:/c:/xampp/htdocs/proyectos/login_restful

rglep@RGL-LAP MINGW64 /c:/xampp/htdocs/proyectos/login_restful (master)
$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    new file:   Rest.php

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:   Api.php

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

    .htaccess

rglep@RGL-LAP MINGW64 /c:/xampp/htdocs/proyectos/login_restful (master)
$
```

# Para persistir los cambios

```
$ git commit
```

- Opciones: `git commit -a` (salta el modo preparación)
- Opción `-m` 'incluir mensaje de confirmación'
- *(para salir del editor **vim** guardando los cambios usar ZZ)*

MINGW64:/c:/xampp/htdocs/proyectos/login\_restful

```
Please enter the commit message for your changes. Lines starting
with '#' will be ignored, and an empty message aborts the commit.
```

```
On branch master
```

```
Changes to be committed:
```

```
    new file:   Rest.php
```

```
Changes not staged for commit:
```

```
    modified:   Api.php
```

```
Untracked files:
```

```
    .htaccess
```

```
proyectos/login_restful/.git/COMMIT_EDITMSG [unix] (22:19 07/06/2018)1,0-1 Tod
tdocs/proyectos/login_restful/.git/COMMIT_EDITMSG" [UNIX] 14L, 299C
```

### Para ver el detalle de los cambios

```
$ git diff
```

### Para ignorar archivos

- Crear un archivo .gitignore

```
$ cat .gitignore  
*.[oa]  
*~
```

### Para eliminar archivos

```
$ rm README.md
```

```
$ git rm README.md
```

# Ramas

```
$ git branch testing
```

- Crea una nueva rama pero no salta a ella
- Para saltar a la nueva rama:

```
$ git checkout testing
```

- Utilizando un atajo para los dos pasos:

```
$ git checkout -b iss53  
Switched to a new branch "iss53"
```



# Fusionar Ramas

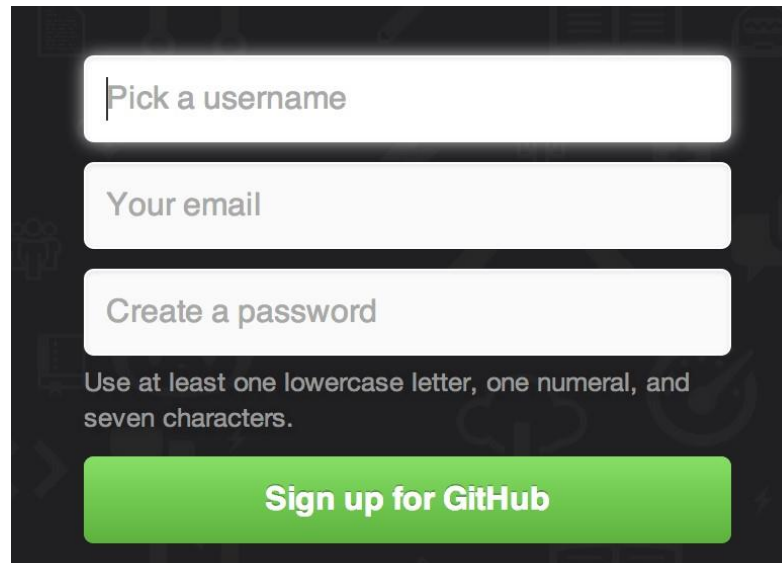
```
$ git checkout master  
$ git merge hotfix
```

# Borrar Ramas

```
$ git branch -d hotfix  
Deleted branch hotfix (3a0874c).
```

# Creación y configuración de la cuenta

- GitHub es el mayor proveedor de alojamiento de repositorios Git
- Visita <https://github.com>, elige un nombre de usuario que no esté ya en uso, proporciona un correo y una contraseña, y pulsa el botón verde grande “Sign up for GitHub”.

A screenshot of the GitHub sign-up form. It features three white input fields on a dark background. The first field is labeled 'Pick a username', the second 'Your email', and the third 'Create a password'. Below the password field, there is a text requirement: 'Use at least one lowercase letter, one numeral, and seven characters.' At the bottom of the form is a large green button with the text 'Sign up for GitHub' in white.

- .GitHub te enviará un correo para verificar la dirección que les has dado

# Comenzar a trabajar

- Pulsando en el logo del gato con patas de pulpo en la parte superior izquierda de la pantalla llegarás a tu escritorio principal. Ahora ya estás listo para comenzar a usar GitHub.

- **Acceso SSH**

Desde ya, puedes acceder a los repositorios Git utilizando el protocolo `https://`, identificándote con el usuario y la contraseña que acabas de elegir. Sin embargo, para simplificar el clonado de proyectos públicos, no necesitas crearte la cuenta. Es decir, la cuenta solo la necesitas cuando comienzas a hacer cosas como bifurcar (fork) proyectos y enviar tus propios cambios más tarde.

- **Autenticación de dos pasos**

Para mayor seguridad, deberías configurar la Autenticación de Dos Pasos o “2FA”. Este tipo de autenticación se está haciendo más popular para reducir el riesgo de que te roben la cuenta. Al activarla, GitHub te pedirá identificarte de dos formas, de forma que si una de ellas resulta comprometida, el atacante no conseguirá acceso a tu cuenta.

Puedes encontrar la configuración de “2FA” en la opción Security de los ajustes de la cuenta

# Participando en Proyectos

### ▫ **Bifurcación (fork) de proyectos**

Para participar en un proyecto existente, en el que no tengas permisos de escritura, puedes bifurcarlo (hacer un “fork”). Esto consiste en crear una copia completa del repositorio totalmente bajo tu control: se encontrará en tu cuenta y podrás escribir en él sin limitaciones.

Para bifurcar un proyecto, visita la página del mismo y pulsa sobre el botón “Fork” del lado superior derecho de la página.



# Participando en Proyectos

## ▫ El Flujo de Trabajo en GitHub

1. Se crea una rama a partir de master.
2. Se realizan algunos commits hacia esa rama.
3. Se envía esa rama hacia tu copia (fork) del proyecto.
4. Abres un Pull Request en GitHub.
5. Se participa en la discusión asociada y, opcionalmente, se realizan nuevos commits.
6. El propietario del proyecto original cierra el Pull Request, bien fusionando la rama con tus cambios o bien rechazándolos.

# Participando en Proyectos

- **Ejemplo: Clonar proyecto store-and-retrieve-image-in-blob del repositorio BCNprogramming**

# Mantenimiento de un proyecto

## ▫ Creación de un repositorio

.Comienza pulsando el botón “New repository” en el lado derecho de tu página principal, o bien desde el botón + en la barra de botones cercano a tu nombre de usuario, tal como se ve en Desplegable “New repository”..

## ▫ Añadir colaboradores

Selecciona “Collaborators” del menú del lado izquierdo. Simplemente, teclea el usuario en la caja, y pulsa en “Add collaborator.” Puedes repetir esto las veces que necesites para dar acceso a otras personas. Si necesitas quitar un acceso, pulsa en la “X” del lado derecho del usuario.

# Trabajar con repositorio remoto

- Para añadir un remoto nuevo y asociarlo a un nombre que puedas referenciar fácilmente, ejecuta `git remote add [nombre] [url]`:
- Para obtener los datos de los proyectos remotos:

```
$ git fetch [remote-name]
```

- El comando `git pull` para traer y combinar automáticamente la rama remota con tu rama actual.
- Enviar a los Remotos

```
$ git push origin master
```