



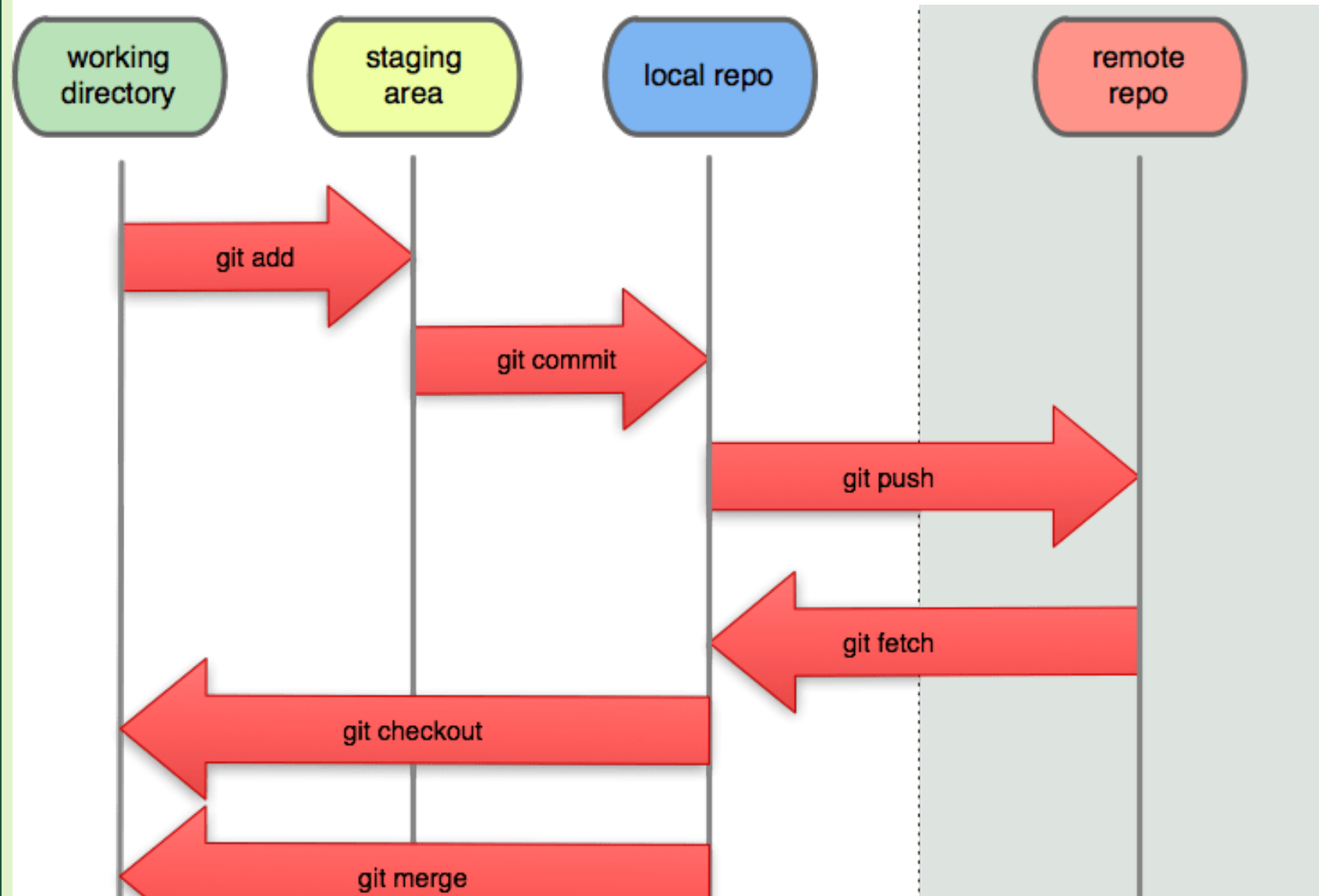
EFA  
MORATALAZ

*1º CFGS Desarrollo de  
Aplicaciones Multiplataforma*

## *ENTORNOS DE DESARROLLO*

*YOLANDA MORENO G<sup>a</sup>-MAROTO*

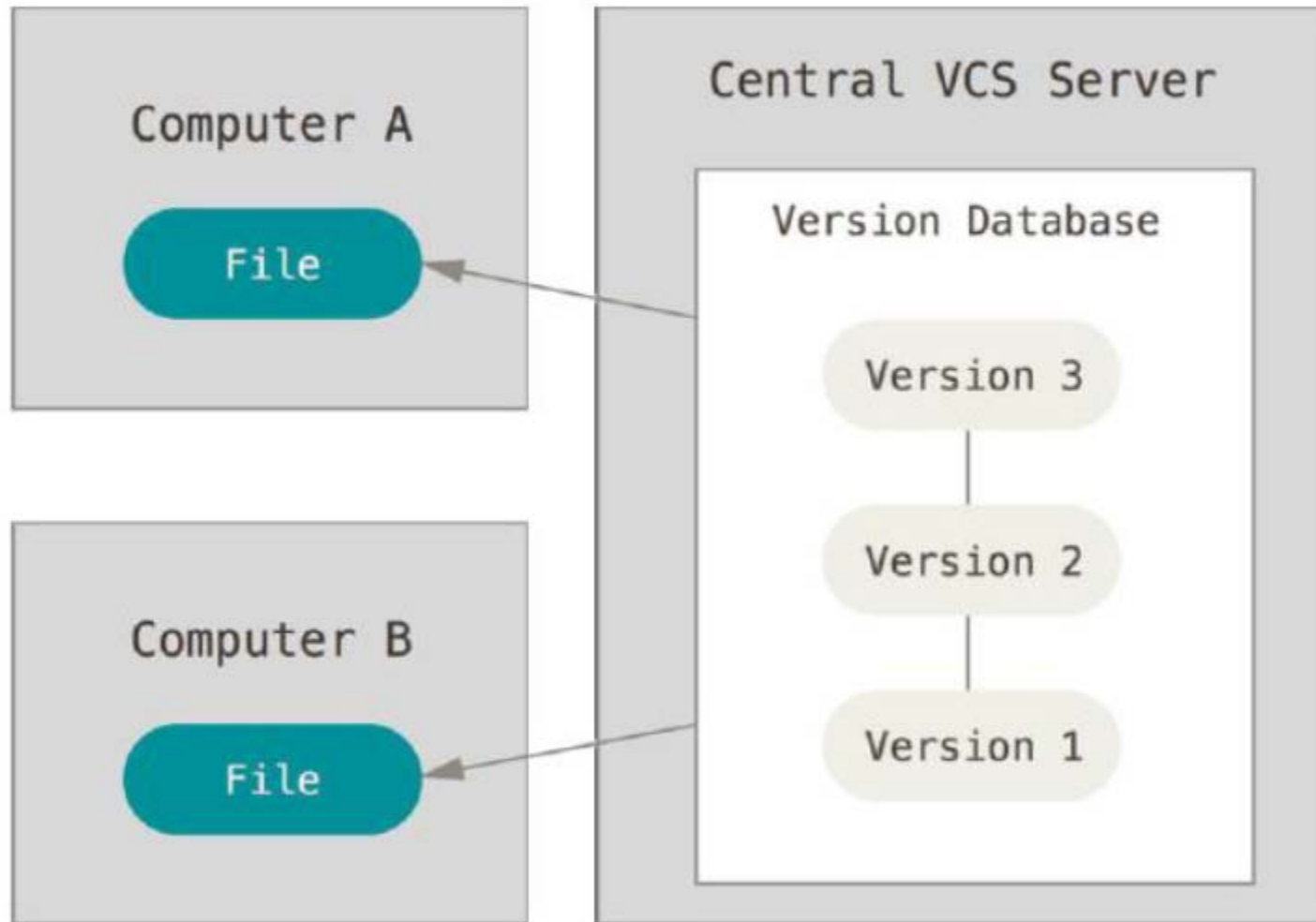
## UT4.4 – CONTROL DE VERSIONES GIT



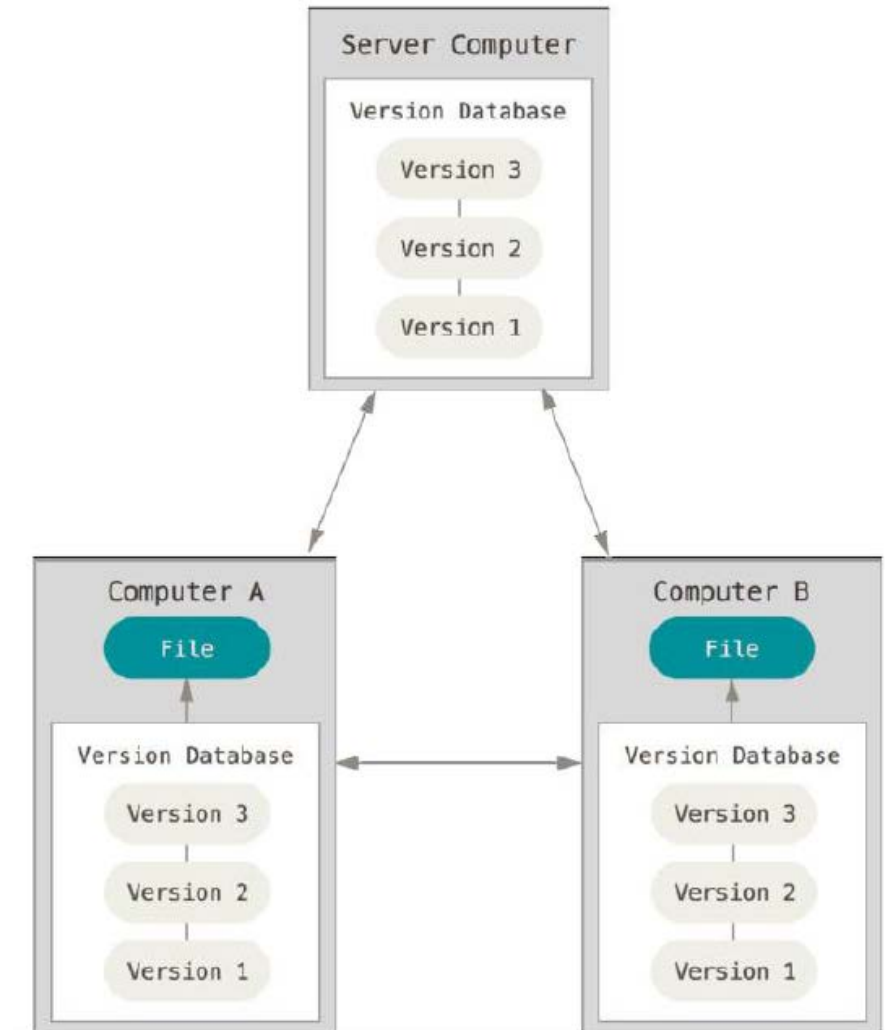
# Introducción a Git:

- Es un Sistema de Control de Versiones distribuido
- Data del año 2005.
- Proyecto de Linus Torvalds(creador de Linux).
- Comenzó siendo el sistema de control de versiones para el Kernel de Linux.
- Actualmente herramienta gratuita y software libre (licencia GNU General PublicLicenseversion2.0).

# Sistemas de control de versiones centralizados:



# Sistemas de control de versiones distribuidos:



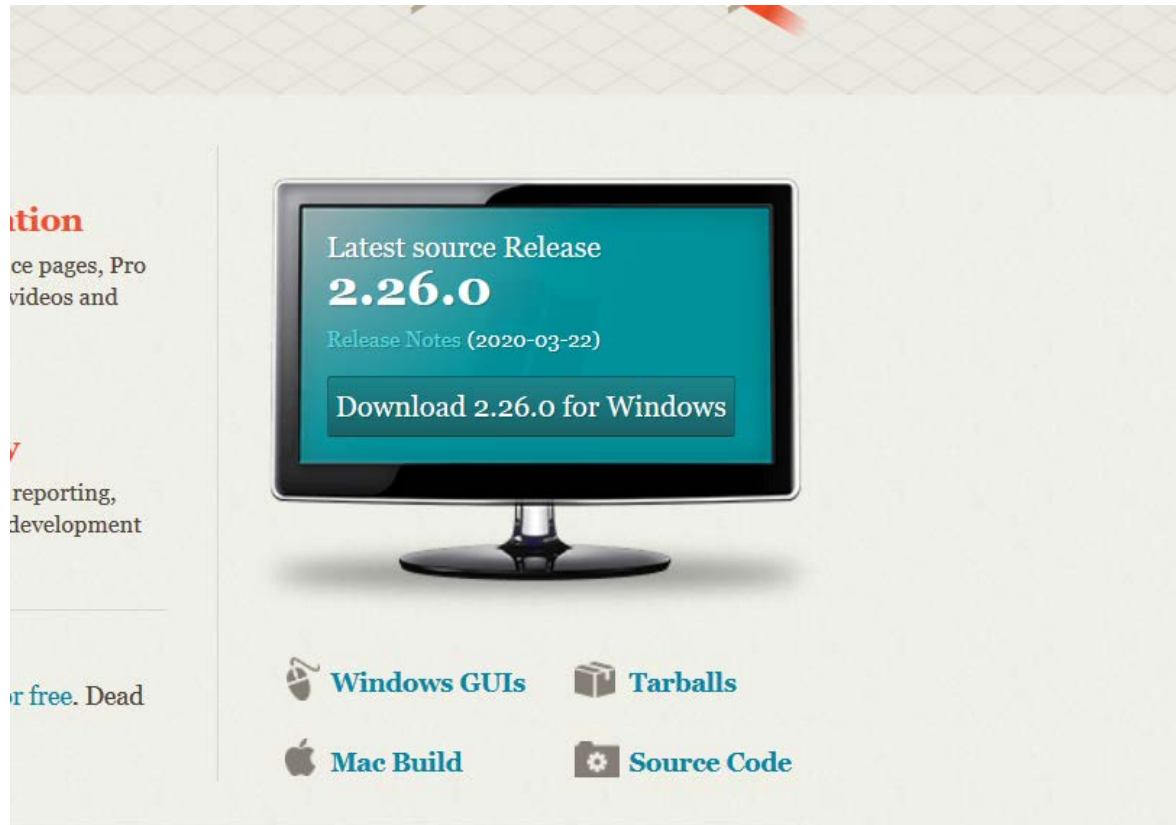
# Características de Git:

- Velocidad
- Diseño sencillo
- Multiplataforma
- Gran soporte al desarrollo no lineal (miles de ramas)
- Completamente distribuido
- Capaz de trabajar con grandes proyectos eficientemente

# Instalación de Git en Windows:

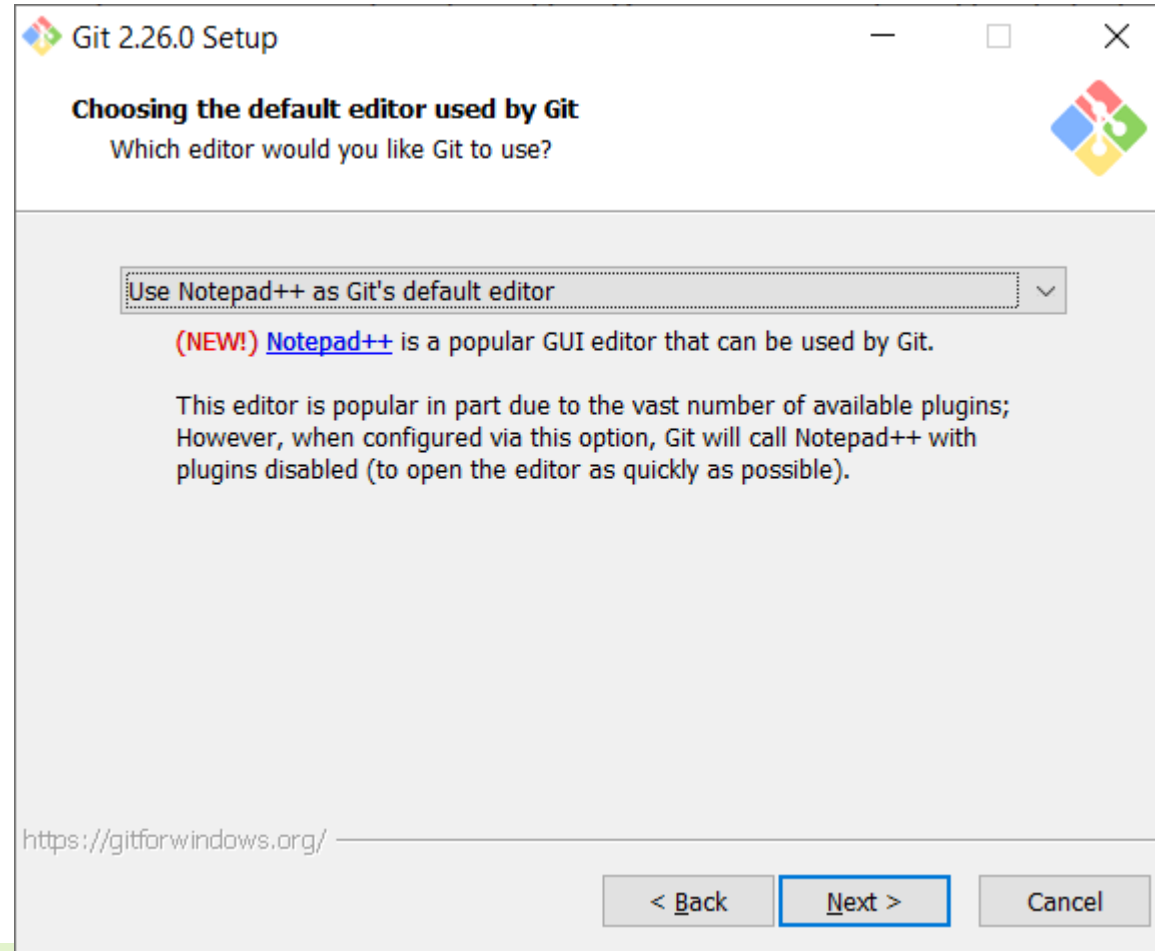
Vamos a la dirección: <https://git-scm.com/>

Instalamos la última versión:

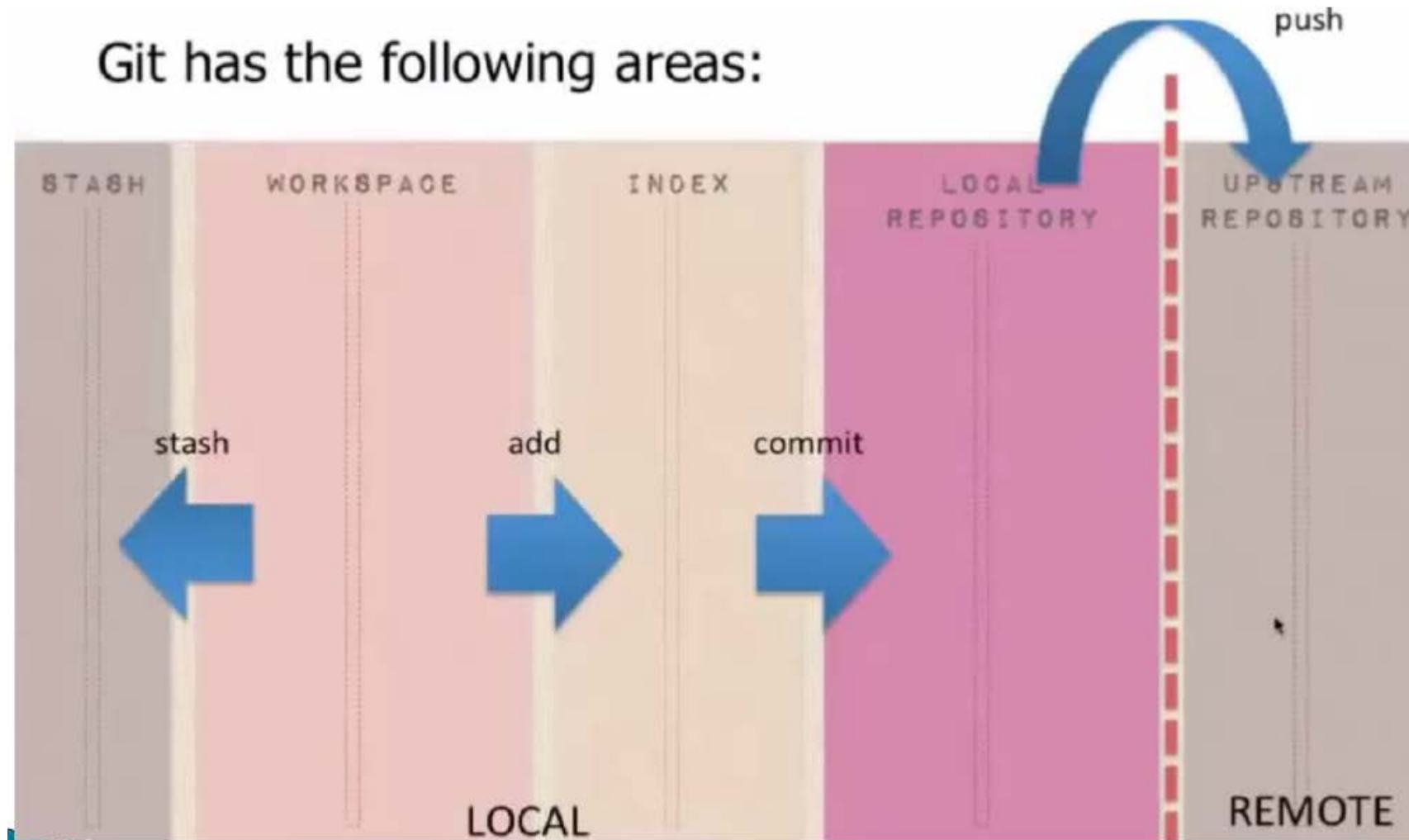


# Instalación de Git en Windows:

Seguiremos los pasos de la instalación y en el paso del editor por defecto podemos elegir el que queramos de los que aparecen en la lista:



# Áreas de trabajo con Git





# Comenzando con Git

Primeros pasos, para configurar tu nombre de usuario y un email que aparecerá en los commit que realices:

It's time to configure your settings:

Username

```
$ git config --global user.name "Your Name Here"  
# Sets the default name for git to use when you commit
```

Email

```
$ git config --global user.email "your_email@example.com"  
# Sets the default email for git to use when you commit
```

# Comenzando con Git

Primeros pasos, para configurar tu nombre de usuario y un email que aparecerá en los commit que realices:

```
C:\> Símbolo del sistema

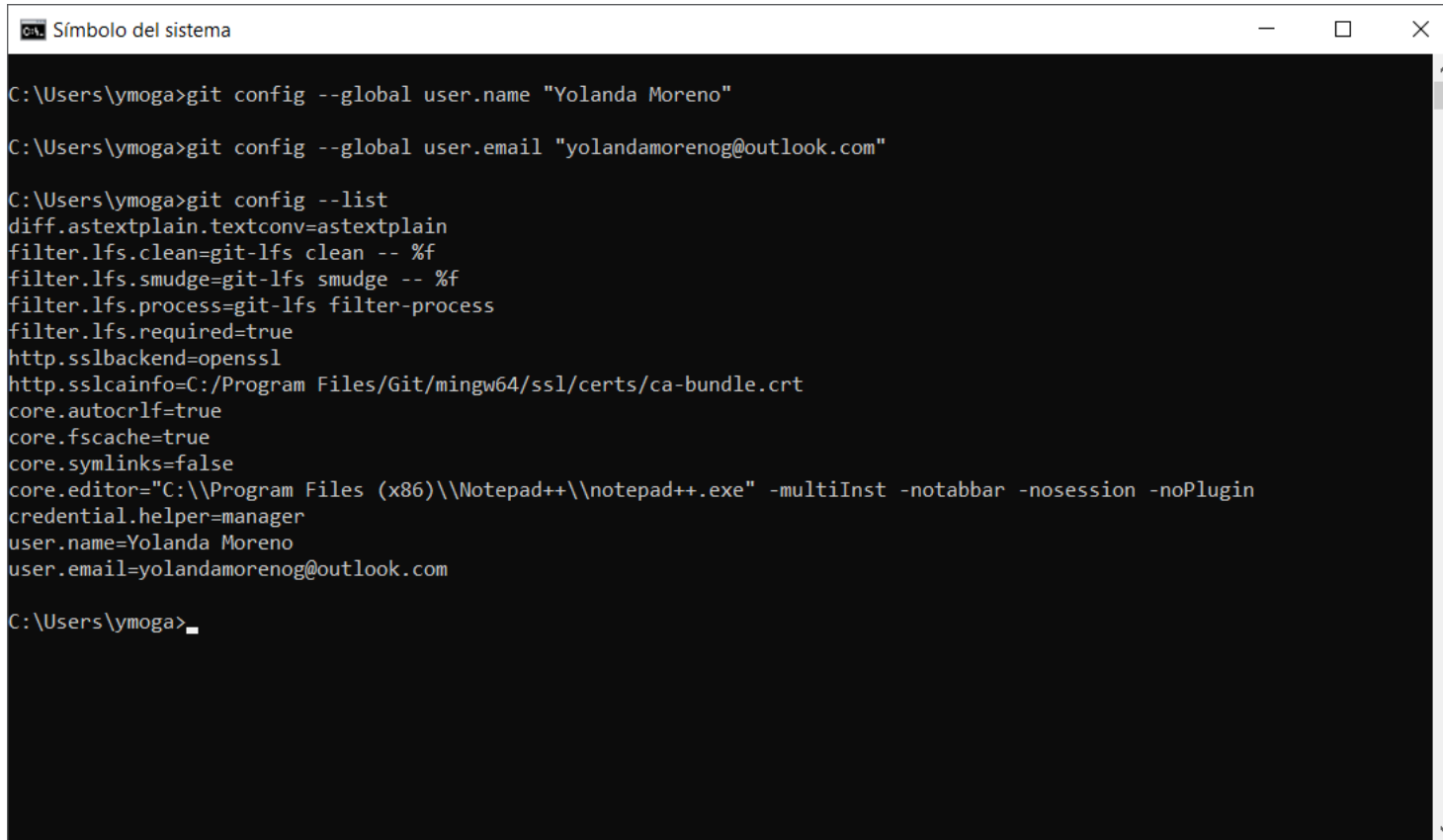
C:\Users\ymoga>git config --global user.name "Yolanda Moreno"

C:\Users\ymoga>git config --global user.email "yolandamorenog@outlook.com"

C:\Users\ymoga>_
```

# Comenzando con Git

Para comprobar la configuración de git: `git config --list`



```
C:\Users\ymoga>git config --global user.name "Yolanda Moreno"

C:\Users\ymoga>git config --global user.email "yolandamorenog@outlook.com"

C:\Users\ymoga>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
core.editor="C:\\Program Files (x86)\\Notepad++\\notepad++.exe" -multiInst -notabbar -nosession -noPlugin
credential.helper=manager
user.name=Yolanda Moreno
user.email=yolandamorenog@outlook.com

C:\Users\ymoga>
```

# Algunos comandos Git

- `git init` → crear repositorio
- `git status` → estado en el que está nuestro repositorio
- `git log` → nos muestra el histórico de commit
- `git log --oneline` → histórico de commit simplificado
- `git add` → añadir fichero/s a la zona de index antes de realizar un commit
- `git remote` → conectar nuestro repositorio local con un repositorio remoto
- `git remote -v` → visualizar los repositorios remotos que tenemos conectados
- `git push` → subir cambios al repositorio remoto
- `git remote rename` → renombrar alias repositorio remoto
- `git pull` → actualizar tu repositorio local a la última versión que hay en el repositorio remoto

# Ejercicio práctico con Git + GitHub

## 1. Trabajando con un proyecto que creamos en local:

- Nos creamos un directorio para nuestros repositorios git.
- Creamos un nuevo proyecto java en el directorio que contenga una clase llamada “ClaseHola” que de momento estará vacía.
- Creamos el repositorio del proyecto: git init (veremos que dentro de la capeta del proyecto se nos ha creado un directorio oculto llamado .git en el que se guardará toda la información del repositorio)

```
Símbolo del sistema
El número de serie del volumen es: F271-2AFF

Directorio de C:\Users\ymoga\Documents\GIT_repo\Proyecto1

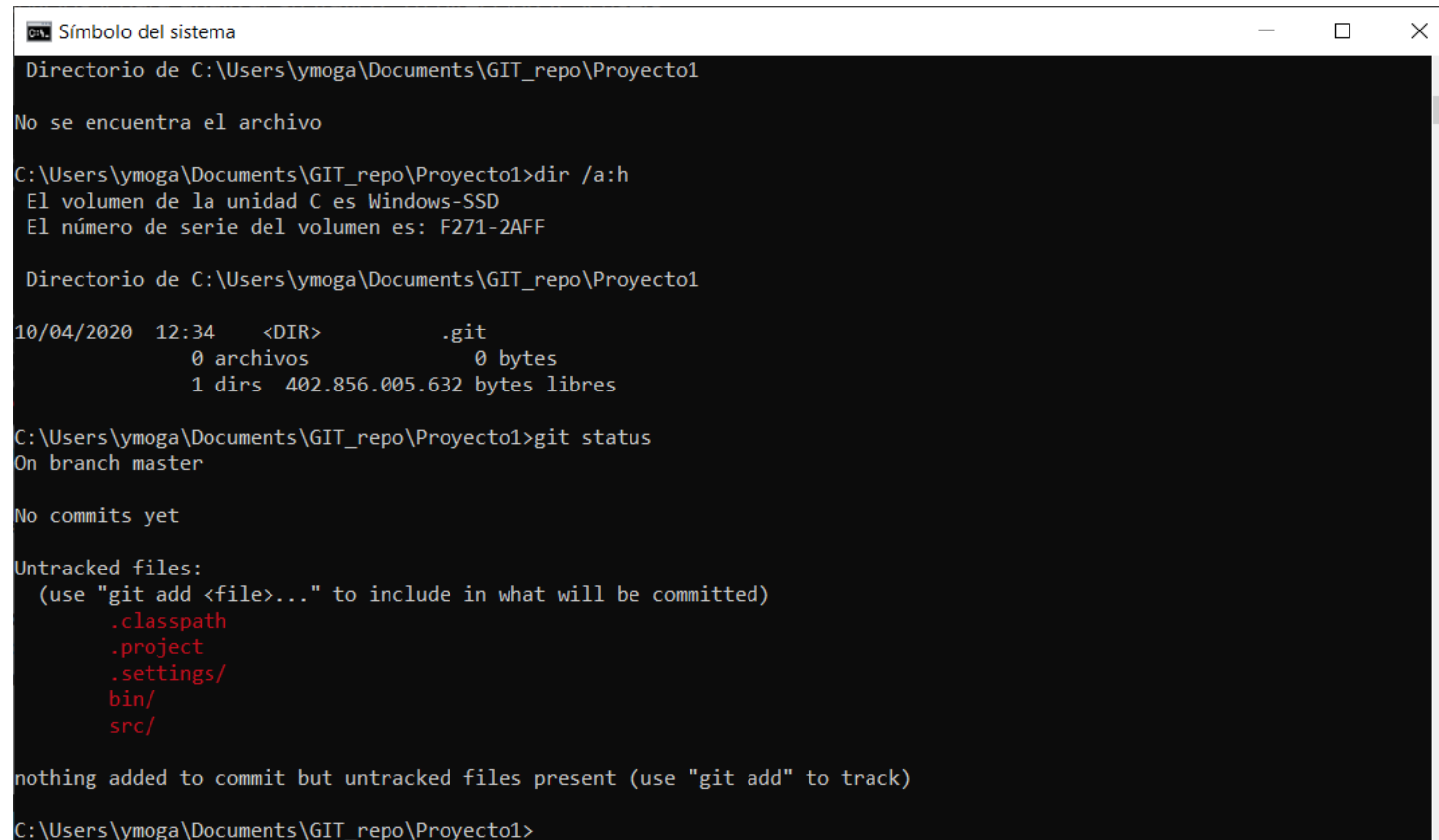
10/04/2020 12:20 <DIR>      .
10/04/2020 12:20 <DIR>      ..
10/04/2020 12:20          301 .classpath
10/04/2020 12:20          385 .project
10/04/2020 12:20 <DIR>      .settings
10/04/2020 12:21 <DIR>      bin
10/04/2020 12:21 <DIR>      src
                2 archivos      686 bytes
                5 dirs 402.867.830.784 bytes libres

C:\Users\ymoga\Documents\GIT_repo\Proyecto1>git init
Initialized empty Git repository in C:/Users/ymoga/Documents/GIT_repo/Proyecto1/.git/

C:\Users\ymoga\Documents\GIT_repo\Proyecto1>
```

# Ejercicio práctico con Git + GitHub

- Si comprobamos el estado del repositorio: `git status`. Nos indicará que tenemos ficheros y directorios de los que no hemos realizado ningún commit.



```
Símbolo del sistema
Directorio de C:\Users\ymoga\Documents\GIT_repo\Proyecto1

No se encuentra el archivo

C:\Users\ymoga\Documents\GIT_repo\Proyecto1>dir /a:h
El volumen de la unidad C es Windows-SSD
El número de serie del volumen es: F271-2AFF

Directorio de C:\Users\ymoga\Documents\GIT_repo\Proyecto1

10/04/2020  12:34    <DIR>          .git
               0 archivos             0 bytes
               1 dirs  402.856.005.632 bytes libres

C:\Users\ymoga\Documents\GIT_repo\Proyecto1>git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .classpath
        .project
        .settings/
        bin/
        src/

nothing added to commit but untracked files present (use "git add" to track)
C:\Users\ymoga\Documents\GIT_repo\Proyecto1>
```

# Ejercicio práctico con Git + GitHub

- Realizamos el primer commit:
  1. Añadimos el proyecto a la zona de index → git add .
  2. Podemos ver lo que hemos realizado hasta el momento → git status
  3. Hacemos el commit → git commit -m "mensaje"

```
C:\Users\ymoga\Documents\GIT_repo\Proyecto1>git add .

C:\Users\ymoga\Documents\GIT_repo\Proyecto1>git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   .classpath
    new file:   .project
    new file:   .settings/org.eclipse.jdt.core.prefs
    new file:   bin/ClaseHola.class
    new file:   src/ClaseHola.java

C:\Users\ymoga\Documents\GIT_repo\Proyecto1>git commit -m "Primer commit - creación del proyecto"
[master (root-commit) 1f3df39] Primer commit - creación del proyecto
 5 files changed, 43 insertions(+)
 create mode 100644 .classpath
 create mode 100644 .project
 create mode 100644 .settings/org.eclipse.jdt.core.prefs
 create mode 100644 bin/ClaseHola.class
 create mode 100644 src/ClaseHola.java
```

# Ejercicio práctico con Git + GitHub

- Realizamos un segundo commit modificando la clase creada para que nos saque por consola el mensaje “Hola, este es mi primer repositorio git”. Si realizamos “git status” veremos que nos indica que hay cambios que no hemos actualizado en el repositorio. Después realizamos los mismos pasos para actualizar el repositorio local (git add + git commit).

```
C:\Users\ymoga\Documents\GIT_repo\Proyecto1>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:   bin/ClaseHola.class
        modified:   src/ClaseHola.java

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

C:\Users\ymoga\Documents\GIT_repo\Proyecto1>git add .

C:\Users\ymoga\Documents\GIT_repo\Proyecto1>git commit -m "Modifico la clase ClaseHola"
[master 59286fe] Modifico la clase ClaseHola
 2 files changed, 1 insertion(+), 1 deletion(-)
 rewrite bin/ClaseHola.class (61%)

C:\Users\ymoga\Documents\GIT_repo\Proyecto1>git log --oneline
59286fe (HEAD -> master) Modifico la clase ClaseHola
1f3df39 Primer commit - creaci<C3><B3>n del proyecto

C:\Users\ymoga\Documents\GIT_repo\Proyecto1>
```



# Ejercicio práctico con Git + GitHub

## 2. Subir cambios a un repositorio remoto (GitHub en nuestro caso)

- Nos creamos una cuenta en GitHub (<https://github.com/>).
- Nos creamos un repositorio:

github.com/new

dos - ymoreno... EFA Importados

Pull requests Issues Marketplace Explore

### Create a new repository

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

Owner: ymorenog / Repository name: Proyecto1

Great repository names are short and memorable. Need inspiration? How about **improved-eureka?**

Description (optional)

☒ **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

# Ejercicio práctico con Git + GitHub

## 2. Subir cambios a un repositorio remoto (GitHub en nuestro caso)

- Conectamos nuestro repositorio local con GitHub (git remote add "aliasrepo" url\_repo\_remoto).
- Si todo ha ido bien, comprobamos listando los repositorios remotos que tengo conectados: git remote -v

```
C:\Users\ymoga\Documents\GIT_repo\Proyecto1>git remote add remotoPrueba1 https://github.com/ymorenog/Proyecto1.git

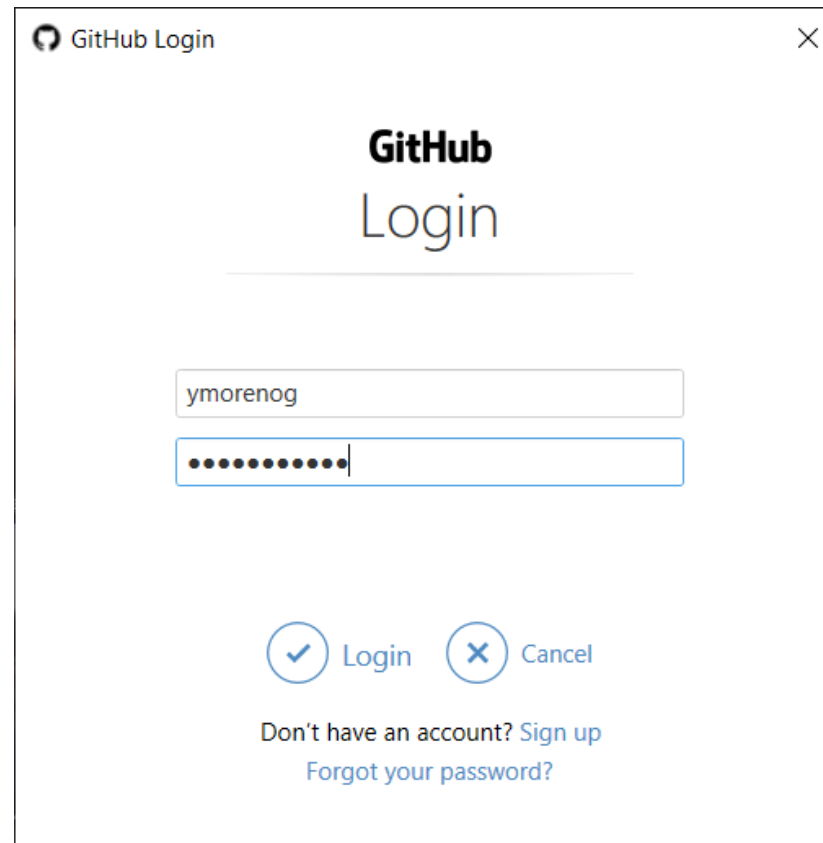
C:\Users\ymoga\Documents\GIT_repo\Proyecto1>git remote -v
remotoPrueba1  https://github.com/ymorenog/Proyecto1.git (fetch)
remotoPrueba1  https://github.com/ymorenog/Proyecto1.git (push)

C:\Users\ymoga\Documents\GIT_repo\Proyecto1>
```

# Ejercicio práctico con Git + GitHub

## 2. Subir cambios a un repositorio remoto (GitHub en nuestro caso)

- Subimos los cambios locales al repositorio remoto conectado (git push)
- Nos pedirá usuario y contraseña del GitHub



The image shows a GitHub Login dialog box. At the top left is the GitHub logo and the text "GitHub Login". At the top right is a close button (X). In the center, the words "GitHub" and "Login" are displayed. Below this is a horizontal line. There are two input fields: the first contains the username "ymorenog" and the second contains a password represented by ten dots. At the bottom, there are two buttons: "Login" with a checkmark icon and "Cancel" with an X icon. Below these buttons, there are two links: "Don't have an account? Sign up" and "Forgot your password?".

# Ejercicio práctico con Git + GitHub

## 2. Subir cambios a un repositorio remoto (GitHub en nuestro caso)

```
C:\Users\ymoga\Documents\GIT_repo\Proyecto1>git push remotoPrueba1 master
Enumerating objects: 16, done.
Counting objects: 100% (16/16), done.
Delta compression using up to 8 threads
Compressing objects: 100% (12/12), done.
Writing objects: 100% (16/16), 2.02 KiB | 1.01 MiB/s, done.
Total 16 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), done.
To https://github.com/ymorenog/Proyecto1.git
 * [new branch]      master -> master

C:\Users\ymoga\Documents\GIT_repo\Proyecto1>_
```

# Ejercicio práctico con Git + GitHub

- Comprobamos si se ha subido el proyecto al repositorio remoto actualizando la página:

The screenshot shows the GitHub interface for a repository named 'Proyecto1' by user 'ymorenog'. At the top, there are buttons for 'Unwatch' (1), 'Star' (0), and 'Fork' (0). Below this is a navigation bar with links for 'Code', 'Issues' (0), 'Pull requests' (0), 'Actions', 'Projects' (0), 'Wiki', 'Security', 'Insights', and 'Settings'. The main content area shows the repository name 'Primer proyecto repo git' with an 'Edit' button. Below this, a summary bar displays '2 commits', '1 branch', '0 packages', '0 releases', and '1 contributor'. A secondary bar contains a 'Branch: master' dropdown, a 'New pull request' button, and buttons for 'Create new file', 'Upload files', 'Find file', and a green 'Clone or download' button. The commit history table shows the following entries:

Commit icon	Commit message	Time ago
ymorenog	Modifico la clase ClaseHola	Latest commit 59286fe 25 minutes ago
Folder icon	.settings	Primer commit - creación del proyecto 38 minutes ago
Folder icon	bin	Modifico la clase ClaseHola 25 minutes ago
Folder icon	src	Modifico la clase ClaseHola 25 minutes ago
File icon	.classpath	Primer commit - creación del proyecto 38 minutes ago
File icon	.project	Primer commit - creación del proyecto 38 minutes ago

At the bottom, there is a light blue box with the text 'Help people interested in this repository understand your project by adding a README.' and a green 'Add a README' button.

# Ejercicio práctico con Git + GitHub

## 3. Trabajar a partir de un repositorio creado en GitHub:

- Creamos un nuevo repositorio en GitHub o utilizamos uno ya existente.
- Clonamos el repositorio remoto en nuestro repositorio local:
  1. Nos posicionamos en el directorio local donde queramos clonar el repositorio.
  2. Escribimos el comando: `git clone url_repositorio`.

```
C:\> Símbolo del sistema
1f3df39 Primer commit - creaci<C3><B3>n del proyecto

C:\Users\ymoga\Documents\GIT_repo\Proyecto1>cd..

C:\Users\ymoga\Documents\GIT_repo>git clone https://github.com/ymorenog/Proyecto2.git
Cloning into 'Proyecto2'...
warning: You appear to have cloned an empty repository.

C:\Users\ymoga\Documents\GIT_repo>
```

# Ejercicio práctico con Git + GitHub

## 3. Trabajar a partir de un repositorio creado en GitHub:

- Podemos comprobar que el directorio clonado está conectado con git remote -v.
- Por defecto a los repositorios que clonamos le asigna el alias “origin”, lo podemos renombrar con git remote rename origin remotoPrueba2.

```
Directorio de C:\Users\ymoga\Documents\GIT_repo\Proyecto2

10/04/2020  18:12    <DIR>          .git
                0 archivos            0 bytes
                1 dirs  403.076.378.624 bytes libres

C:\Users\ymoga\Documents\GIT_repo\Proyecto2>git remote -v
origin  https://github.com/ymorenog/Proyecto2.git (fetch)
origin  https://github.com/ymorenog/Proyecto2.git (push)

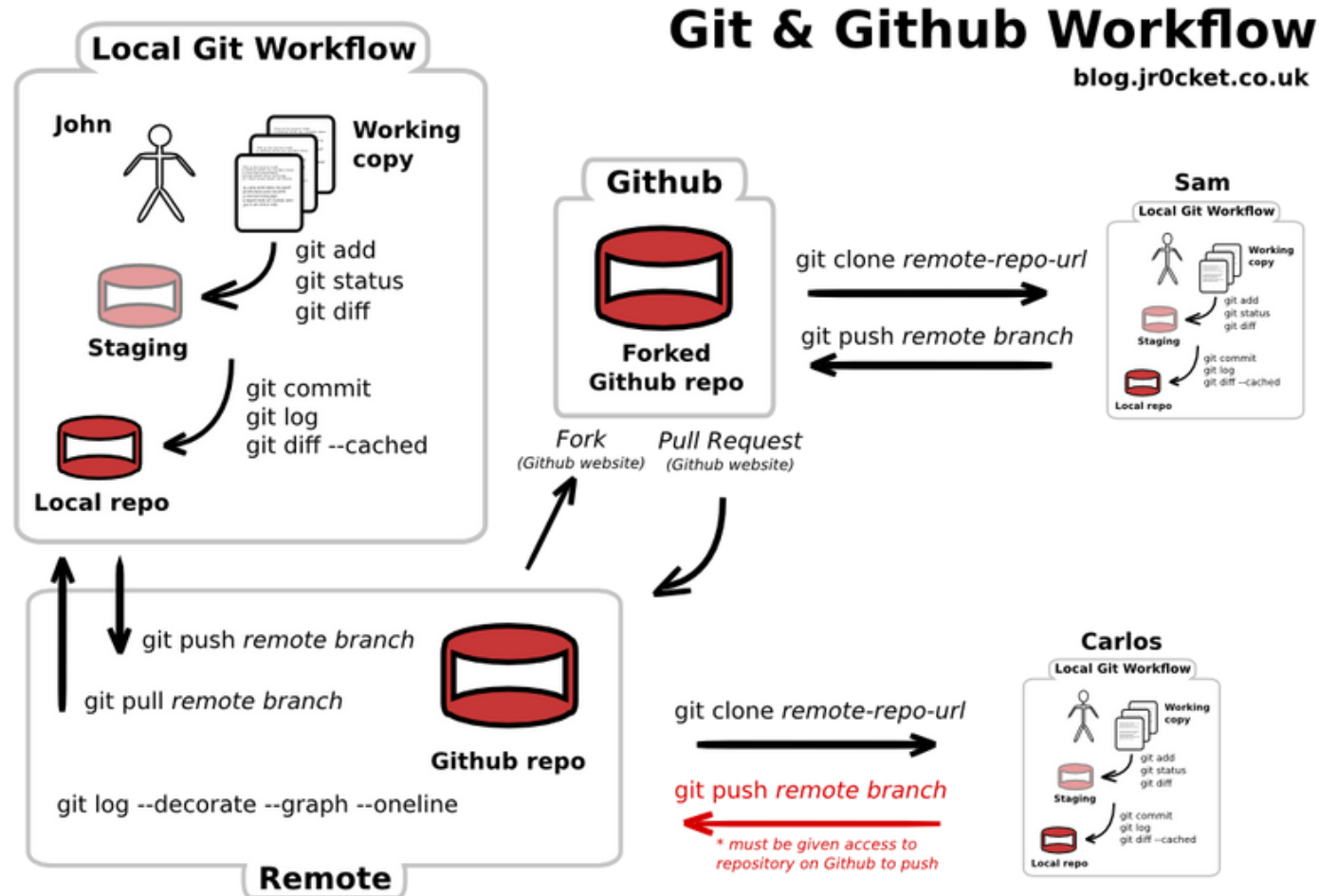
C:\Users\ymoga\Documents\GIT_repo\Proyecto2>
```

```
C:\Users\ymoga\Documents\GIT_repo\Proyecto2>git remote rename origin remotoPrueba2

C:\Users\ymoga\Documents\GIT_repo\Proyecto2>git remote -v
remotoPrueba2  https://github.com/ymorenog/Proyecto2.git (fetch)
remotoPrueba2  https://github.com/ymorenog/Proyecto2.git (push)

C:\Users\ymoga\Documents\GIT_repo\Proyecto2>
```

# Ejercicio práctico con Git + GitHub

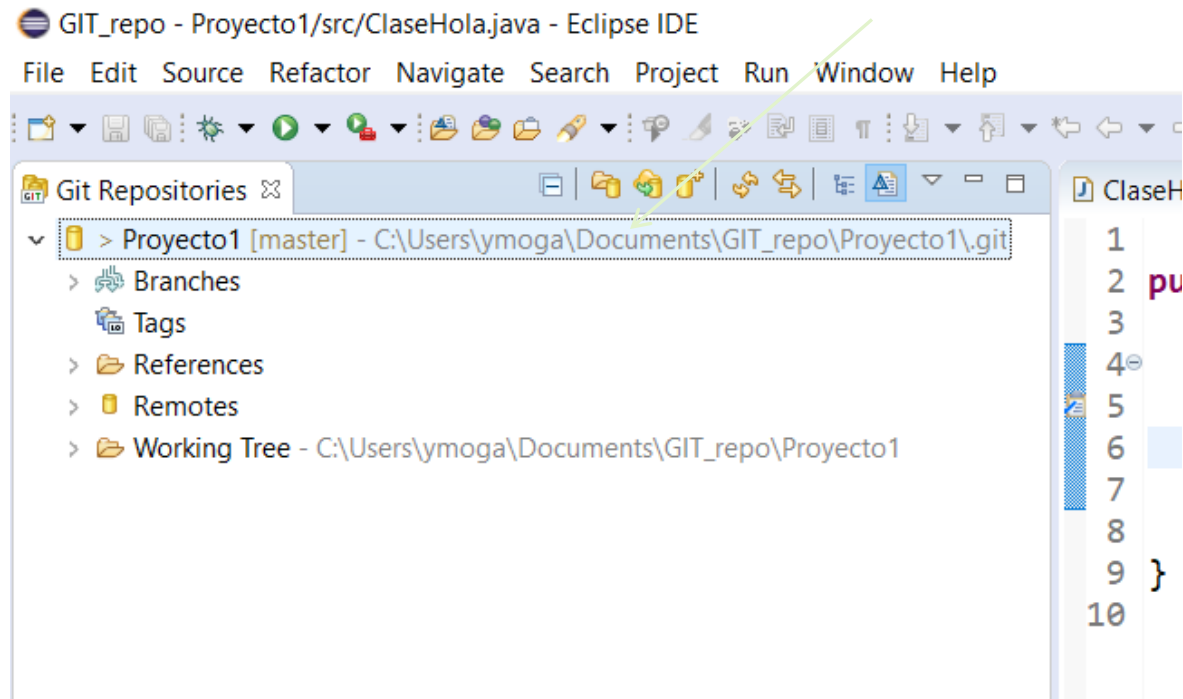


<http://jr0cket.co.uk/2013/08/getting-to-grips-with-git-understanding-the-simple-workflow.html.html>



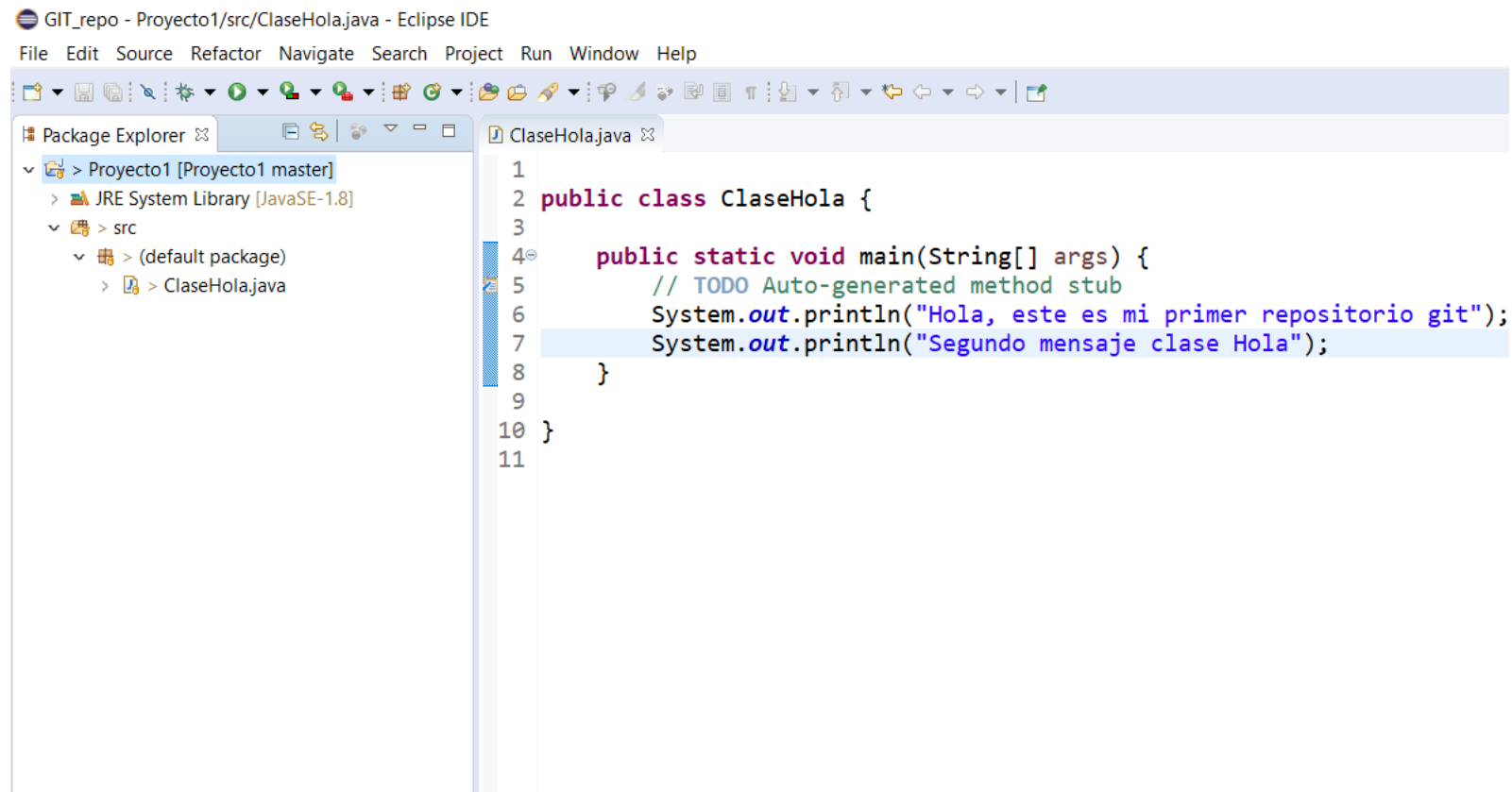
# Plugin egit de eclipse para Git

- Menú Help→Eclipse Marketplace..
- Buscamos e instalamos el plugin egit.
- Abrimos la perspectiva Git en eclipse.
- Añadimos el repositorio local que hemos creado antes: Proyecto1 en el primer botón de la barra de herramientas de Git.



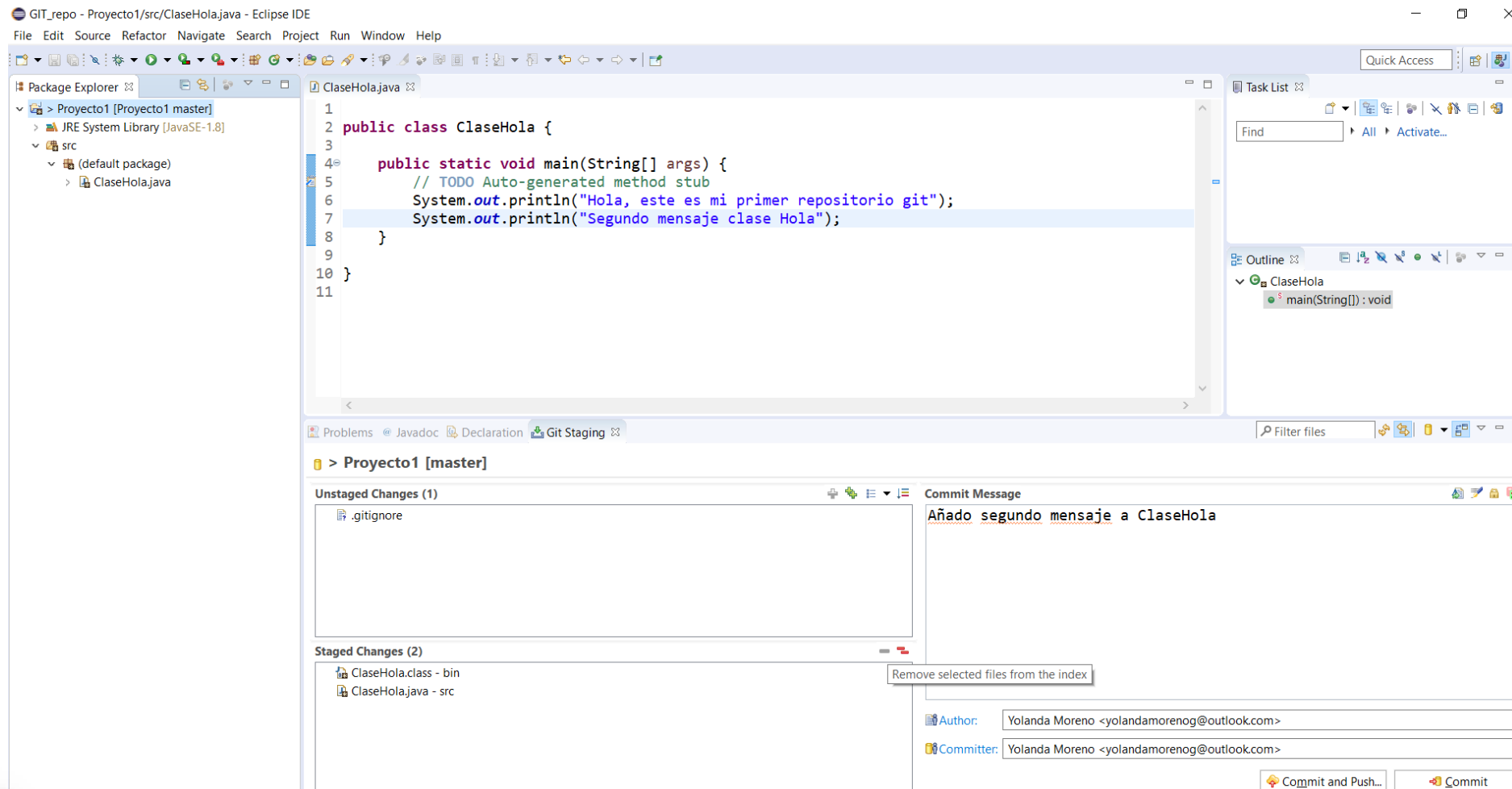
# Plugin egit de eclipse para Git

- Modificamos la clase añadiendo un nuevo mensaje por ejemplo.
- Al modificarla nos aparece en la vista java un símbolo > delante de las partes del proyecto que se han modificado.



# Plugin egit de eclipse para Git

- Hacemos commit y push de los cambios realizados: botón derecho → team → commit



# Plugin egit de eclipse para Git

- Podemos hacer los dos pasos en uno, la acción de añadir al repositorio local (add) y subirlo al repositorio remoto (push).
- Una vez realizado vemos que en la vista java han desaparecido los símbolos >.
- Para descargar la última versión del proyecto que hay en el repositorio remoto utilizaremos git pull.
- Puede ocurrir que haya conflictos entre las modificaciones de varios usuarios, en ese caso al realizar el pull eclipse nos lo indicará. En este vídeo hay un ejemplo práctico de resolución de conflictos:
  - <https://www.youtube.com/watch?v=O2tJEALJ4RU>

# Revertir cambios en Git

- En el proyecto que estábamos trabajando, incluimos una nueva línea en la clase ClaseHola. Si nos vamos a la consola y comprobamos nuestro repositorio:

```
C:\Users\ymoga\Documents\GIT_repo\Proyecto1>git status
On branch master
Your branch is up to date with 'remotoPrueba1/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:   bin/ClaseHola.class
        modified:   src/ClaseHola.java

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

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

C:\Users\ymoga\Documents\GIT_repo\Proyecto1>git diff
diff --git a/bin/ClaseHola.class b/bin/ClaseHola.class
index 3ce2681..ae1c953 100644
Binary files a/bin/ClaseHola.class and b/bin/ClaseHola.class differ
diff --git a/src/ClaseHola.java b/src/ClaseHola.java
index e8cef85..1d20b09 100644
--- a/src/ClaseHola.java
+++ b/src/ClaseHola.java
@@ -5,6 +5,7 @@ public class ClaseHola {
    // TODO Auto-generated method stub
    System.out.println("Hola, este es mi primer repositorio git");
    System.out.println("Segundo mensaje clase Hola");
+   System.out.println("Tercer mensaje clase Hola");
+ }
```

# Revertir cambios en Git

- Podemos volver a la versión anterior de nuestra clase con el comando `git checkout`, ten en cuenta que aún no hemos añadido nuestros cambios a la zona de index con `add`:

```
C:\Users\ymoga\Documents\GIT_repo\Proyecto1>git status
On branch master
Your branch is up to date with 'remotoPrueba1/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:   bin/ClaseHola.class
        modified:   src/ClaseHola.java

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

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

C:\Users\ymoga\Documents\GIT_repo\Proyecto1>git checkout -- src/ClaseHola.java

C:\Users\ymoga\Documents\GIT_repo\Proyecto1>git status
On branch master
```

# Revertir cambios en Git

- Si volvemos a modificar nuestra clase, hacemos git add y queremos revertir cambios, en este caso como ya hemos enviado las modificaciones al area index no podemos utilizar checkout --, podemos utilizar git reset (HEAD le indica que queremos volver al último commit realizado de la rama en la que estamos)

```
C:\Users\ymoga\Documents\GIT_repo\Proyecto1>git reset HEAD
Unstaged changes after reset:
M      bin/ClaseHola.class
M      src/ClaseHola.java

C:\Users\ymoga\Documents\GIT_repo\Proyecto1>git status
On branch master
Your branch is up to date with 'remotoPrueba1/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:   bin/ClaseHola.class
        modified:   src/ClaseHola.java

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

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

C:\Users\ymoga\Documents\GIT_repo\Proyecto1>
```

# Revertir cambios en Git

- Una vez aquí ya sí que podemos realizar checkout para deshacer cambios del mismo modo que hicimos antes.
- Si queremos revertir cambios cuando ya hemos hecho un commit podemos utilizar git reset indicando el índice del commit al que queremos regresar:

```
C:\Users\ymoga\Documents\GIT_repo\Proyecto1>git log --oneline
0a77ede (HEAD -> master) A<C3><B1>ado tercer mensaje a clase Hola
b44f3f7 (remotoPrueba1/master) A<C3><B1>ado segundo mensaje a ClaseHola
59286fe Modifico la clase ClaseHola
1f3df39 Primer commit - creaci<C3><B3>n del proyecto

C:\Users\ymoga\Documents\GIT_repo\Proyecto1>git reset b44f3f7
Unstaged changes after reset:
M      bin/ClaseHola.class
M      src/ClaseHola.java

C:\Users\ymoga\Documents\GIT_repo\Proyecto1>git log --oneline
b44f3f7 (HEAD -> master, remotoPrueba1/master) A<C3><B1>ado segundo mensaje a ClaseHola
59286fe Modifico la clase ClaseHola
1f3df39 Primer commit - creaci<C3><B3>n del proyecto

C:\Users\ymoga\Documents\GIT_repo\Proyecto1>_
```



# Revertir cambios en Git

- Si queremos eliminar tanto el commit como los cambios en local deberemos utilizar `git reset --hard` y el índice del commit al que queremos regresar.

```
C:\Users\ymoga\Documents\GIT_repo\Proyecto1>git log --oneline
e47b027 (HEAD -> master) Incluyo tercer mensaje en clase Hola
b44f3f7 (remotoPrueba1/master) A<C3><B1>ado segundo mensaje a ClaseHola
59286fe Modifico la clase ClaseHola
1f3df39 Primer commit - creaci<C3><B3>n del proyecto

C:\Users\ymoga\Documents\GIT_repo\Proyecto1>git reset --hard b44f3f7
HEAD is now at b44f3f7 Añado segundo mensaje a ClaseHola

C:\Users\ymoga\Documents\GIT_repo\Proyecto1>git log --oneline
b44f3f7 (HEAD -> master, remotoPrueba1/master) A<C3><B1>ado segundo mensaje a ClaseHola
59286fe Modifico la clase ClaseHola
1f3df39 Primer commit - creaci<C3><B3>n del proyecto

C:\Users\ymoga\Documents\GIT_repo\Proyecto1>_
```

# Revertir cambios en Git

- Comando revert: más aconsejado que reset a la hora de revertir cambios ya que con reset modificamos el historial que teníamos y con revert añadimos nuevos cambios al historial.

```
C:\Users\ymoga\Documents\GIT_repo\Proyecto1>git log --oneline --decorate
8af0d9c (HEAD -> master) Tercer mensaje clase Hola
b44f3f7 (remotoPrueba1/master) A<C3><B1>ado segundo mensaje a ClaseHola
59286fe Modifico la clase ClaseHola
1f3df39 Primer commit - creaci<C3><B3>n del proyecto
```

```
C:\Users\ymoga\Documents\GIT_repo\Proyecto1>git revert HEAD
Removing .gitignore
[master cd934b8] Revert "Tercer mensaje clase Hola"
 3 files changed, 2 deletions(-)
 delete mode 100644 .gitignore
```

```
C:\Users\ymoga\Documents\GIT_repo\Proyecto1>git log --oneline
cd934b8 (HEAD -> master) Revert "Tercer mensaje clase Hola"
8af0d9c Tercer mensaje clase Hola
b44f3f7 (remotoPrueba1/master) A<C3><B1>ado segundo mensaje a ClaseHola
59286fe Modifico la clase ClaseHola
1f3df39 Primer commit - creaci<C3><B3>n del proyecto
```

```
C:\Users\ymoga\Documents\GIT_repo\Proyecto1>_
```

<https://www.youtube.com/watch?v=jSJ8xhKtfP4&list=PLTd5ehIj0goMCnj6V5NdzSIHBgrIXckGU&index=1> (Serie de videos sobre Git)

# Revertir cambios en Git

- Para hacer las acciones anteriores en eclipse nos vamos a la vista de sincronización, pestaña History:

