



Git

Git

Git es un sistema de control de versiones con arquitectura distribuida. Actualmente el más utilizado. Anteriormente se utilizaban CVS o Subversion (SVN).

Descargar Git desde <https://gitforwindows.org/>

Una vez instalado se puede utilizar de la consola cmd o mejor es utilizar git-bash.

Ejecutamos git bash y se nos abre una consola tipo Linux.

Nos posicionamos en una carpeta que queramos que sea nuestro repositorio y ejecutamos

git init

```
MINGW64:/c/Users/plope/repositorio_git

plope@ASUS-PABLO MINGW64 ~
$ mkdir repositorio_git

plope@ASUS-PABLO MINGW64 ~
$ cd repositorio_git/

plope@ASUS-PABLO MINGW64 ~/repositorio_git
$ git init
Initialized empty Git repository in c:/Users/plope/repositorio_git/.git/

plope@ASUS-PABLO MINGW64 ~/repositorio_git (master)
$ |
```

Acabamos de crear un repositorio y una rama (branch) principal llamada “master” pero está aún vacía. Todo lo que pongamos en ese directorio será gestionado por git.

Podemos verificar el estado del repositorio con:

git status

```
MINGW64:/c/Users/plope/repositorio_git

plope@ASUS-PABLO MINGW64 ~/repositorio_git (master)
$ git status
On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to track)

plope@ASUS-PABLO MINGW64 ~/repositorio_git (master)
$ |
```

Y nos indica que no hay nada pendiente de guardar (commit)

Después de realizar cambios en el proyecto

Incorporamos ficheros y/o hacemos cambios a nuestro proyecto. Tendremos los cambios en lo que se denomina "Working Area" pero no actualizados en el repositorio.

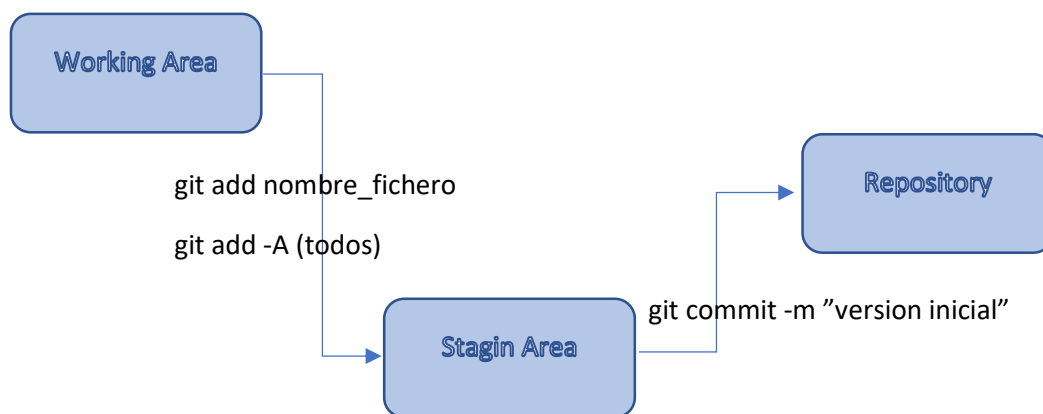
Para que queden actualizados, tendremos primero que indicar cuáles ficheros queremos actualizar o incorporar al repositorio. Para ello pasamos los ficheros elegidos a la "Stagin Area". Finalmente haremos el commit para pasar los cambios de la "Stagin Area" al "Repository".

git add nombre_fichero (sube un fichero)

git add *.html (varios ficheros usando comodines)

git add -A todos los ficheros

git commit -m "version inicial"



```
MINGW64:/c/Users/plope/repositorio_git

pllope@ASUS-PABLO MINGW64 ~/repositorio_git (master)
$ git add -A

pllope@ASUS-PABLO MINGW64 ~/repositorio_git (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

        new file:   css/a.css
        new file:   index.html

pllope@ASUS-PABLO MINGW64 ~/repositorio_git (master)
$ git commit -m "version inicial"
[master (root-commit) f4e185f] version inicial
 2 files changed, 82 insertions(+)
 create mode 100644 css/a.css
 create mode 100644 index.html

pllope@ASUS-PABLO MINGW64 ~/repositorio_git (master)
$ git status
On branch master
nothing to commit, working tree clean

pllope@ASUS-PABLO MINGW64 ~/repositorio_git (master)
$ |
```

git log

Muestra el historial de commits

```
MINGW64:/c/Users/plope/repositorio_git

plope@ASUS-PABLO MINGW64 ~/repositorio_git (master)
$ git log
commit 6eb868b42f962bf4fedd473bfa5dec43ee8218ea (HEAD -> master)
Author: Pablo Lopez <pablo.lopez@kolbit.es>
Date: Tue Sep 21 19:37:14 2021 +0200

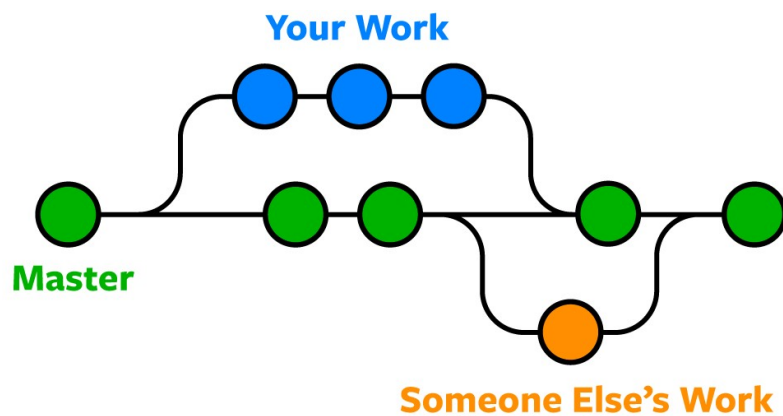
    incorporamos otra

commit f4e185f51e5d798250219edf630422325df071c5
Author: Pablo Lopez <pablo.lopez@kolbit.es>
Date: Sun Sep 19 21:46:57 2021 +0200

    version inicial

plope@ASUS-PABLO MINGW64 ~/repositorio_git (master)
$
```

Ramas (branches)





Git con Eclipse

Git

Git es un sistema de control de versiones con arquitectura distribuida. Actualmente el más utilizado. Anteriormente se utilizaban CVS o Subversion (SVN).

Configurar usuario y mail

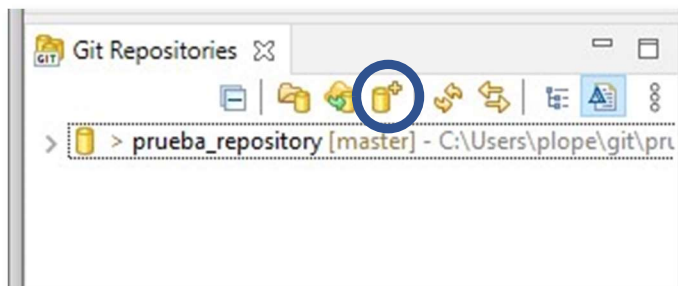
Window > Preferences > Team o Version Control (Team) > Git > Configuration

- user.email pepito.perez@kolbit.es
- user.name Pepito Perez
- pull.rebase true

Crear un nuevo repositorio local

Window > Show View > Other > Git > Git Repositories

Crear nuevo repositorio local



Crearlo fuera del workspace. Por ejemplo bajo el usuario.

Crear el fichero .gitignore

Esto hay que hacerlo fuera de eclipse. Crear un fichero de texto en la carpeta raíz del repositorio que se llame “.gitignore” (cuidado con la extensión .txt que agrega Windows)

En este fichero indicaremos los ficheros que queremos que ignore Git, por ejemplo todos los .class

En el fichero pondremos:

```
C:\Users\plope\git\prueba_repository>type .gitignorefile
bin
*.class
C:\Users\plope\git\prueba_repository>
```

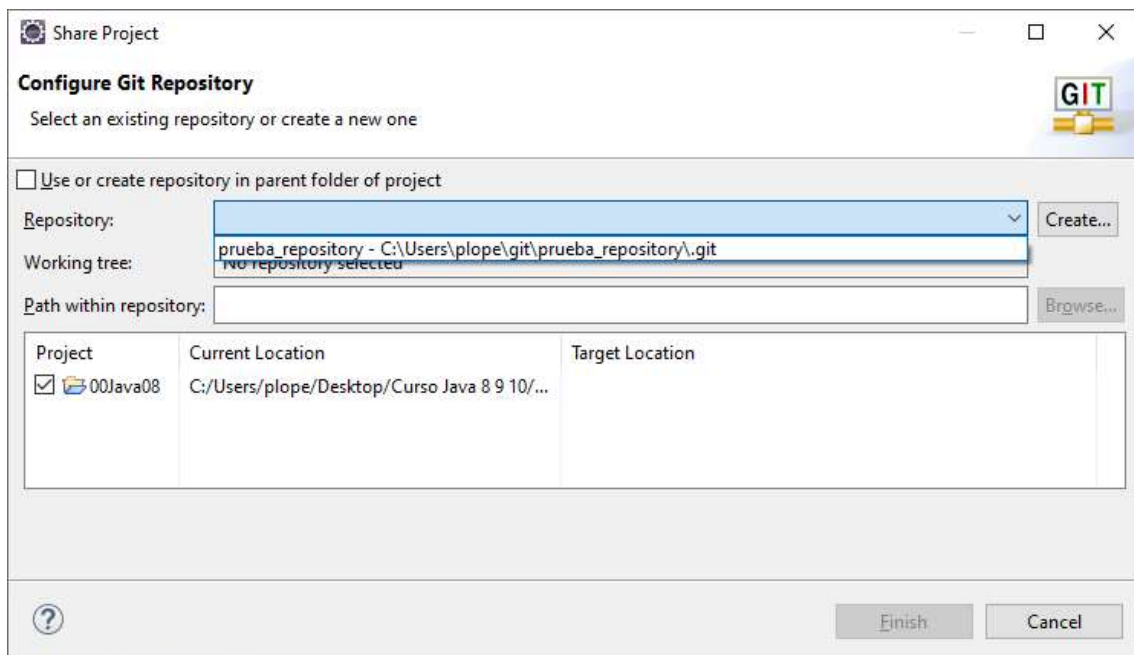
Crear un proyecto java

Creamos un proyecto java normal. Agregamos un clase GitTest

```
public class GitTest {
    public static void main(String[] args) {
        System.out.println("Git configurado!!!");
    }
}
```

Utilizar el control de versiones en este proyecto

Botón derecho en el proyecto, Team > Share Project y seleccionar el repositorio

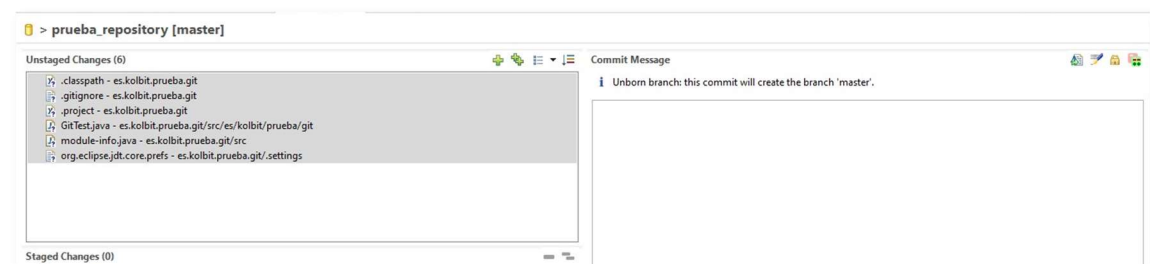


Y Finish

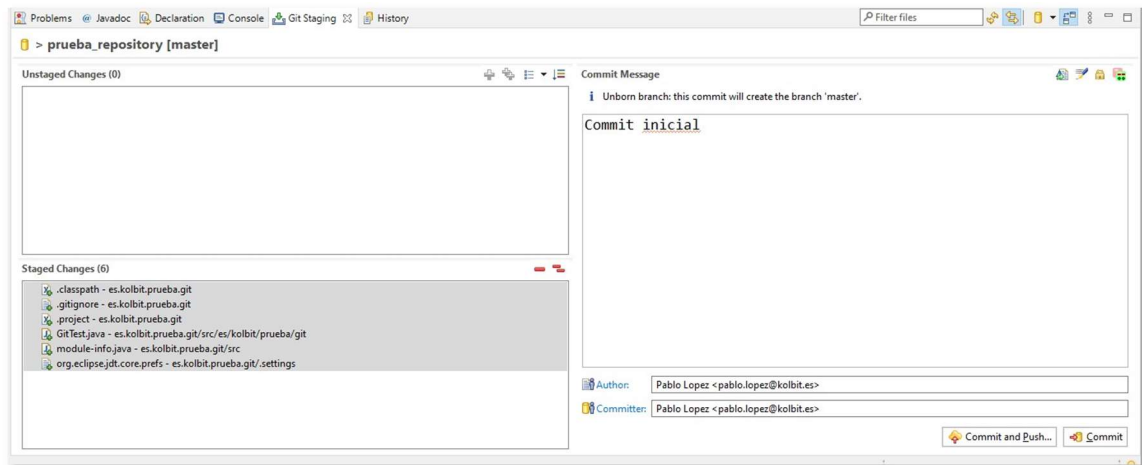
El proyecto ya está controlado por Git.

Usar "Git Staging view" para realizar los commits

Window > Show View > Other > Git > Git Staging



Pasar los ficheros de «Unstaged Changes» a «Staged Changes», agregar un mensaje y hacer Commit

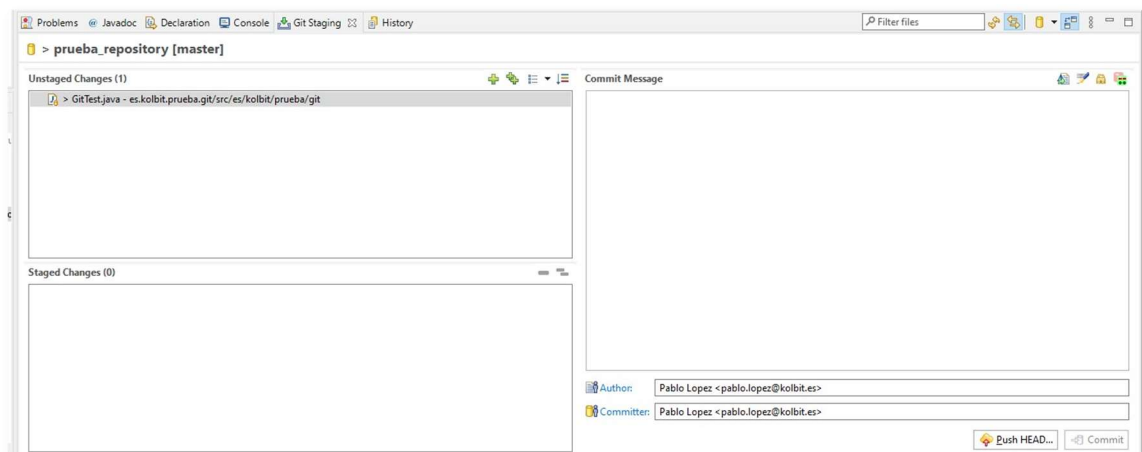


Hacer cambios en el código

Modificamos nuestro fuente

```
public class GitTest {  
    public static void main(String[] args) {  
        System.out.println("He modificado el código!!!");  
    }  
}
```

Y al grabar, automáticamente aparece nuestro fuente en “Unstaged Changes” :



Lo pasamos a “Staged Changes”, ponemos un mensaje referido a las modificaciones y hacemos Commit

Ver el histórico

Window > Show View >