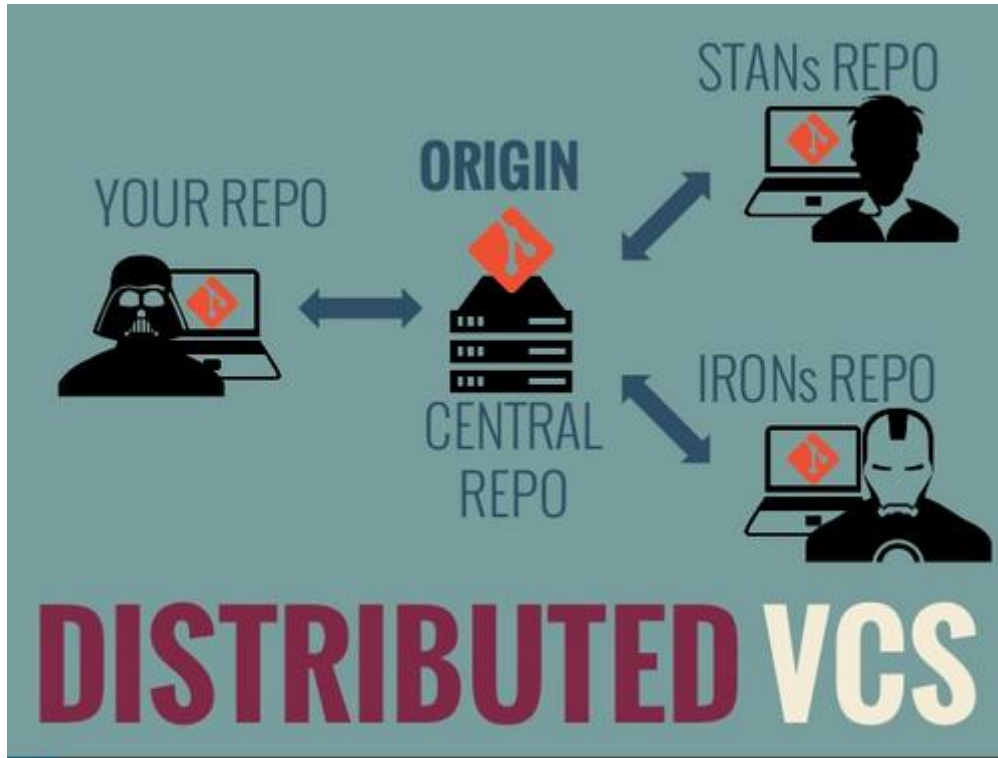


## Guió GIT

Recordeu, GIT és un gestor de versions (CVS) distribuït. Cada usuari té un repositori de codi amb tota la història.



### Treballant només amb repositori local

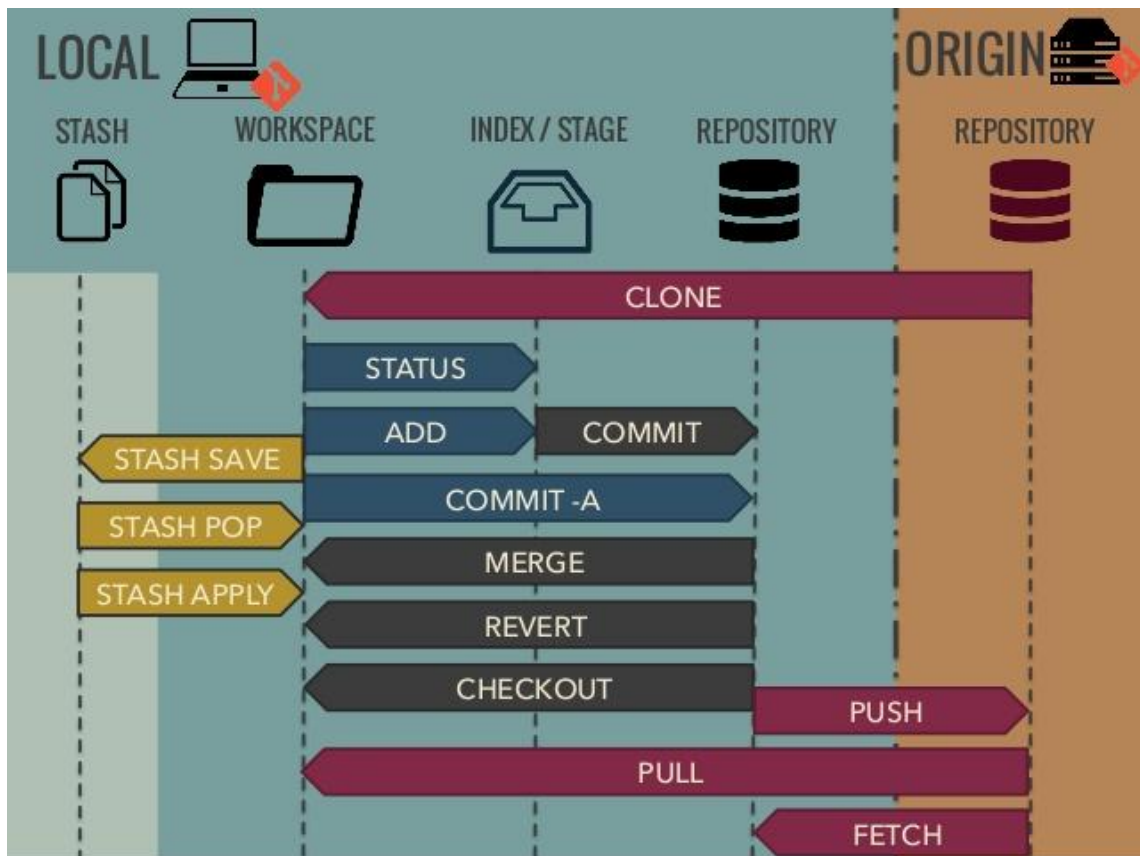
1. Introducció a la gestió de versions
2. Sistemes de control de versions locals, centralitzats i distribuïts
3. Setup ( GIT Windows)
4. Definició de credencials a GIT

```
git config --global user.name "John Doe"
git config --global user.email johndoe@example.com
```

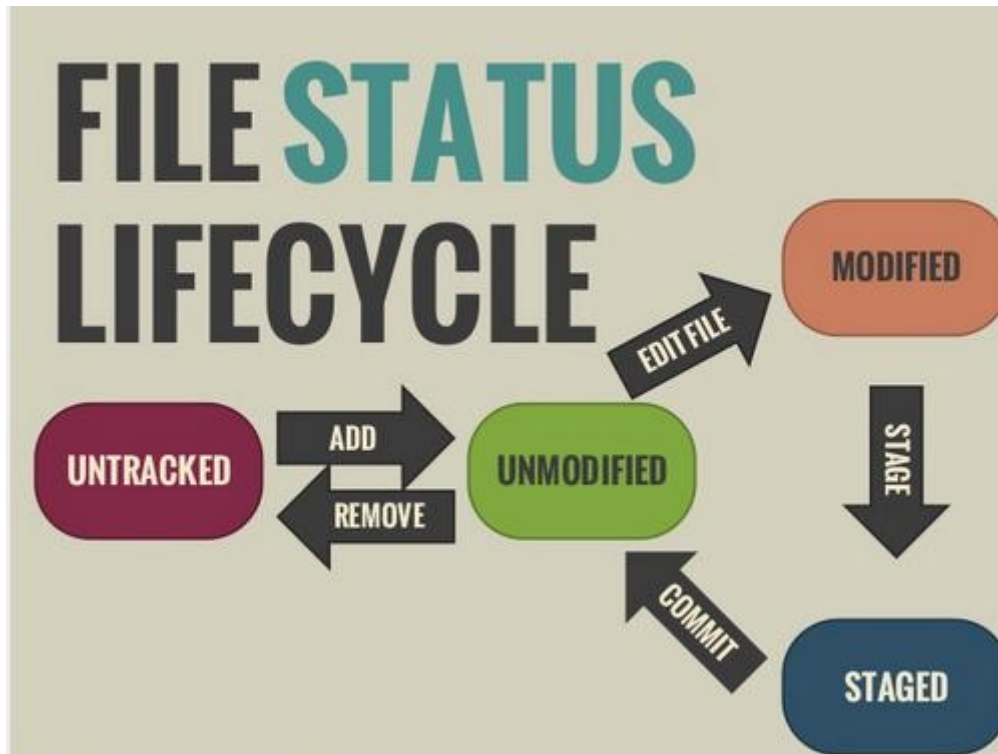
5. Inicialització del repositori local:

```
git init
```

6. Utilització de GIT local



Stash és la zona de còpies ràpides quan està la feina a mitges i no es pot fer un commit, però desitgem fer una operació amb el repositori local.	<ul style="list-style-type: none"> <li>• arxius unstaged (modificats o esborrats)</li> <li>• arxius untracked (nous)</li> </ul>	Stage area és on s'han de posar els arxius per poder acceptar els canvis	Committed accepta els canvis.	
---	---	--	-------------------------------	--



7. Add i commit

7.1. Passar a Staged els arxius nous, modificats i esborrats :

```
git add .
```

7.2. Commit :

```
git commit -m 'Missatge'
```

7.3. Add+commit ( **només** arxius modificats, no arxius nous) :

```
git commit -a -m 'Missatge'
```

8. Llistar tots els canvis respecte la última versió confirmada ( últim commit ):

```
git diff
```

9. Canvi de nom d'un arxiu

```
git mv [nom_vell] [nom_nou]
```



10. **SI L'HEM CAGAT** . Operacions per arreglar destrosses:

10.1. Desfer els canvis “staged” ( desfer git add ):

```
git reset
```

10.2. Descartar tots els canvis respecte la última versió confirmada (últim commit ):

```
git checkout -- *
```

10.3. Desfer l'últim commit ( desfà TOTS els canvis introduïts pel commit, i fa un altre commit amb la versió “restaurada” )

```
git revert HEAD
```

11. Ignorant arxius: Crear un arxiu `.gitignore` a l'arrel del projecte. Es mostra un exemple a continuació dels seus continguts. Useu una plantilla existent

<https://www.gitignore.io/>

```
# Created by https://www.gitignore.io/api/java
### Java ###
*.class
# BlueJ files
*.ctxt
# Mobile Tools for Java (J2ME)
.mtj.tmp/
# Package Files #
*.jar
*.war
*.ear
# 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
```

12. Git log, ( és el *Git Show History* des de Netbeans ):

```
git log
```

12.1. Per veure jerarquia:

```
git log --graph --all
```

### 13. Definició d'Àlies i millora gràfica del log:

Dins de C:\Program Data\Git\config, editeu

~/.gitconfig

Afegiu les línies següents:

```
[alias]
lg1 = log --graph --abbrev-commit --decorate --date=relative --
format=format:'%C(bold blue)%h%C(reset) - %C(bold green)(%ar)%C(reset)
%C(white)%s%C(reset) %C(dim white)- %an%C(reset)%C(bold
yellow)%d%C(reset)' --all
lg2 = log --graph --abbrev-commit --decorate --format=format:'%C(bold
blue)%h%C(reset) - %C(bold cyan)%aD%C(reset) %C(bold
green)(%ar)%C(reset)%C(bold yellow)%d%C(reset)%n'
%C(white)%s%C(reset) %C(dim white)- %an%C(reset)' --all
lg = !"git lg1"
```

### 14. Concepte de ID de commit ( hash )

Cada commit s'identifica únicament amb un codi Hash:

```
git log
commit 649eee4d5f85f8bcfda6637087e769a328ad98d0
author: Bernat Orellana <borellan@xtec.cat>
date: Thu Feb 2 13:16:57 2017 +0100

segon commit

commit 2c5ab53c2ef97bdba02d53a267ba577de7f41a23
author: Bernat Orellana <borellan@xtec.cat>
date: Thu Feb 2 13:10:48 2017 +0100

1st
```

### 15. Git checkout

15.1. A un commit concret (usant HASH):

```
git checkout {inici del hash del commit}
```

15.2. A l'última versió de la branca ( master ) :

```
git checkout master
```

16. El paper de HEAD ( apunta a la versió actual )
17. Marques de versió ( TAGS! )
  - 17.1. Llistar tags: git tag
  - 17.2. Crear un tag: ( git tag {tagname} )
  - 17.3. Crear un tag anotat (per versions importants ): ( git tag -a {tagname} -m {missatge de versió} )
  - 17.4. Compartir un tag a un repo remot: git push --follow-tags
  - 17.5. Veure dades d'un tag: git show {tagname}
  - 17.6. Git checkout tag
  - 17.7. Detached HEAD: no podem fer commits si hem fet un checkout a un tag o a un commit concret !!
18. Branques
  - 18.1. Concepte de branca i organització interna de GIT
  - 18.2. Creació de branques: git branch {branchname}
  - 18.3. Moure's a una branca : git checkout {branchname}
  - 18.4. Merge de branques
    - 18.4.1. Git merge
  - 18.5. Conflictes de merge i la seva resolució.

## Treballant amb repositoris remots:

1. Llista de repositoris remots:

```
git remote
```

2. Clonar un repositori remot ( per defecte s'anomenarà origin ):

```
git clone {url} {opcional:nom del directori}
```

```
git clone https://github.com/infomila/GitTesting.git → Crea una carpeta GitTesting
```

```
git clone https://github.com/infomila/GitTesting.git GitTestingPaco → Crea una carpeta GitTestingPaco
```

Si volem canviar el nom del repositori remot podem indicar-ho amb "-o"

```
git clone -o otherorigin https://github.com/infomila/GitTesting.git
```

3. Pujar els canvis locals al repositori remot:

```
git push {repo remot}
```

```
git push
```

4. Sincronitzar el nostre repositori amb el repositori remot ( sense barrejar canvis i )

```
git fetch {repo remot}
```

git fetch origin

git fetch {origin}: descarregar els canvis del repositori remot que no teniem al local

Les branques per defecte no són compartides als repositoris remots. Si volem fer-ho , cal fer un push explícit:

git push {origin} {nom\_branca}

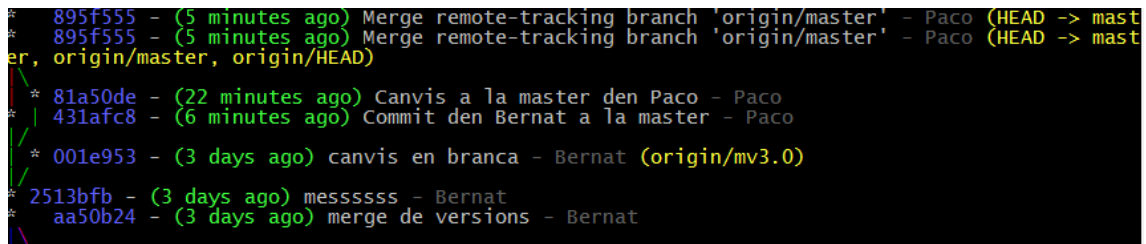
git merge origin/master

git fetch origin + git merge origin/master = git pull origin

## 5. Llista de branques remotes

git branch -r

## 6. Seguir una branca remota:



```
* 895f555 - (5 minutes ago) Merge remote-tracking branch 'origin/master' - Paco (HEAD -> mast
* 895f555 - (5 minutes ago) Merge remote-tracking branch 'origin/master' - Paco (HEAD -> mast
er, origin/master, origin/HEAD)
* 81a50de - (22 minutes ago) Canvis a la master den Paco - Paco
| 431afc8 - (6 minutes ago) Commit den Bernat a la master - Paco
|
* 001e953 - (3 days ago) canvis en branca - Bernat (origin/mv3.0)
|
* 2513bfb - (3 days ago) messssss - Bernat
* aa50b24 - (3 days ago) merge de versions - Bernat
```

git checkout --track -b refactored origin/refactored

```
$ git checkout --track -b mv3.0 origin/mv3.0
Branch mv3.0 set up to track remote branch mv3.0 from origin.
Switched to a new branch 'mv3.0'
```