

Git avanzado



Temario

- Propósito y alcance
- Arquitectura
- Componentes fundamentales
- Operaciones CRUD
- Operaciones de organización de código
- Gestión de ramas y etiquetas
- Resolución de conflictos
- Integración con Maven
- Configuración de GIT
- Introducción a GIT en el lado servidor
- Programas cliente para operar con GIT

Conceptos iniciales

THIS IS GIT. IT TRACKS COLLABORATIVE WORK
ON PROJECTS THROUGH A BEAUTIFUL
DISTRIBUTED GRAPH THEORY TREE MODEL.

COOL. HOW DO WE USE IT?

NO IDEA. JUST MEMORIZE THESE SHELL
COMMANDS AND TYPE THEM TO SYNC UP.
IF YOU GET ERRORS, SAVE YOUR WORK
ELSEWHERE, DELETE THE PROJECT,
AND DOWNLOAD A FRESH COPY.



Conceptos iniciales

Conceptos iniciales

- Qué es el control de versiones

Conceptos iniciales

- Qué es el control de versiones
- VCS centralizados / distribuidos

Conceptos iniciales

- Qué es el control de versiones
- VCS centralizados / distribuidos
- CLI vs GUI

Conceptos iniciales

- Qué es el control de versiones
- VCS centralizados / distribuidos
- CLI vs GUI
- Branching

Instalación

Instalación

Instalación

- Notepad++

Instalación

- Notepad++
- [Git](#)
- Marcar o asegurarse de que está marcado:
 - Git Bash Here
 - Use Notepad++ as Git's default editor
 - Use Git from the Windows Command Prompt
 - Checkout Windows-style, commit Unix-style etc.
 - Use MinTTY

Instalación

- Notepad++
- [Git](#)
- Marcar o asegurarse de que está marcado:
 - Git Bash Here
 - Use Notepad++ as Git's default editor
 - Use Git from the Windows Command Prompt
 - Checkout Windows-style, commit Unix-style etc.
 - Use MinTTY
- Crear una cuenta en [Bitbucket](#) (con un email real)

Configuración inicial

Configuración inicial

- Configuración proxy:
 - `git config --global http.proxy http://username:password@host:port`
 - `git config --global https.proxy http://username:password@host:port`

Configuración inicial

- Configuración proxy:
 - `git config --global http.proxy http://username:password@host:port`
 - `git config --global https.proxy http://username:password@host:port`
- Configuración usuario:
 - `git config --global user.name "Mario González"`
 - `git config --global user.email mario@mariogl.com`

Fundamentos de Git

Fundamentos de Git

Fundamentos de Git

- Snapshots (commits)

Fundamentos de Git

- Snapshots (commits)
- Repositorio

Fundamentos de Git

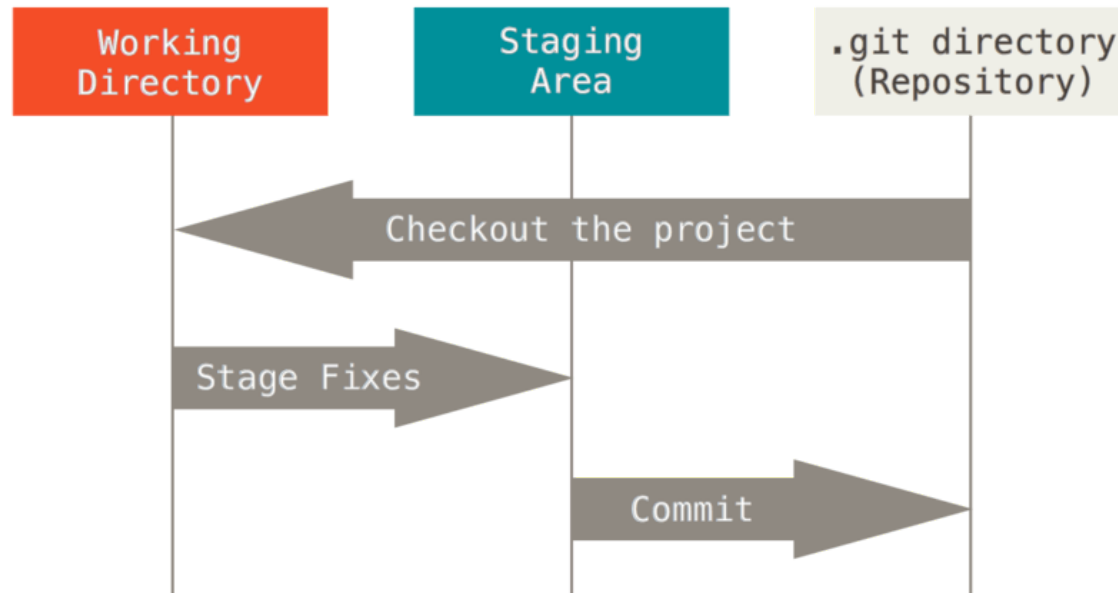
- Snapshots (commits)
- Repositorio
- Working directory

Fundamentos de Git

- Snapshots (commits)
- Repositorio
- Working directory
- Stage/index/cache

Fundamentos de Git

- Snapshots (commits)
- Repositorio
- Working directory
- Stage/index/cache



Fundamentos de Git

Fundamentos de Git

- Referencias
 - Ramas
 - HEAD
 - Tags

Fundamentos de Git

- Referencias
 - Ramas
 - HEAD
 - Tags
- .gitignore

Commits

Preparando el commit

Preparando el commit

- Añadir al índice con `git add`

Preparando el commit

- Añadir al índice con `git add`
- Añadir de manera interactiva con `git add -i`

Preparando el commit

- Añadir al índice con `git add`
- Añadir de manera interactiva con `git add -i`
- Ver el staging area con `git status`

Preparando el commit

- Añadir al índice con `git add`
- Añadir de manera interactiva con `git add -i`
- Ver el staging area con `git status`
- Quitar archivos o modificaciones del staging area con `git reset`

Preparando el commit

- Añadir al índice con `git add`
- Añadir de manera interactiva con `git add -i`
- Ver el staging area con `git status`
- Quitar archivos o modificaciones del staging area con `git reset`
- Hacer el commit con `git commit`

Preparando el commit

- Añadir al índice con `git add`
- Añadir de manera interactiva con `git add -i`
- Ver el staging area con `git status`
- Quitar archivos o modificaciones del staging area con `git reset`
- Hacer el commit con `git commit`
- Modificar el último commit con `git commit --amend`

El commit

El commit

- Hash SHA-1

El commit

- Hash SHA-1
- Autor / commiteador

El commit

- Hash SHA-1
- Autor / commiteador
- Fecha

El commit

- Hash SHA-1
- Autor / commiteador
- Fecha
- Mensaje

El commit

- Hash SHA-1
- Autor / commiteador
- Fecha
- Mensaje
- Padre/s

El log

El log

- `git log` para ver el histórico de commits

El log

- `git log` para ver el histórico de commits
- Referencias en el log

El log

- `git log` para ver el histórico de commits
- Referencias en el log
- Ver el log compacto con `git log --oneline`

El log

- `git log` para ver el histórico de commits
- Referencias en el log
- Ver el log compacto con `git log --oneline`
- Ver un commit con `git show` y `git ls-tree`

El log

- `git log` para ver el histórico de commits
- Referencias en el log
- Ver el log compacto con `git log --oneline`
- Ver un commit con `git show` y `git ls-tree`
- Cómo referenciar a un commit

El log

- `git log` para ver el histórico de commits
- Referencias en el log
- Ver el log compacto con `git log --oneline`
- Ver un commit con `git show` y `git ls-tree`
- Cómo referenciar a un commit
 - SHA-1

El log

- git log para ver el histórico de commits
- Referencias en el log
- Ver el log compacto con git log --oneline
- Ver un commit con git show y git ls-tree
- Cómo referenciar a un commit
 - SHA-1
 - HEAD^

El log

- git log para ver el histórico de commits
- Referencias en el log
- Ver el log compacto con git log --oneline
- Ver un commit con git show y git ls-tree
- Cómo referenciar a un commit
 - SHA-1
 - HEAD^
 - HEAD~

El log

- git log para ver el histórico de commits
- Referencias en el log
- Ver el log compacto con git log --oneline
- Ver un commit con git show y git ls-tree
- Cómo referenciar a un commit
 - SHA-1
 - HEAD^
 - HEAD~
 - HEAD@{n}

El log

El log

- Saltar a otro commit con git checkout

El log

- Saltar a otro commit con git checkout
- Borrar commits con git reset

El log

- Saltar a otro commit con `git checkout`
- Borrar commits con `git reset`
- Ver el histórico de HEADs con `git reflog`

El log

- Saltar a otro commit con `git checkout`
- Borrar commits con `git reset`
- Ver el histórico de HEADs con `git reflog`
- Apartar cambios provisionalmente con `git stash`

El log

- Saltar a otro commit con git checkout
- Borrar commits con git reset
- Ver el histórico de HEADs con git reflog
- Apartar cambios provisionalmente con git stash
 - git stash

El log

- Saltar a otro commit con git checkout
- Borrar commits con git reset
- Ver el histórico de HEADs con git reflog
- Apartar cambios provisionalmente con git stash
 - git stash
 - git stash list

El log

- Saltar a otro commit con `git checkout`
- Borrar commits con `git reset`
- Ver el histórico de HEADs con `git reflog`
- Apartar cambios provisionalmente con `git stash`
 - `git stash`
 - `git stash list`
 - `git stash show stash@{n} -v`

El log

- Saltar a otro commit con `git checkout`
- Borrar commits con `git reset`
- Ver el histórico de HEADs con `git reflog`
- Apartar cambios provisionalmente con `git stash`
 - `git stash`
 - `git stash list`
 - `git stash show stash@{n} -v`
 - `git stash apply [stash@{n}]`

El log

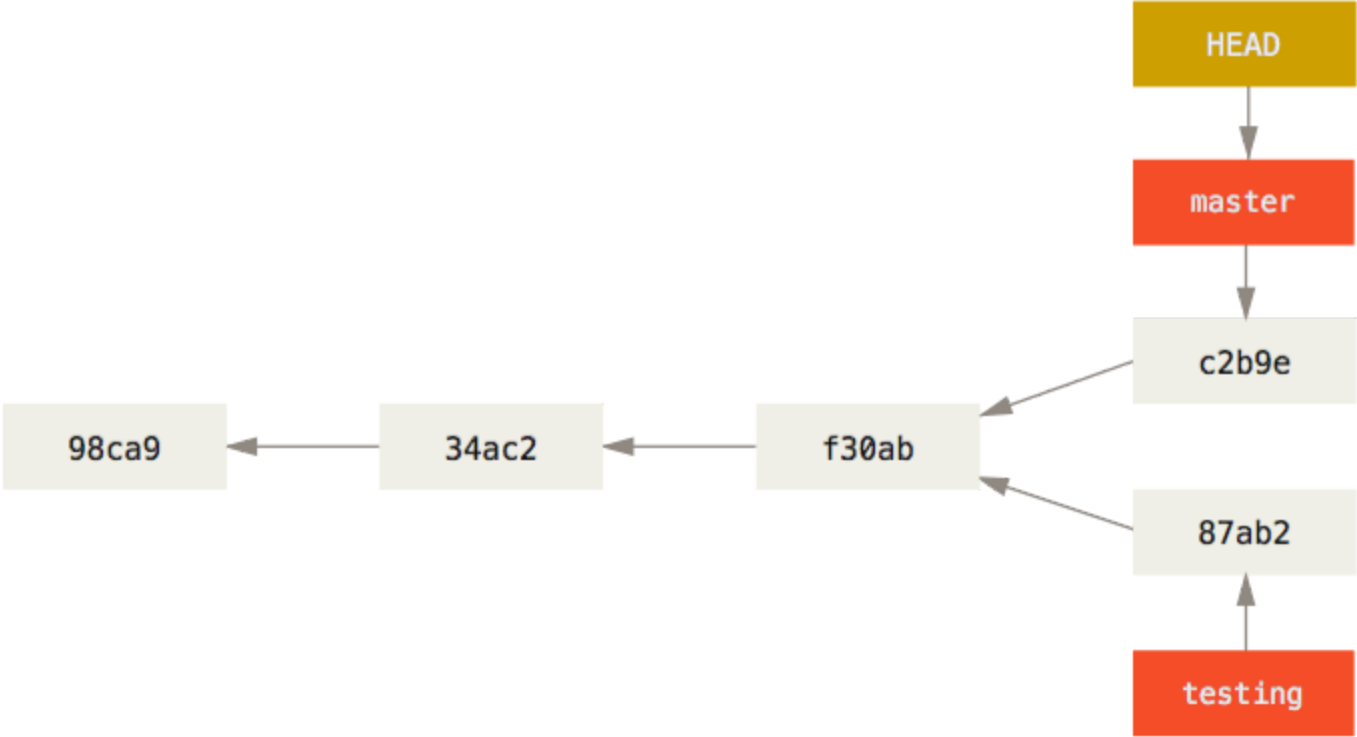
- Saltar a otro commit con `git checkout`
- Borrar commits con `git reset`
- Ver el histórico de HEADs con `git reflog`
- Apartar cambios provisionalmente con `git stash`
 - `git stash`
 - `git stash list`
 - `git stash show stash@{n} -v`
 - `git stash apply [stash@{n}]`
 - `git stash pop [stash@{n}]`

El log

- Saltar a otro commit con `git checkout`
- Borrar commits con `git reset`
- Ver el histórico de HEADs con `git reflog`
- Apartar cambios provisionalmente con `git stash`
 - `git stash`
 - `git stash list`
 - `git stash show stash@{n} -v`
 - `git stash apply [stash@{n}]`
 - `git stash pop [stash@{n}]`
 - `git stash drop [stash@{n}]`

Branching

Ramas



Ramas

Ramas

- Referencia

Ramas

- Referencia
- La rama master

Ramas

- Referencia
- La rama master
- Listar ramas

Ramas

- Referencia
- La rama master
- Listar ramas
- Crear una rama con git branch

Ramas

- Referencia
- La rama master
- Listar ramas
- Crear una rama con git branch
- Ir a una rama con git checkout

Ramas

- Referencia
- La rama master
- Listar ramas
- Crear una rama con git branch
- Ir a una rama con git checkout
- Borrar una rama con git branch -d

Ramas

- Referencia
- La rama master
- Listar ramas
- Crear una rama con `git branch`
- Ir a una rama con `git checkout`
- Borrar una rama con `git branch -d`
- Mezclar dos ramas con `git merge`

Ramas

- Referencia
- La rama master
- Listar ramas
- Crear una rama con `git branch`
- Ir a una rama con `git checkout`
- Borrar una rama con `git branch -d`
- Mezclar dos ramas con `git merge`
- Mezclar dos ramas con `git rebase`

Ramas

- Referencia
- La rama master
- Listar ramas
- Crear una rama con `git branch`
- Ir a una rama con `git checkout`
- Borrar una rama con `git branch -d`
- Mezclar dos ramas con `git merge`
- Mezclar dos ramas con `git rebase`
- Aplicar un commit de otra rama con `cherry-pick -x`

El modelo git flow

El modelo git flow

- master branch
- bugfix branch

El modelo git flow

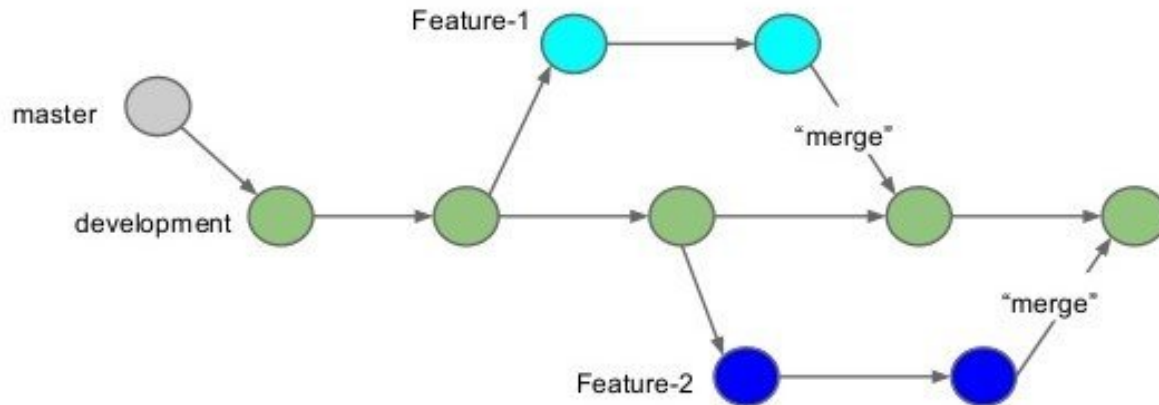
- master branch
- develop branch
- bugfix branch
- hotfix branch

El modelo git flow

- master branch
- develop branch
- feature branch
- bugfix branch
- hotfix branch
- release branch

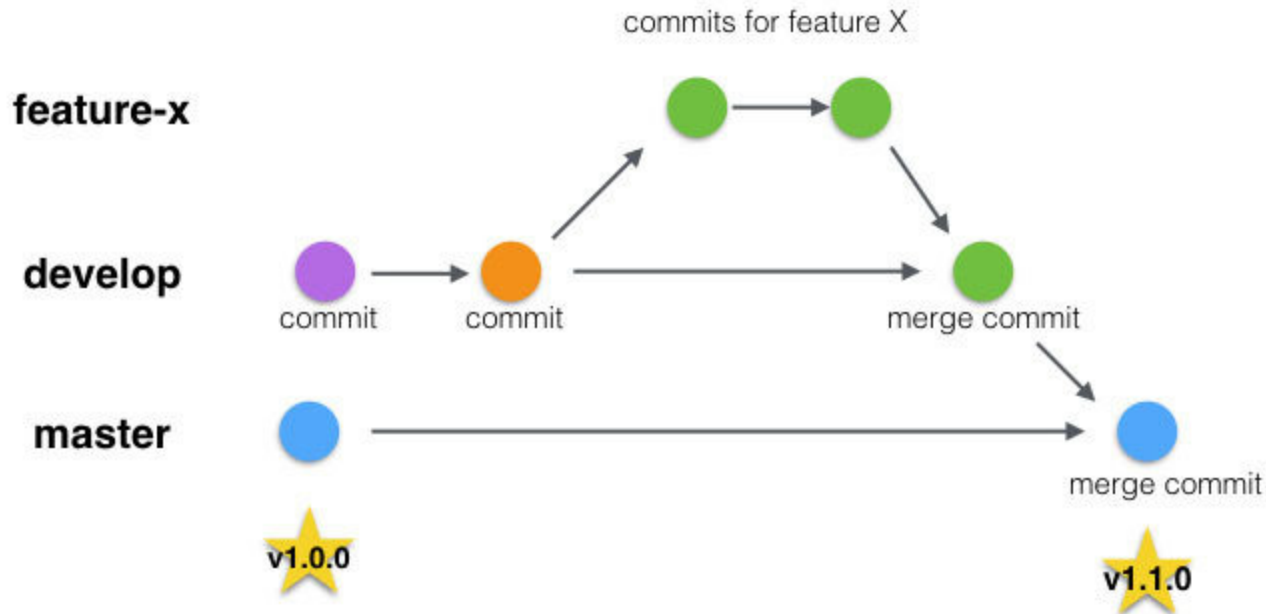
El modelo git flow

- master branch
- develop branch
- feature branch
- bugfix branch
- hotfix branch
- release branch



El modelo git flow

- master branch
- develop branch
- feature branch
- bugfix branch
- hotfix branch
- release branch



Repositorios remotos

Repositorios online

Repositorios online

- [GitHub](#)

Repositorios online

- [GitHub](#)
- [Bitbucket](#)

Repositorios online

- [GitHub](#)
- [Bitbucket](#)
- [GitLab](#)

Repositorios remotos

Repositorios remotos

- Clonar con git clone

Repositorios remotos

- Clonar con git clone
- El remote origin

Repositorios remotos

- Clonar con `git clone`
- El remote origin
- Añadir un remote con `git remote add`

Repositorios remotos

- Clonar con `git clone`
- El remote origin
- Añadir un remote con `git remote add`
- Renombrar un remote con `git remote rename`

Repositorios remotos

- Clonar con `git clone`
- El remote origin
- Añadir un remote con `git remote add`
- Renombrar un remote con `git remote rename`
- Borrar un remote con `git remote remove`

Repositorios remotos

- Clonar con `git clone`
- El remote origin
- Añadir un remote con `git remote add`
- Renombrar un remote con `git remote rename`
- Borrar un remote con `git remote remove`
- Tracking branch

Repositorios remotos

Repositorios remotos

- Enviar cambios al repositorio remoto con git push

Repositorios remotos

- Enviar cambios al repositorio remoto con git push
- Sincronizar mi copia local del remoto con git fetch

Repositorios remotos

- Enviar cambios al repositorio remoto con git push
- Sincronizar mi copia local del remoto con git fetch
- Mergear cambios del remoto con git merge

Repositorios remotos

- Enviar cambios al repositorio remoto con git push
- Sincronizar mi copia local del remoto con git fetch
- Mergear cambios del remoto con git merge
- git fetch + git merge = git pull

Repositorios remotos

- Enviar cambios al repositorio remoto con git push
- Sincronizar mi copia local del remoto con git fetch
- Mergear cambios del remoto con git merge
- git fetch + git merge = git pull
- Borrar ramas del remoto

Repositorios remotos

- Enviar cambios al repositorio remoto con `git push`
- Sincronizar mi copia local del remoto con `git fetch`
- Mergear cambios del remoto con `git merge`
- `git fetch + git merge = git pull`
- Borrar ramas del remoto
- Marcar ramas remotas borradas con `git fetch --prune`

Repositorios remotos

- Enviar cambios al repositorio remoto con `git push`
- Sincronizar mi copia local del remoto con `git fetch`
- Mergear cambios del remoto con `git merge`
- `git fetch + git merge = git pull`
- Borrar ramas del remoto
- Marcar ramas remotas borradas con `git fetch --prune`
- Ver lo que voy a enviar en el push con `git diff origin/master master`

GUI

GUI para Git

GUI para Git

- Git GUI (instalada con Git)

GUI para Git

- Git GUI (instalada con Git)
- [TortoiseGit](#)

GUI para Git

- Git GUI (instalada con Git)
- [TortoiseGit](#)
- [SourceTree](#)

GUI para Git

- Git GUI (instalada con Git)
- TortoiseGit
- SourceTree
- GitKraken

GUI para Git

- Git GUI (instalada con Git)
- [TortoiseGit](#)
- [SourceTree](#)
- [GitKraken](#)
- Plugin [EGit](#) para Eclipse

Git & Maven

Git & Maven

Git & Maven

- Añadir al `.gitignore` los directorios de builds y de dependencias ([ejemplo](#))

Git & Maven

- Añadir al `.gitignore` los directorios de builds y de dependencias ([ejemplo](#))
- Si hay jerarquía de repositorios: `git submodule`

Git & Maven

- Añadir al `.gitignore` los directorios de builds y de dependencias ([ejemplo](#))
- Si hay jerarquía de repositorios: `git submodule`
 - `git submodule add URL`

Git & Maven

- Añadir al `.gitignore` los directorios de builds y de dependencias ([ejemplo](#))
- Si hay jerarquía de repositorios: `git submodule`
 - `git submodule add URL`
 - `git clone URL --recurse-submodules`
o
`git clone URL`
`git submodule init`
`git submodule update`

Pull Requests

Pull Requests

Pull Requests

- En equipos grandes

Pull Requests

- En equipos grandes
- Los miembros no pueden mergear sus ramas a develop

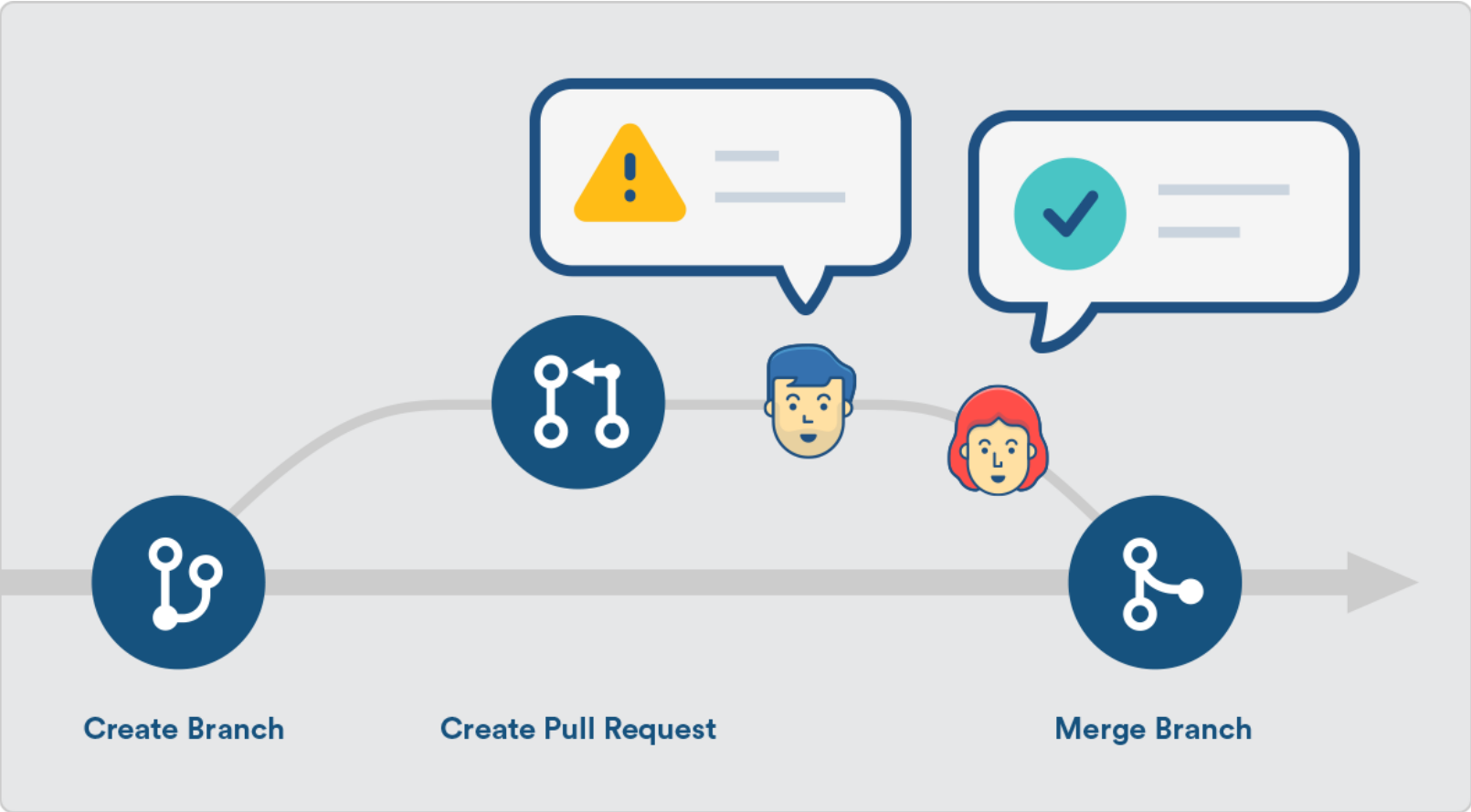
Pull Requests

- En equipos grandes
- Los miembros no pueden mergear sus ramas a develop
- Se solicita el merge a través de la interfaz del servidor

Pull Requests

- En equipos grandes
- Los miembros no pueden mergear sus ramas a develop
- Se solicita el merge a través de la interfaz del servidor
- Hay un encargado (o varios) de revisar las peticiones de PR, y de realizar el merge de las ramas a develop.

Pull Requests



Links

Links

- [Documentación oficial de Git](#)

Links

- [Documentación oficial de Git](#)
- [Libro Pro Git](#)

Links

- [Documentación oficial de Git](#)
- [Libro Pro Git](#)
- [Git Flow](#)

Links

- [Documentación oficial de Git](#)
- [Libro Pro Git](#)
- [Git Flow](#)
- [Artículo sobre Git Flow](#)

Links

- [Documentación oficial de Git](#)
- [Libro Pro Git](#)
- [Git Flow](#)
- [Artículo sobre Git Flow](#)
- [Explicación de git reset](#)

Links

- [Documentación oficial de Git](#)
- [Libro Pro Git](#)
- [Git Flow](#)
- [Artículo sobre Git Flow](#)
- [Explicación de git reset](#)
- [SubGit](#)

Links

- [Documentación oficial de Git](#)
- [Libro Pro Git](#)
- [Git Flow](#)
- [Artículo sobre Git Flow](#)
- [Explicación de git reset](#)
- [SubGit](#)
- [Capítulo de Pro Git sobre submódulos](#)

mario@mariogl.com
@marioglweb