



git

Novembre 2023



Presentación

- Nombre
- Puesto de trabajo
- Experiencia
- Tecnologías
- Conocimiento y experiencia con controles de versiones
- Conocimiento y experiencia específicos de GIT
- Correo a utilizar para el curso
- Contacto: formacion.ecs.es@arrow.com

Software necesario

- GIT - <https://git-scm.com>
- SourceTree - <https://www.sourcetreeapp.com/>
- VS Code - <https://code.visualstudio.com/>

Introducción a GIT

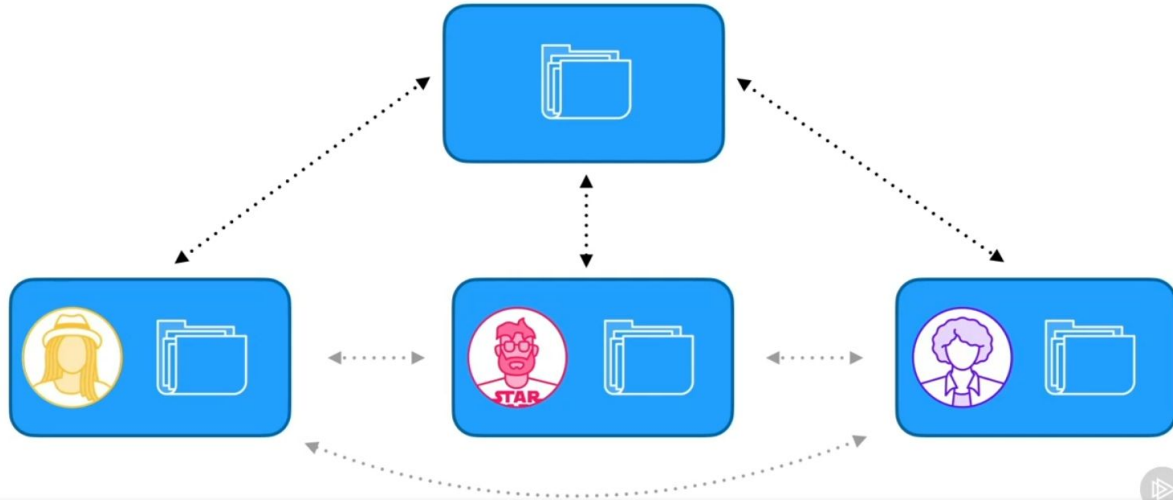
- ¿Qué es GIT?
- Conceptos básicos teóricos de GIT
 - Control de versiones distribuido
 - Características de GIT
 - Ficheros en GIT y el estatus
- Introducción a línea de comando
- Comandos básicos de GIT
 - config (configurar entorno)
 - init
 - add
 - status
 - commit
 - log
 - diff

¿Qué es GIT?

- GIT es un sistema de control de versiones diseñado para manejar todo tipo de proyectos de todo tamaño con velocidad y eficiencia
- Permite volver a cualquier estado del proyecto en cualquier momento del tiempo
- Un clon local del proyecto es un repositorio de control de versiones completo

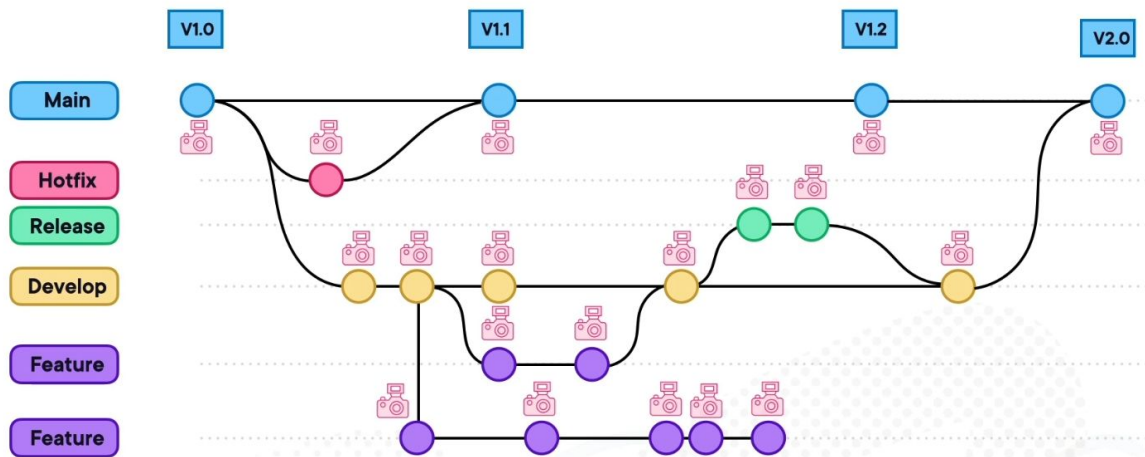


¿Qué es un sistema de control de versiones distribuido?

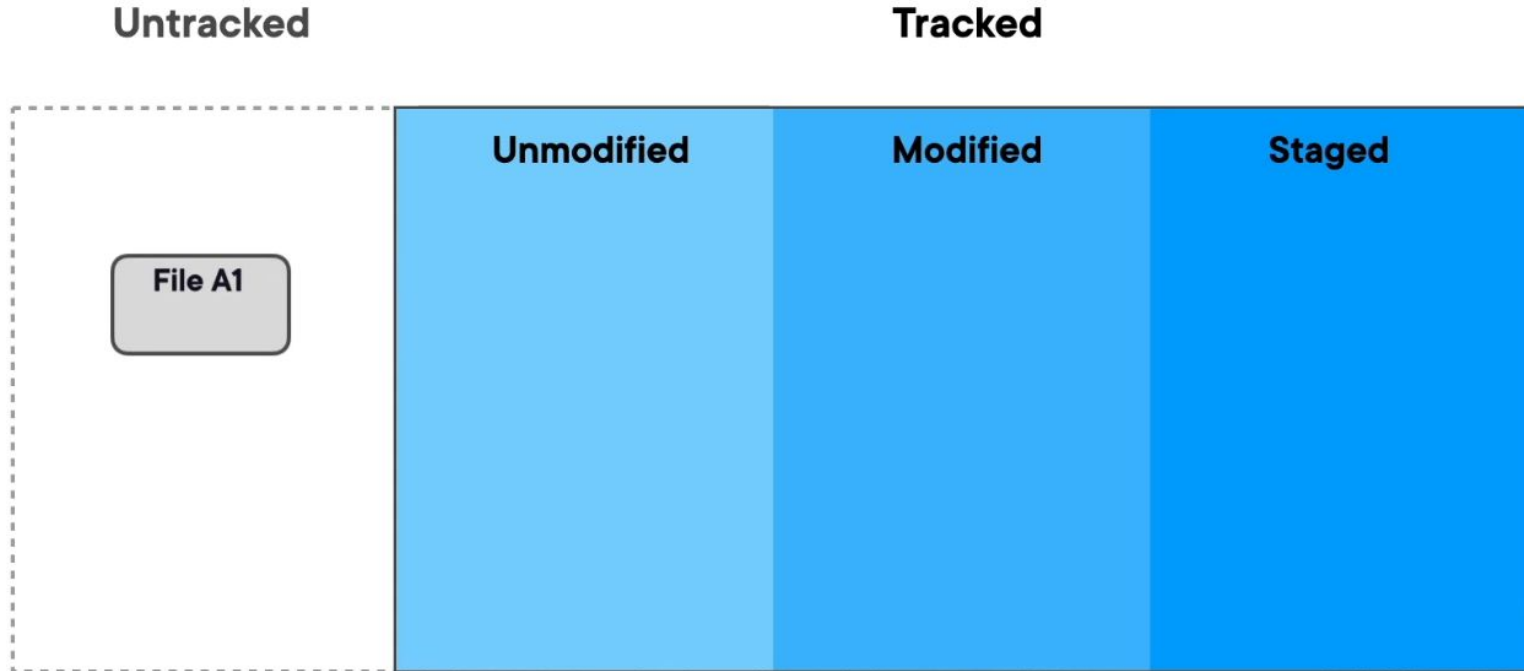


Características de GIT

- Se puede trabajar completamente offline
- Cualquier cambio es intencionado: GIT no realiza acciones por sí solo
- Está diseñado para desarrollos no lineales
- Snapshots: manera en que se guarda el estado de algo en un punto específico del tiempo



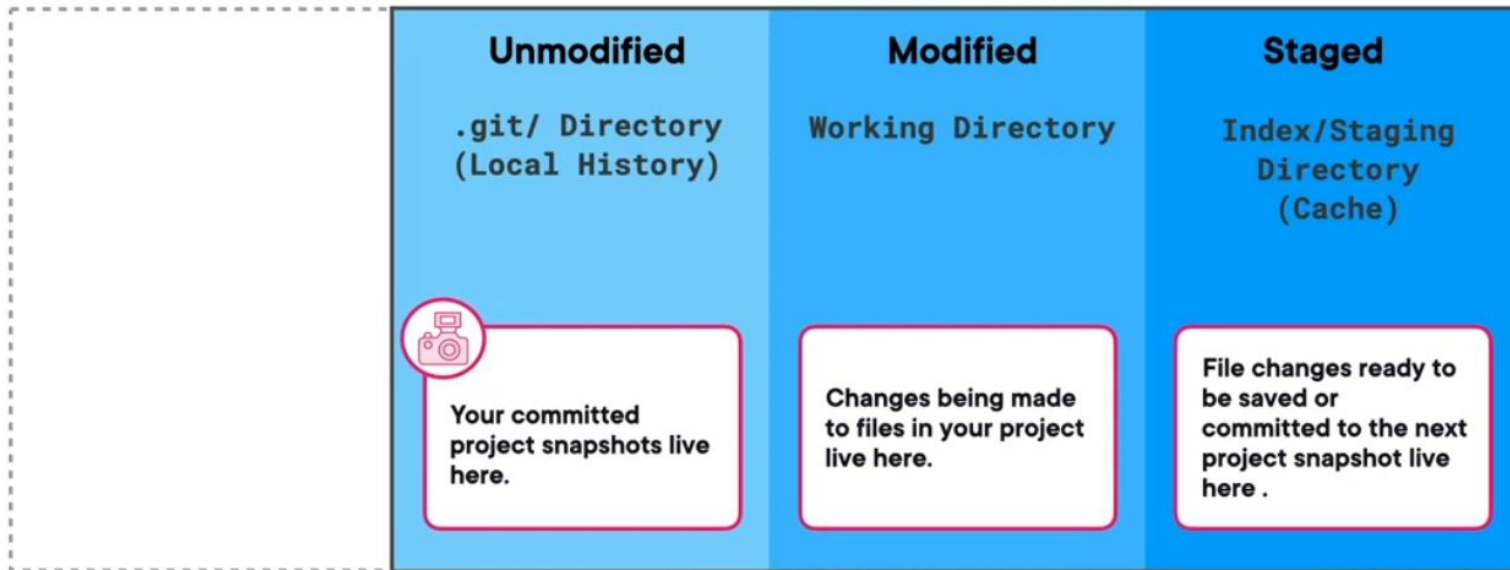
Ficheros en GIT: status



Directorios en GIT: status

Untracked

Tracked



Introducción a línea de comando

pwd

mkdir

ls

cd

touch

Comandos básicos de GIT

Git Basics

<code>git init</code>	Create a new .git repository and begins tracking
<code>git add</code>	Move modified files into the staging area
<code>git status</code>	Shows you the status of your files
<code>git commit</code>	Create a snapshot and commit to .git
<code>git config</code>	Set and read specific Git configurations
<code>git log</code>	Shows the committed snapshot history
<code>git diff</code>	Shows changes between your working directory and staging area

Comandos de GIT

Git Basics

<code>git init</code>	Create a new .git repository and begins tracking
<code>git add</code>	Move modified files into the staging area
<code>git status</code>	Shows you the status of your files
<code>git commit</code>	Create a snapshot and commit to .git
<code>git config</code>	Set and read specific Git configurations
<code>git log</code>	Shows the committed snapshot history
<code>git diff</code>	Shows changes between your working directory and staging area

Git Branches

<code>git branch</code>	List, create, or delete branches
<code>git checkout</code>	Switch between branches
<code>git merge</code>	Bring changes from one branch into another

Remote Repositories

<code>git clone</code>	Copies an entire repository into a new local .git directory
<code>git remote</code>	Create and show linked repositories
<code>git push</code>	Send updates to associate repositories
<code>git pull</code>	Retrieves and integrates changes from other repositories
<code>git fetch</code>	Retrieves but doesn't integrate changes from other repositories

Undoing Changes

<code>git revert</code>	Create a new commit that undoes a previous commit
<code>git reset</code>	Remove files from the staging area

Niveles de configuración de GIT



`--system`

Username: aaron.stewart
Email: aaron.stewart@company.com



`--global`

Username: a-a-ron
Email: personal@gmail.com



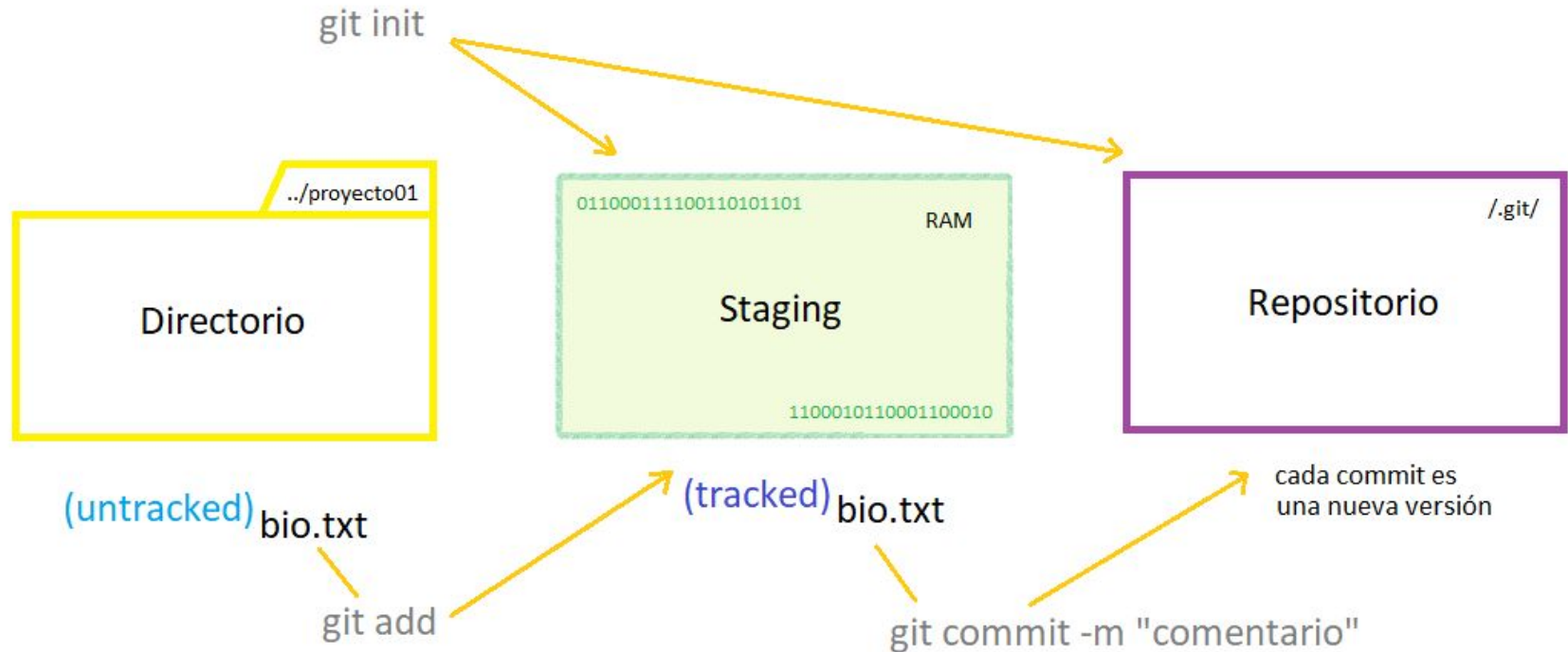
`--local`

Username: astewart01
Email: student@university.com

Conceptos básicos de GIT

- Instalación de GIT
- Uso de git bash
- Niveles de configuración de GIT
- Diferencia entre windows y unix based
- Comandos y ayuda
- Uso de tags
- Versiones
- Releases
- Versiones vs Releases

Repositorio Local



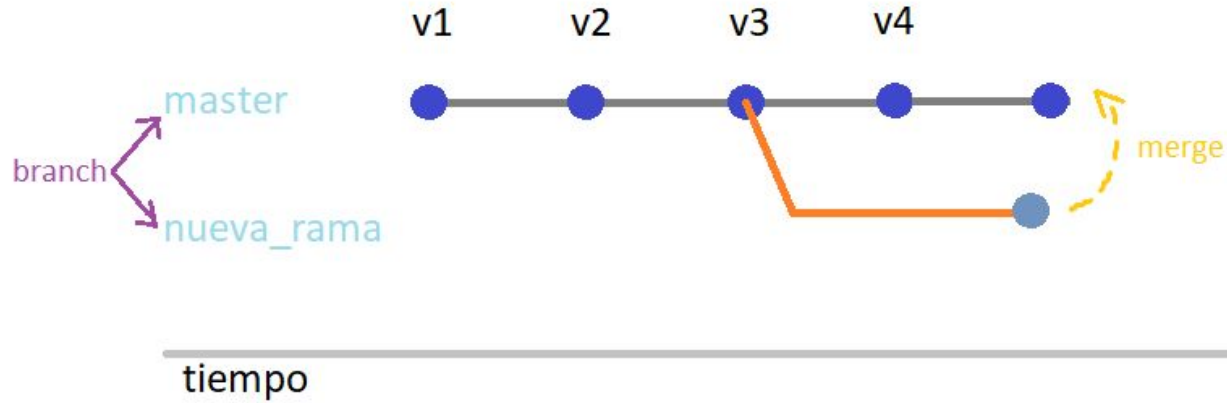
Repositorios y flujo de trabajo

- Trabajar con otros en un repositorio compartido
- Conflictos con ficheros: qué son y cómo arreglarlos
- Modificar y arreglar commits
- Casos prácticos
 - Flujos de trabajo
- Malas prácticas

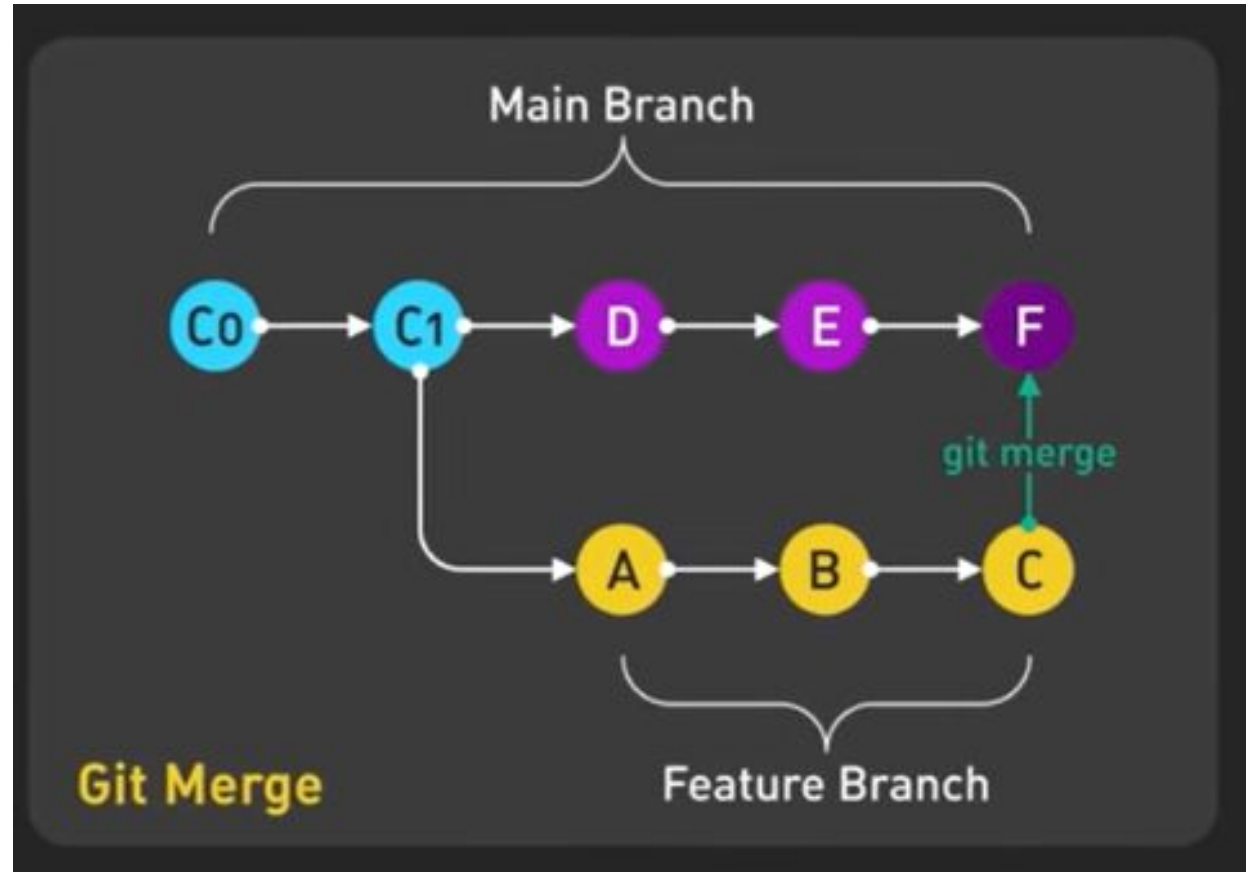
Repositorios remotos – GitHub

- ¿Qué es GitHub?
- Instalación de GITHUB
- Creación de repos
- Clonación de repositorios
- Trabajo en equipo
- Uso de grupos y administración
- Permisos
- GitHub pages
- Alternativas a GitHub

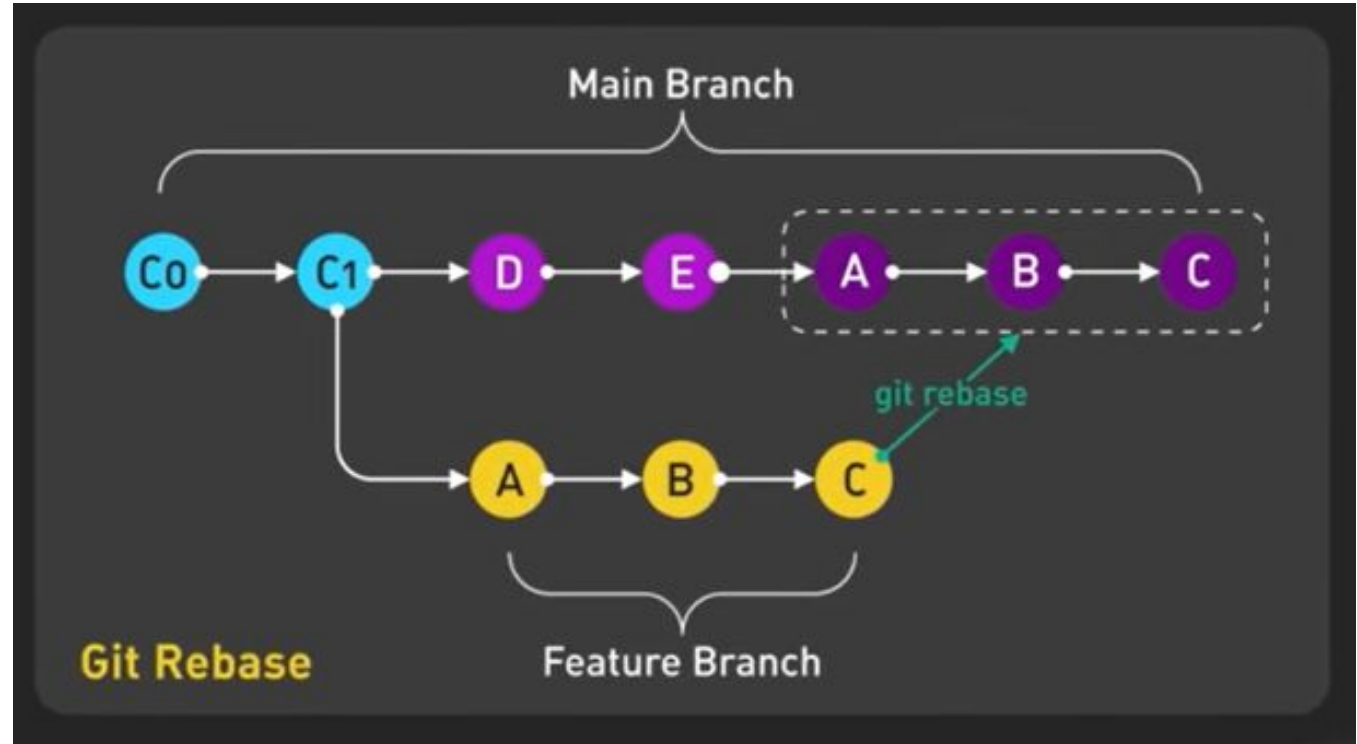
Ramas (branch)



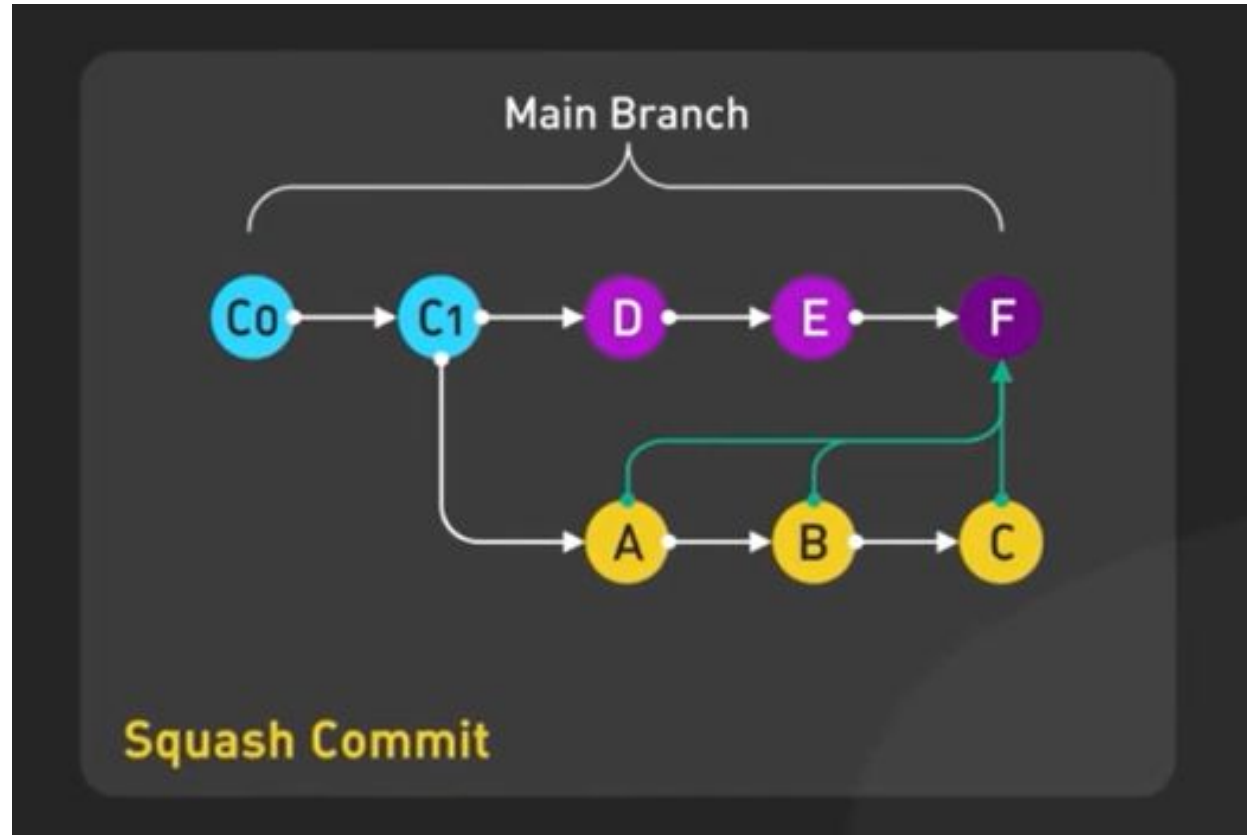
Merge



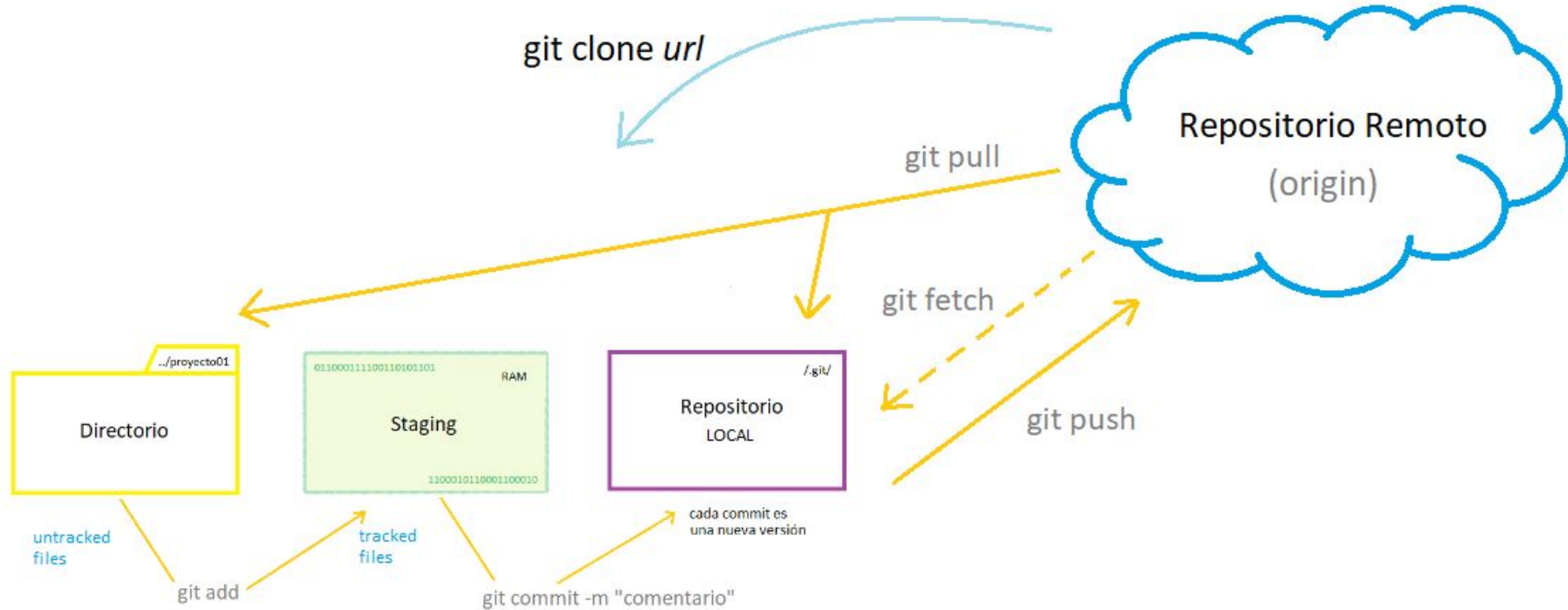
Rebase



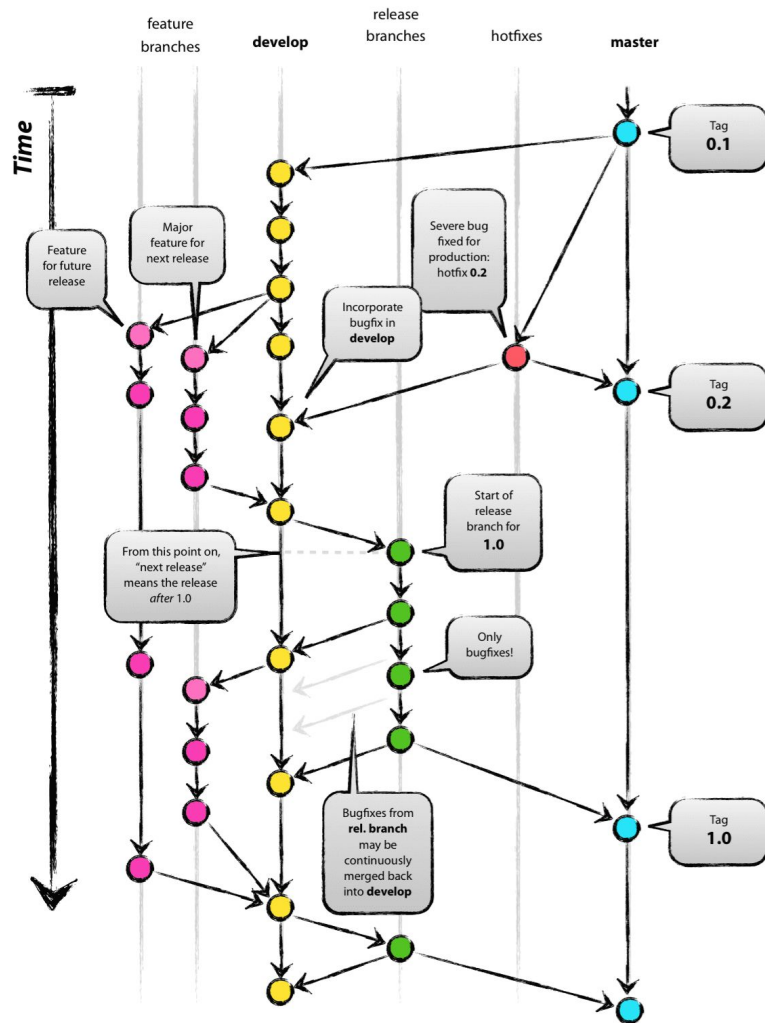
Squash



Repositorio Local vs Remoto



Git Flow



Estrategias de branching

- Mainline based
- Branch based
 - GitFlow - el habitual
 - main
 - develop
 - feature
 - release
 - hotfix
 - GitHub flow - igual pero sin release
 - GitLab flow - añade ramas por entorno
 - OneFlow - no tiene develop