

Introducción a Git

Git

- Software de control de versiones
- Almacena nuestros cambios de códigos

Git

- **Nueva cuenta**
- ¿Qué es Git?
- Comandos

Creando una cuenta

- Cualquier proveedor nos sirve igual
- Usaremos GitHub como ejemplo
- Lo importante es la URL del repositorio

Git

- Nueva cuenta
- **¿Qué es Git?**
- Comandos

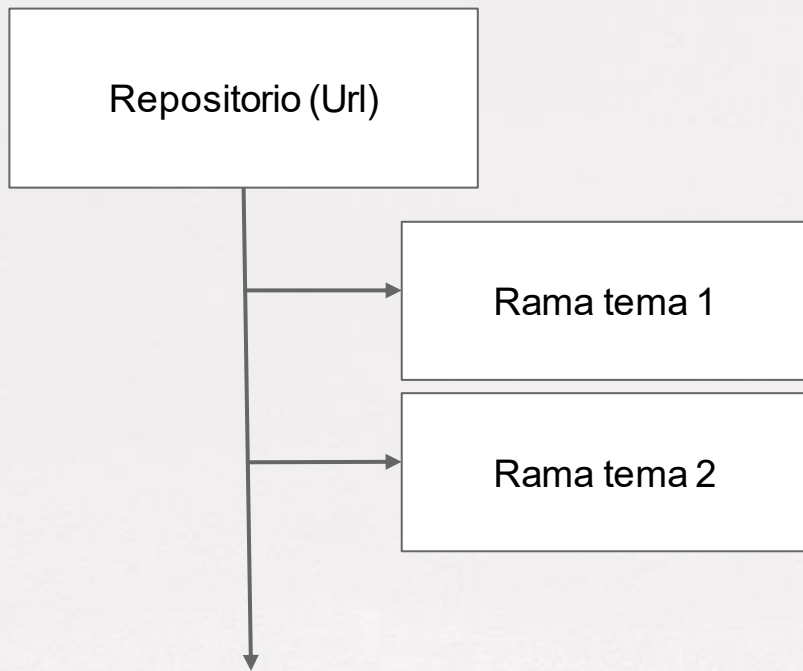
¿Qué es Git?

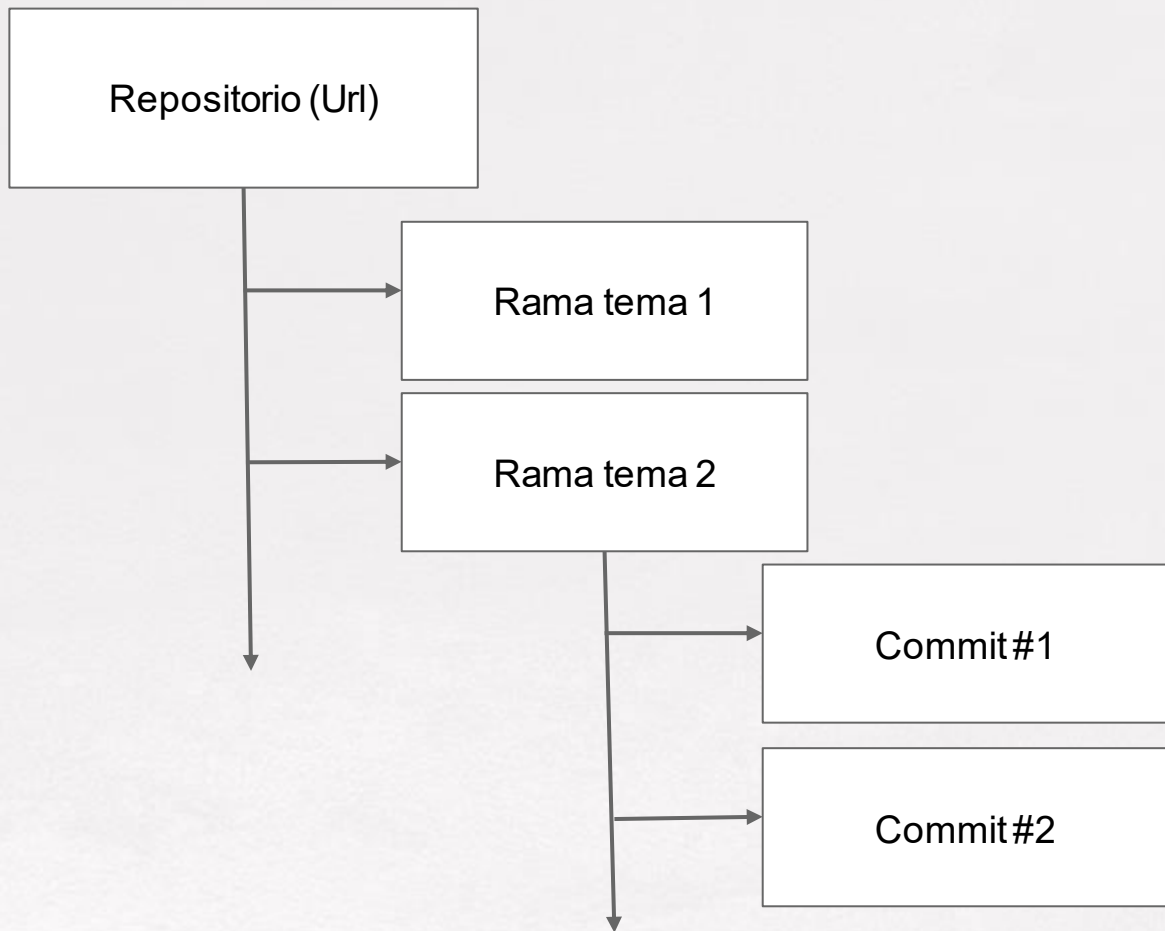
- Repositorios: almacenes de copias de seguridad
- Ramas: copia concreta
- Commit: punto de guardado en esa rama

¿Qué es Git?

- Sistema distribuido y cada usuario tiene una copia
- Debemos actualizar nuestro cambios periódicamente
- Los cambios se deben guardar en el repositorio al finalizar

Repositorio (Url)





Git

- Nueva cuenta
- ¿Qué es Git?
- **Comandos**

Comandos

- Resumen de comandos para usar en el curso
- Ampliar conocimientos sobre el uso de Git:
 - [Cursos de OpenWebinars](#)

Comandos

Repositorios:

- **git init**: Inicia un nuevo proyecto Git localmente.
- **git clone**: Conecta con un proveedor de Git y descarga el repositorio.

Comandos

- **git remote add origin repositorio.git**: Conecta con un repositorio dado
- **git fetch**: Obtener las ramas del repositorio

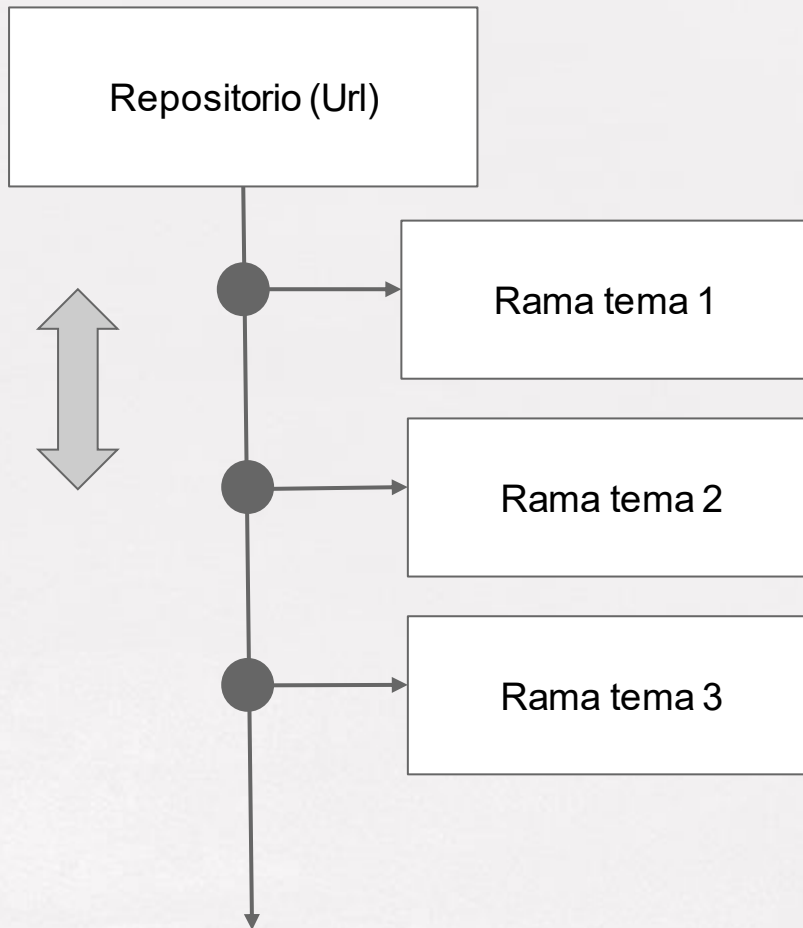
Esta opción será necesaria en nuestro proyecto al haber escogido la opción de crear repositorio y no iniciarlo

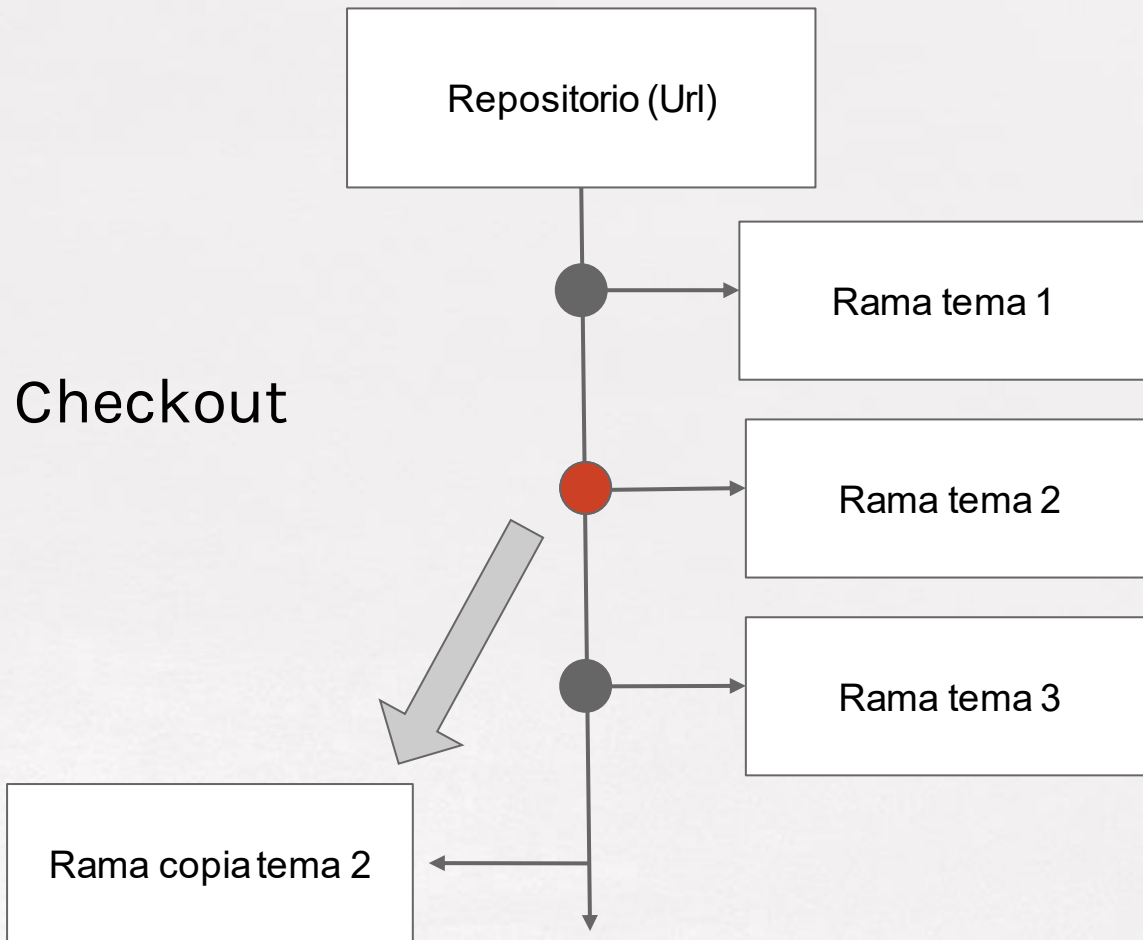
COMANDOS

Ramas:

- **git checkout rama**: Nos movemos a una rama ya creada (Movernos)
- **git checkout -b rama-nueva**: Creamos y nos movemos a una nueva rama (Nueva copia)

Checkout

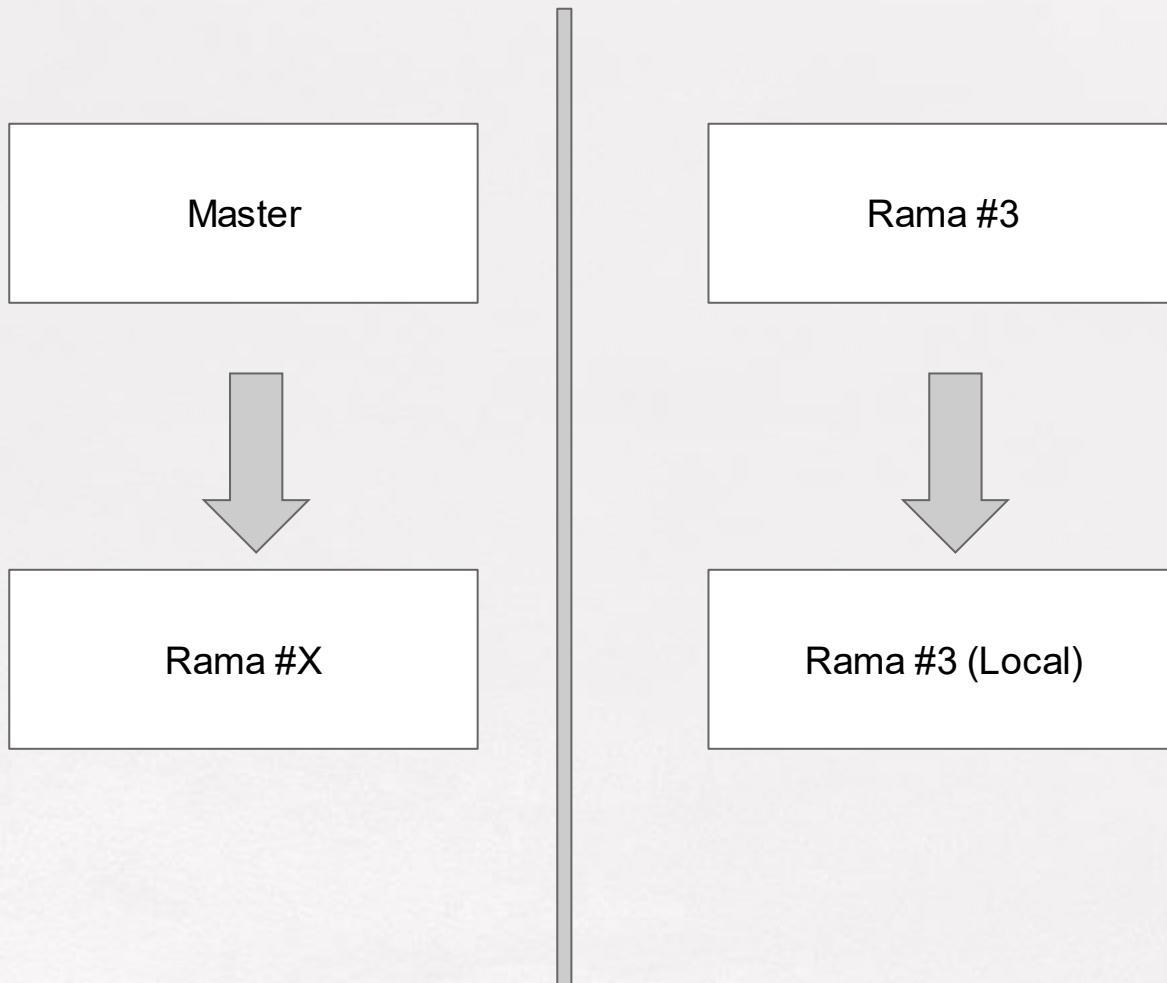




Comandos

Ramas:

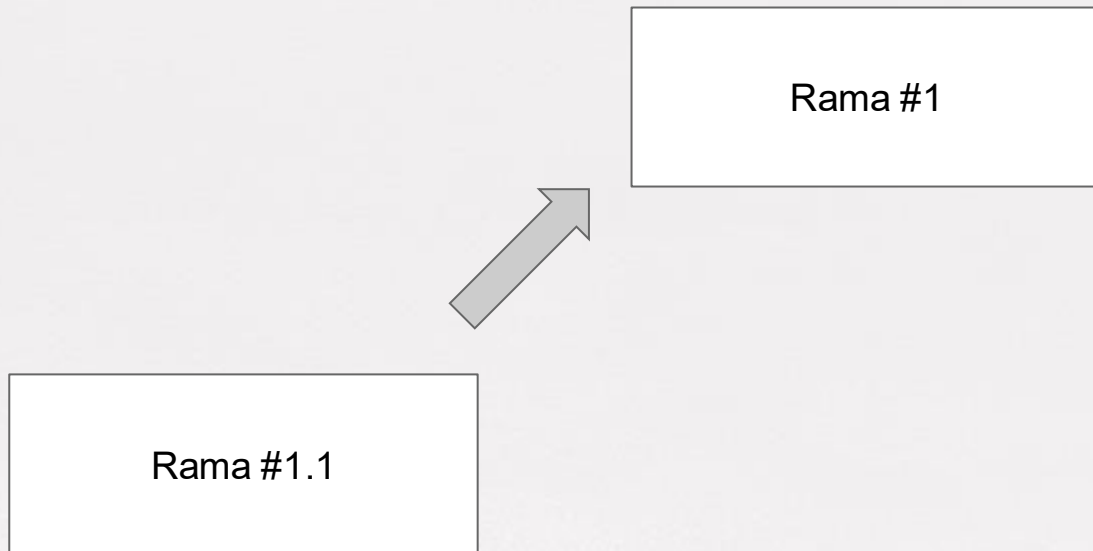
- **git pull rama-origen**: Actualizamos la rama actual con los nuevos cambios (Actualizar)



Comandos

Ramas:

- **git push origin rama-destino**: Guardamos los commits en la rama-destino (Guardar)

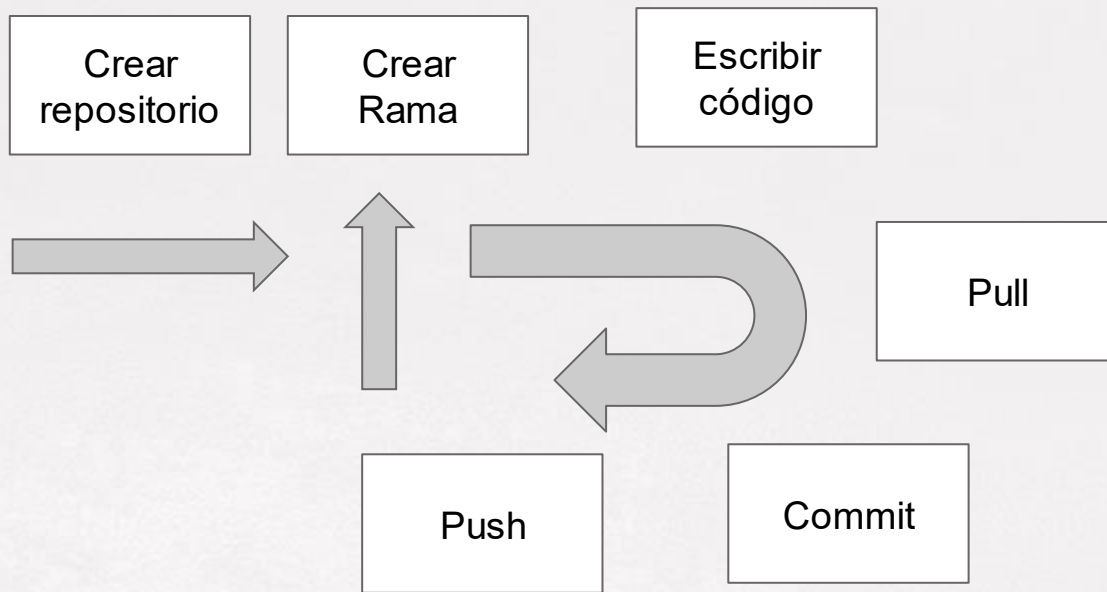


Comandos

Commit:

- **git add**: Nos permite añadir ficheros al próximo commit
- **git commit -m 'texto'**: El guardado de código junto con un comentario descriptivo

Flujo de trabajo



¡Gracias!