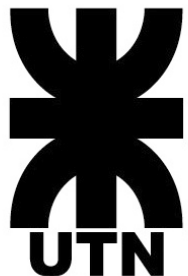


Sistemas de Control de Versiones (Introducción a Git)

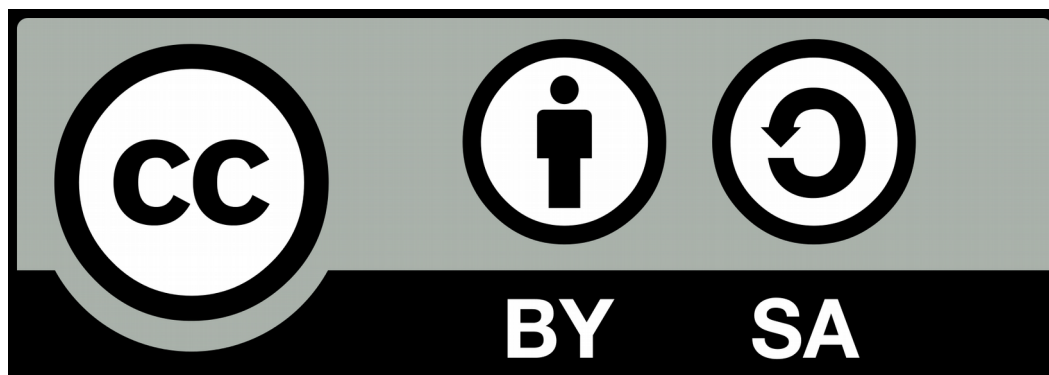
Dr. Ricardo Medel

ricardo.h.medel@gmail.com



Ingeniería de Software de Fuentes Abiertas/Libre – Cód. 845
Departamento de Ingeniería en Sistemas de Información
UTN-FRC - Año 2018

Licencia Creative Commons

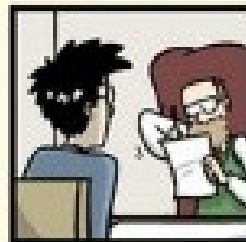


Esta obra está bajo una Licencia
Creative Commons Atribución-
CompartirIgual 4.0 Internacional

"FINAL".doc



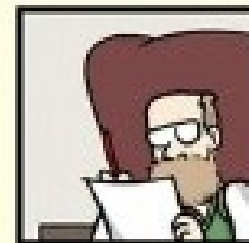
FINAL.doc!



FINAL_rev.2.doc



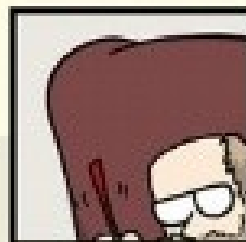
FINAL_rev.6.COMMENTS.doc



FINAL_rev.8.comments5.
CORRECTIONS.doc



FINAL_rev.18.comments7.
corrections9.MORE.30.doc



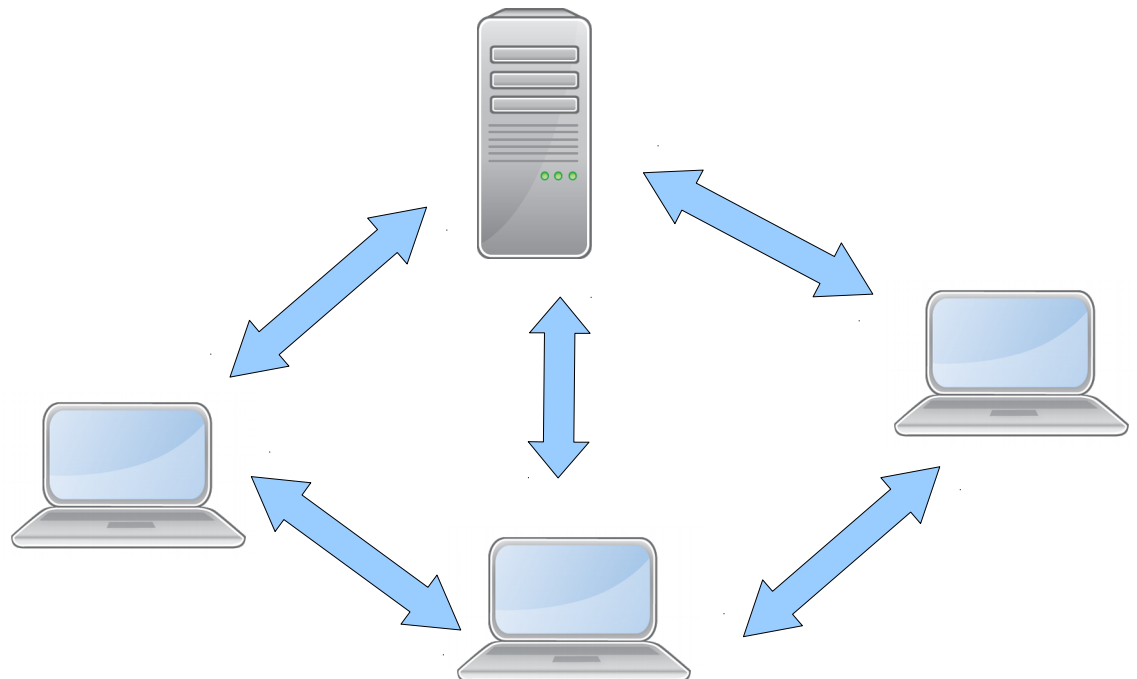
FINAL_rev.22.comments49.
corrections.10.##\$%WHYDID
ICOMETOGRADSCHOOL????.doc

Sistemas de Control de Versiones

- Permiten manejar el histórico de archivos y carpetas de un proyecto, a fin de recuperar cierto estado en caso de necesidad.
- Se quiere:
 - ◇ Saber qué cambios se realizaron
 - ◇ Saber quién y cuándo realizó los cambios
 - ◇ Recuperar el estado del proyecto previo a los cambios
 - ◇ Permitir hacer cambios en paralelo

Sistema de Control de Versiones Distribuido

- No tiene necesariamente **repositorio central**.
- Permite trabajo en **ramas**, para trabajo en equipo.



Crear o clonar un repositorio en máquina local

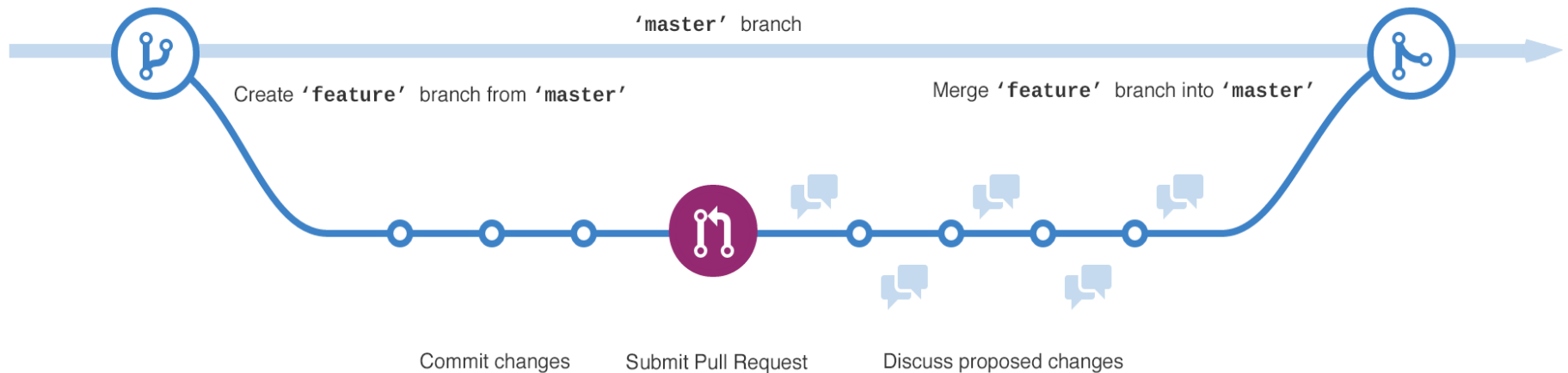
```
git init repo
```

```
git clone URL/repo
```

```
git pull origin master
```

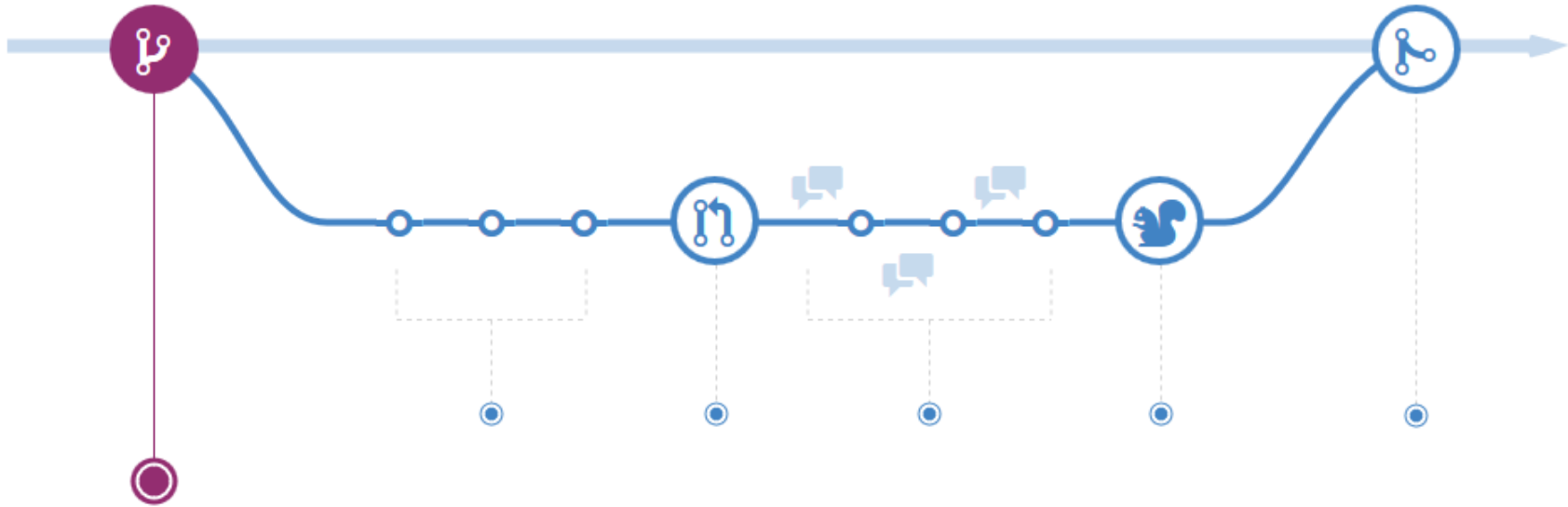
- El repositorio local es donde se realizan los cambios

Flujo de control (simplificado)



- Abrir rama
- Hacer cambios
- Hacer “commit”
- Sincronizar con repositorios remotos

Crear una rama (“branch”)

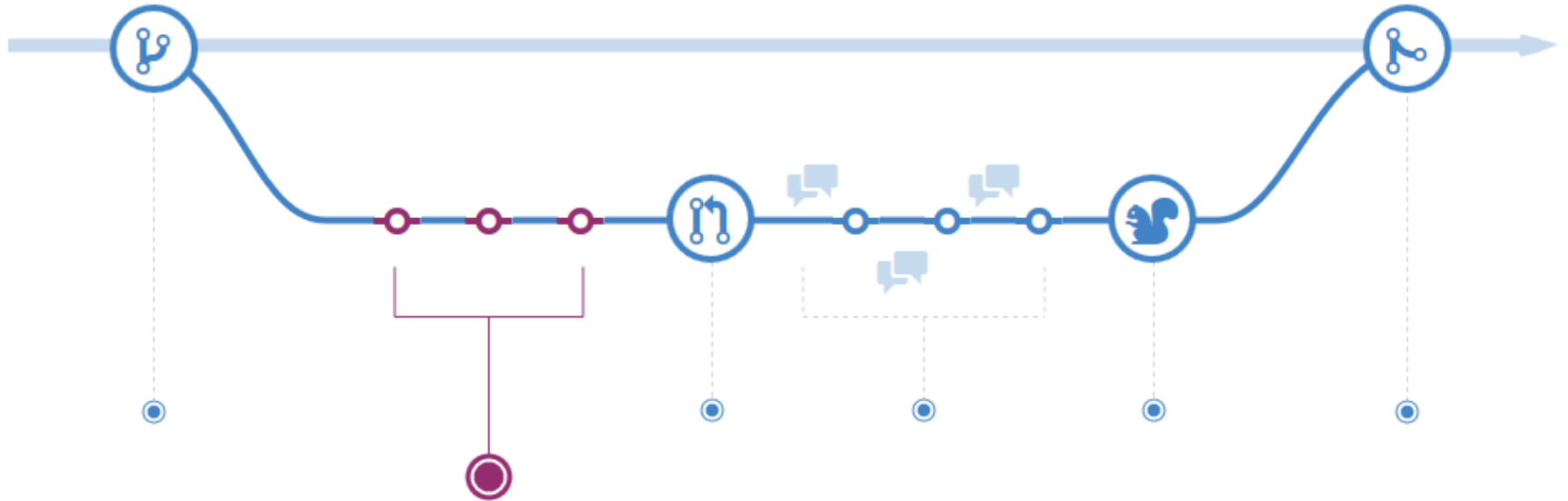


```
git branch rama
```

```
git checkout rama
```

- Las ramas minimizan los cambios en el repositorio principal
- Permiten manejar versiones distintas de un mismo proyecto

Realizar cambios y “commit”

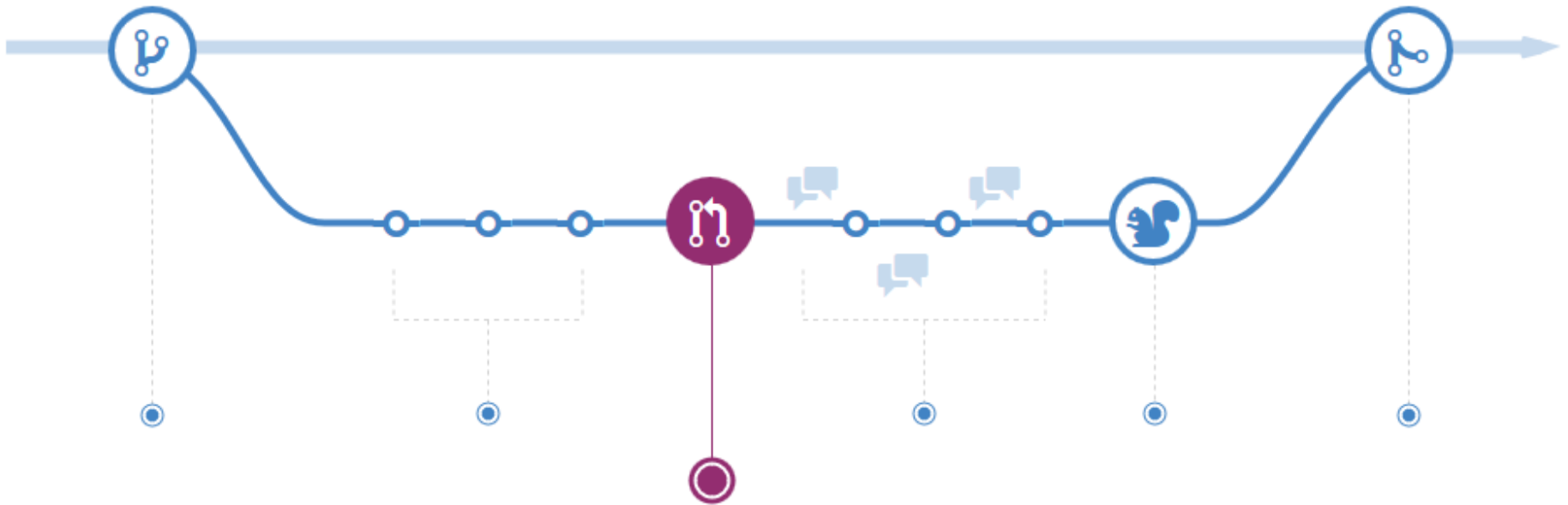


```
git add archivo
```

```
git commit -m 'comentario explicando el cambio'
```

- Add agrega un archivo al área de “staging”
- Commit sube nuevos archivos y modificados al repositorio local

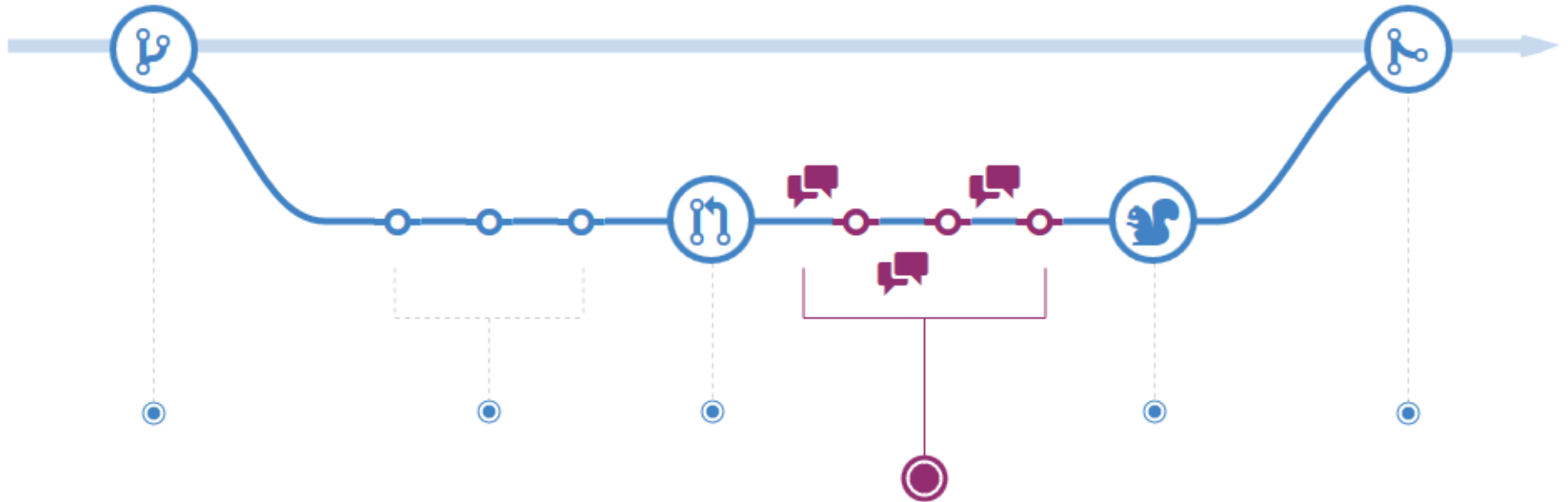
“Push” cambios en repo



```
git push origin rama
```

- Ahora los cambios locales están visibles para todos los usuarios del repo

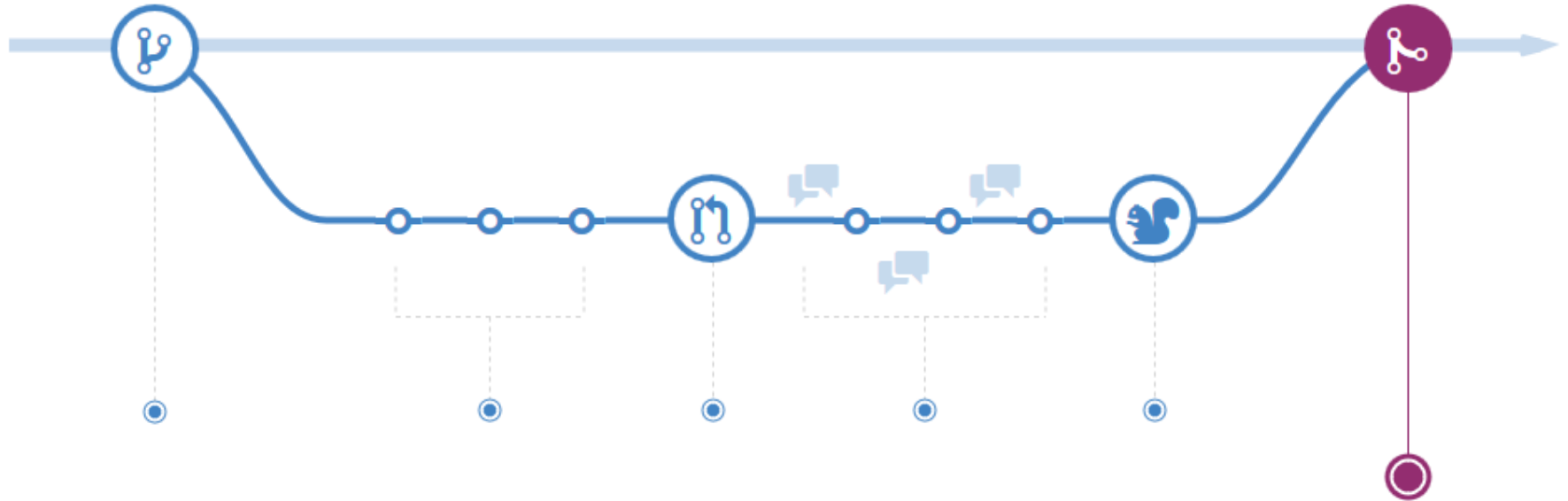
Solicitar “pull” (“pull request”)



`git request-pull` **rama** **URL** `origin`

- Permite que se revisen los cambios antes de combinarlos en la rama master
- No todos los repositorios lo soportan desde línea de comandos

Combinar ramas (“merge”)



`git checkout master`

`git merge rama`

- Combina la rama con el master

Otros comandos útiles

```
git status
```

```
git log
```

```
git diff
```

```
git fetch
```

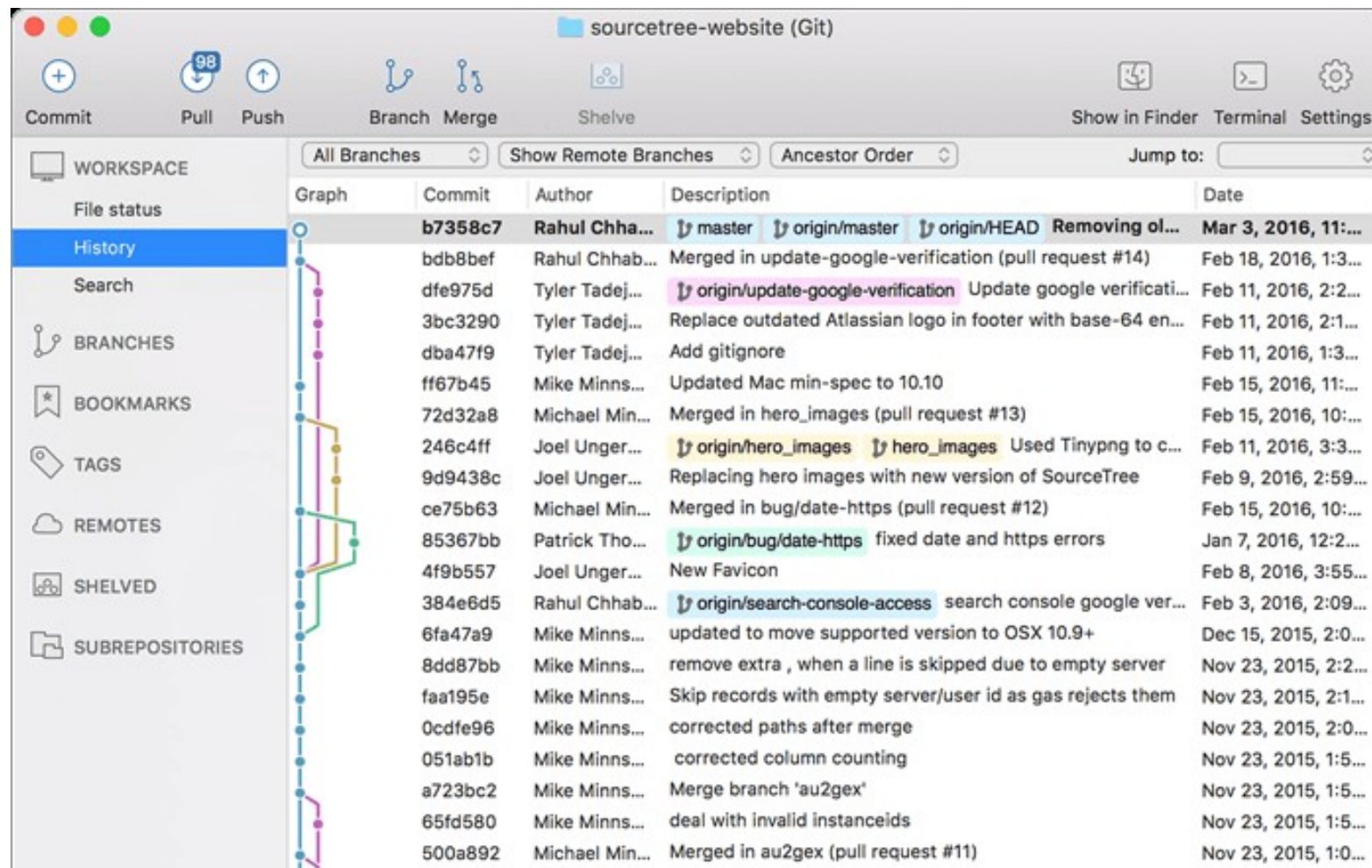
```
git config --global user.name "nombre de usuario"
```

```
git config --global user.email "usuario@email.com"
```

```
git help comando
```

Cientes GUI para Git

`https://git-scm.com/downloads/guis`





- Repositorio online basado en Git
- Incluye repositorios privados (pagos)
- 55+ millones de proyectos

<https://github.com>

The screenshot shows the GitHub interface for the repository 'RHMedel / puma', which is a fork of 'emravera/puma'. The repository description is 'A meta scripting programming language based on LayerD foundations. — Edit'. The repository statistics show 164 commits, 2 branches, 0 releases, and 9 contributors. The interface includes navigation tabs for 'Code', 'Pull requests', 'Wiki', 'Pulse', 'Graphs', and 'Settings'. The 'Code' tab is active, showing the repository's main page. The repository is currently on the 'master' branch. The interface also includes buttons for 'New pull request', 'New file', 'Upload files', 'Find file', and 'Download ZIP'.

GitHub repository page for **RHMedel / puma** (forked from **emravera/puma**).

Repository description: A meta scripting programming language based on LayerD foundations. — Edit

Repository statistics: 164 commits, 2 branches, 0 releases, 9 contributors.

Navigation tabs: Code, Pull requests, Wiki, Pulse, Graphs, Settings.

Repository actions: Branch: master, New pull request, New file, Upload files, Find file, HTTPS, <https://github.com/RHMedel/puma>, Download ZIP.

Trabajo Práctico N°1

- 1) Instale Git en su máquina local <https://git-scm.com>
- 2) Si lo desea, instale una GUI para Git
<https://git-scm.com/downloads/guis>
- 3) Si aún no lo ha hecho, suscríbase a Github
- 4) Interiorícese sobre el uso de Git y Github, puede usar:
<https://guides.github.com/activities/hello-world/>
- 5) Clone nuestro repositorio en su máquina local:
<https://github.com/RHMedel/ingSWL2018>