



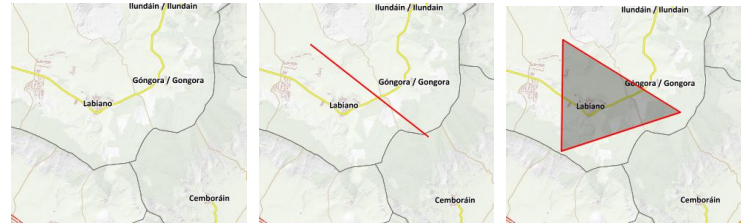
GIT

Isabel Vigil Morán

CONTROL DE VERSIONES

VCS (Version Control System)

- Almacena ficheros y conserva un registro histórico de los cambios que se han producido sobre estos, pudiendo recuperar un estado anterior del proyecto.
- Repositorio: lugar en el que se almacenan los datos actualizados y el histórico de cambios.
- Útil para desarrolladores, diseñadores...



LÍNEA DE COMANDOS

- La única forma de acceder a todas las posibilidades de Git.
- Disponible en todos los entornos.
- Una vez se maneja, cambiar a GUI es más sencillo.



CONFIGURACIÓN

```
$ git config --global user.name "Wonder Woman"
```

```
$ git config --global user.email diana@themiscyra.th
```

```
$ git config --list
```

```
$ git config -h
```

```
$ git help config
```



INICIALIZACIÓN

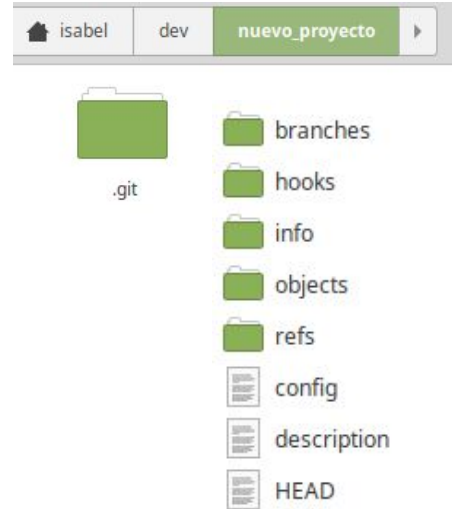
- Opción 1: convertir un proyecto local en un repositorio de Git

```
$ cd nuevo_proyecto
```

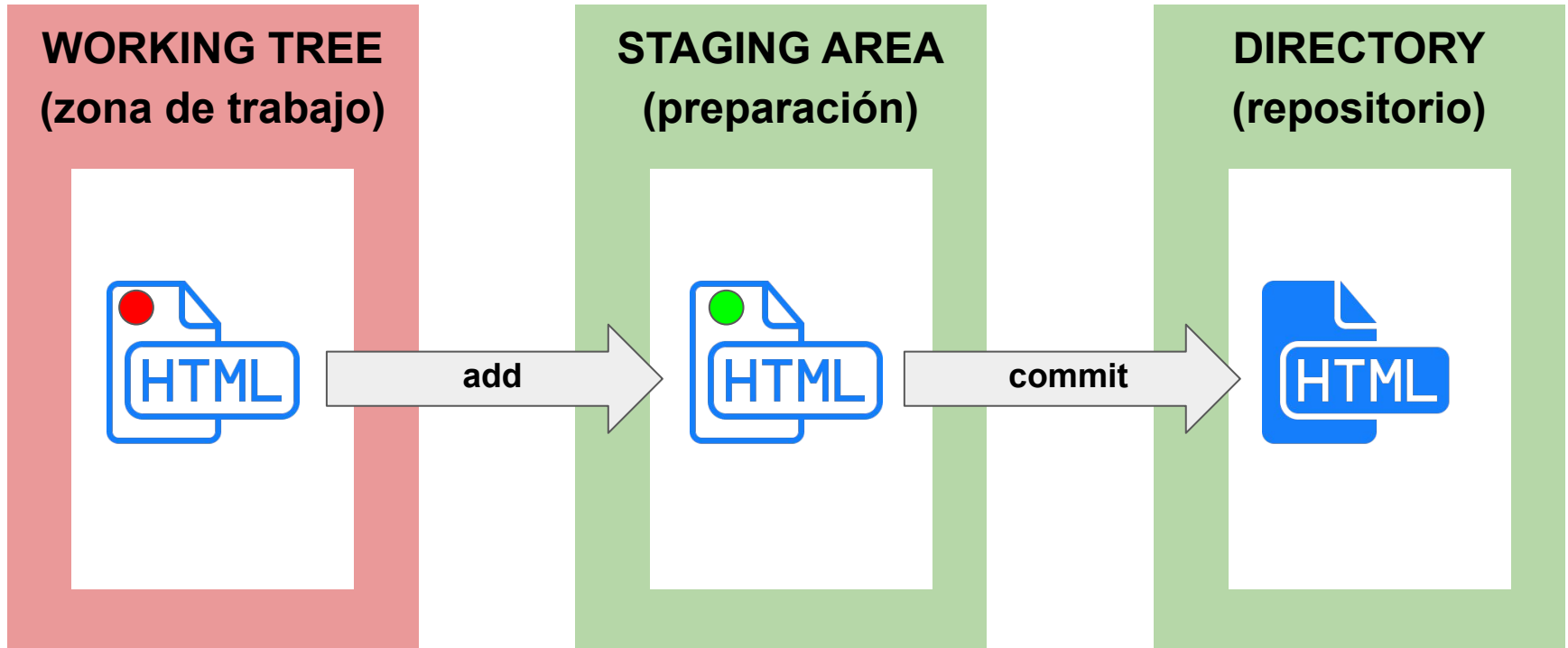
```
$ git init
```

- Opción 2: clonar un proyecto externo

```
$ git clone url_del_proyecto nombre_dir
```



COMANDOS BÁSICOS



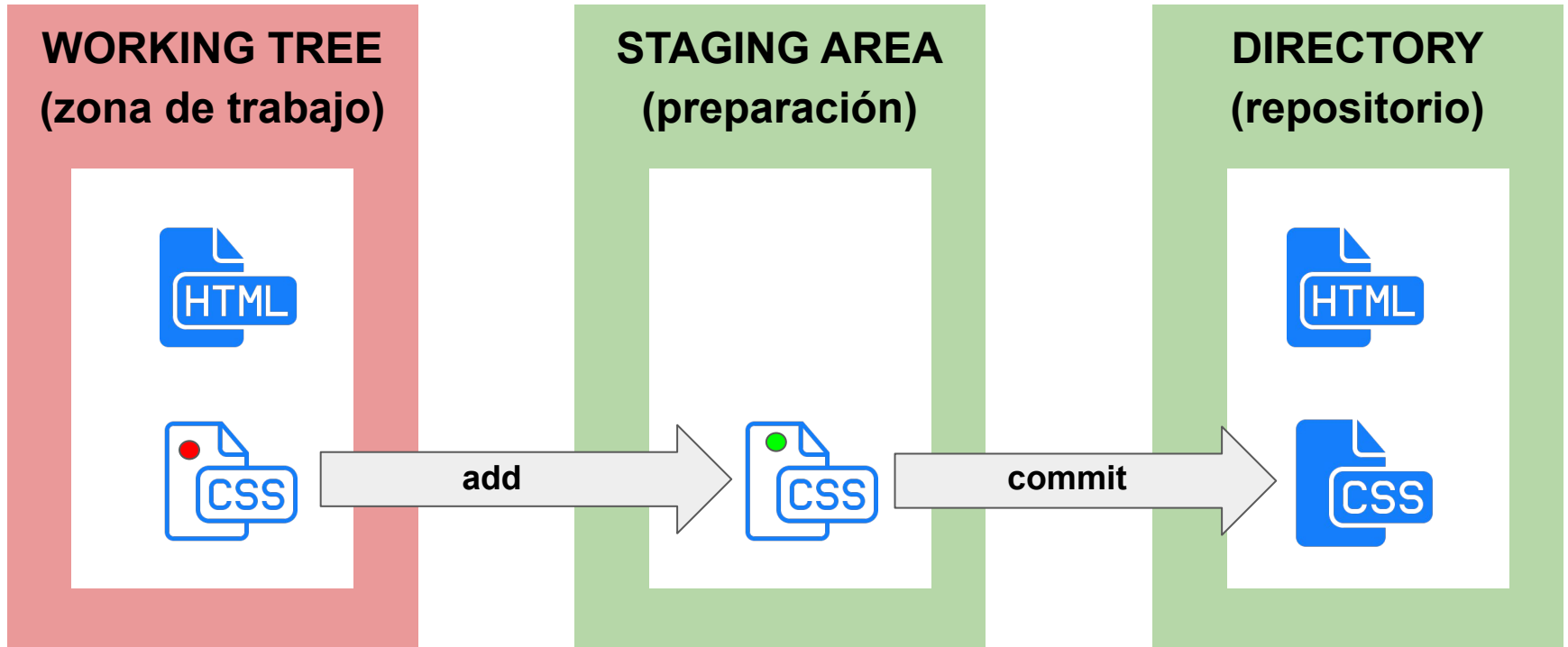
git add

git commit

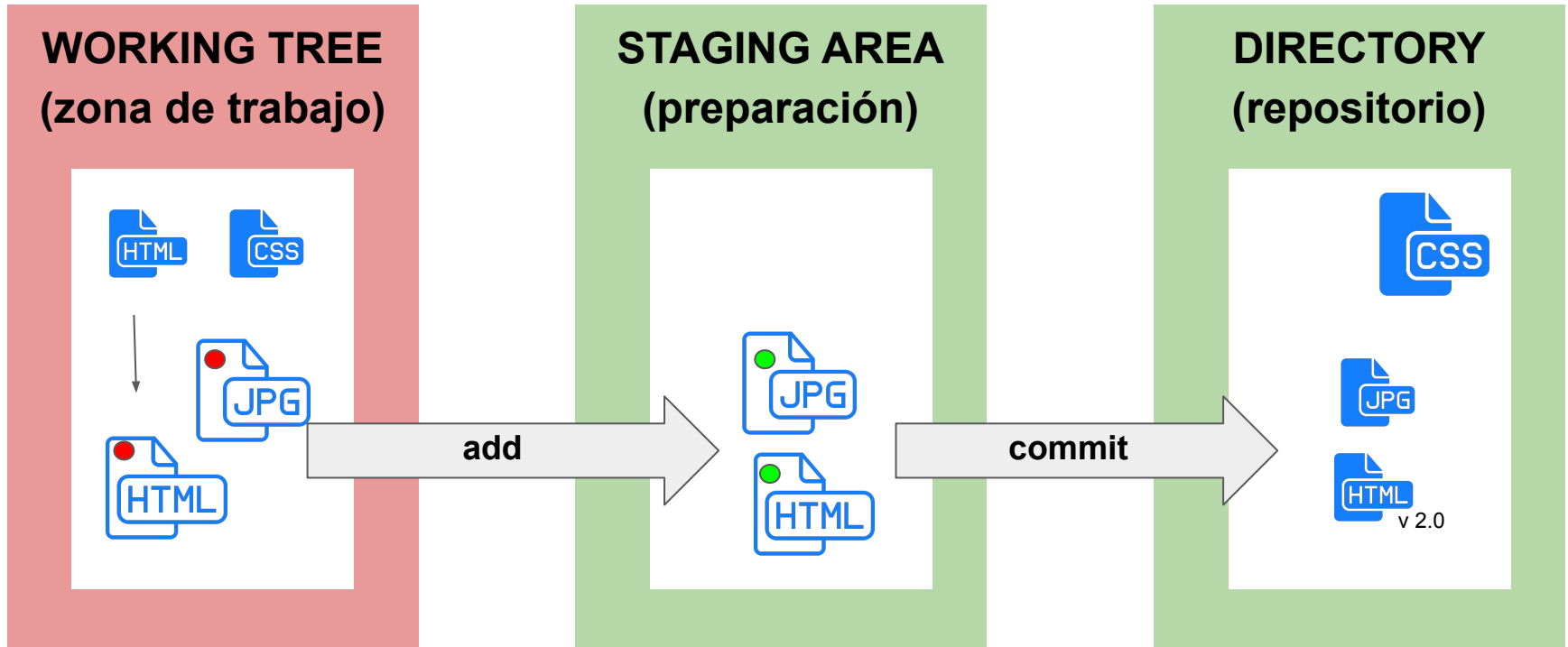
git add index.html

git commit -m "Creo el index"

COMANDOS BÁSICOS



COMANDOS BÁSICOS



COMANDOS BÁSICOS

git status

- muestra el estado de nuestros ficheros -



Working tree (zona de trabajo)

Tracked files (ficheros seguidos): estaban en la última fotografía, Git sabe que existen.

- Modificado
- Sin modificar

Untracked: el resto del working directory, que no estaba en la última foto ni está en el staging area.



Staging area (zona de preparación)

Staged (marcado para guardar)



Directory

Committed (guardado)

COMANDOS BÁSICOS

git status

```
nothing to commit, working directory clean
```

```
Untracked files:
```

```
(use "git add <file>..." to include in what will be committed)
```

```
README
```

```
nothing added to commit but untracked files present (use "git add" to track)
```

```
$ git add README
```

```
Changes to be committed:
```

```
(use "git reset HEAD <file>..." to unstage)
```

```
new file:   README
```

```
Changes not staged for commit:
```

```
(use "git add <file>..." to update what will be committed)
```

```
(use "git checkout -- <file>..." to discard changes in working directory)
```

```
modified:   CONTRIBUTING.md
```

```
Changes to be committed:
```

```
(use "git reset HEAD <file>..." to unstage)
```

```
new file:   README
```

```
modified:   CONTRIBUTING.md
```

COMANDOS BÁSICOS

git status

```
isabel@serenity:~/git/prueba$ git init
Inicializado repositorio Git vacío en /home/isabel/git/prueba/.git/
isabel@serenity:~/git/prueba$ git status
En la rama master

No hay commits todavía

no hay nada para confirmar (crea/copia archivos y usa "git add" para hacerles seguimiento)
isabel@serenity:~/git/prueba$ touch textol.txt
isabel@serenity:~/git/prueba$ git status
En la rama master

No hay commits todavía

Archivos sin seguimiento:
  (usa "git add <archivo>..." para incluirlo a lo que se será confirmado)

    textol.txt

no hay nada agregado al commit pero hay archivos sin seguimiento presentes
(usa "git add" para hacerles seguimiento)
```

COMANDOS BÁSICOS

git status

```
isabel@serenity:~/git/prueba$ git add textol.txt
isabel@serenity:~/git/prueba$ git status
En la rama master

No hay commits todavía

Cambios a ser confirmados:
  (usa "git rm --cached <archivo>..." para sacar del área de stage)

    nuevo archivo:  textol.txt

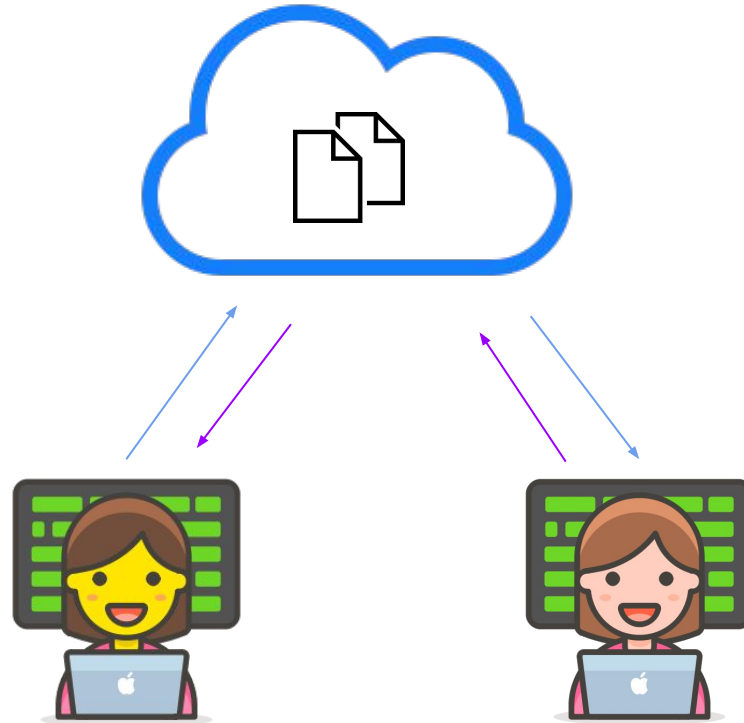
isabel@serenity:~/git/prueba$ git commit -m "prueba v1"
[master (commit-raíz) 600b552] prueba v1
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 textol.txt
isabel@serenity:~/git/prueba$ git status
En la rama master
nada para hacer commit, el árbol de trabajo esta limpio
```

GIT EN LA NUBE



Github
Gitlab

GIT EN LA NUBE



[Pull requests](#)[Issues](#)[Marketplace](#)[Explore](#)[DjangoGirls](#) / [tutorial](#)[Watch](#)

78

[★ Unstar](#)

1.2k

[Fork](#)

1.3k

[Code](#)[Issues](#) 68[Pull requests](#) 12[Actions](#)[Projects](#) 0[Wiki](#)[Security](#)[Insights](#)

✓ Clarify that code has to be added in the "Create a template for the p...

[Browse files](#)

...ost details" section.

master (#1606)

deeptikorwar authored and **das-g** committed 2 days ago

1 parent [17b5f86](#)

commit [92740b7e358c73e8e18b0fd23ccdde81d6be4408](#)

Showing **1 changed file** with **1 addition** and **1 deletion**.

[Unified](#)[Split](#)

2 en/extend_your_application/README.md

[Code](#) ...

```
@@ -134,7 +134,7 @@ Oh no! Another error! But we already know how to deal with it, right? We need to
134 134
135 135     We will create a file in `blog/templates/blog` called `post_detail.html`, and open it in the code editor.
136 136
137 137     - It will look like this:
137 137     + Let's type the following code:
138 138
139 139     {% filename %}blog/templates/blog/post_detail.html{% endfilename %}
140 140     {{>html
```


OTROS COMANDOS

git log

git remote add origin

git push

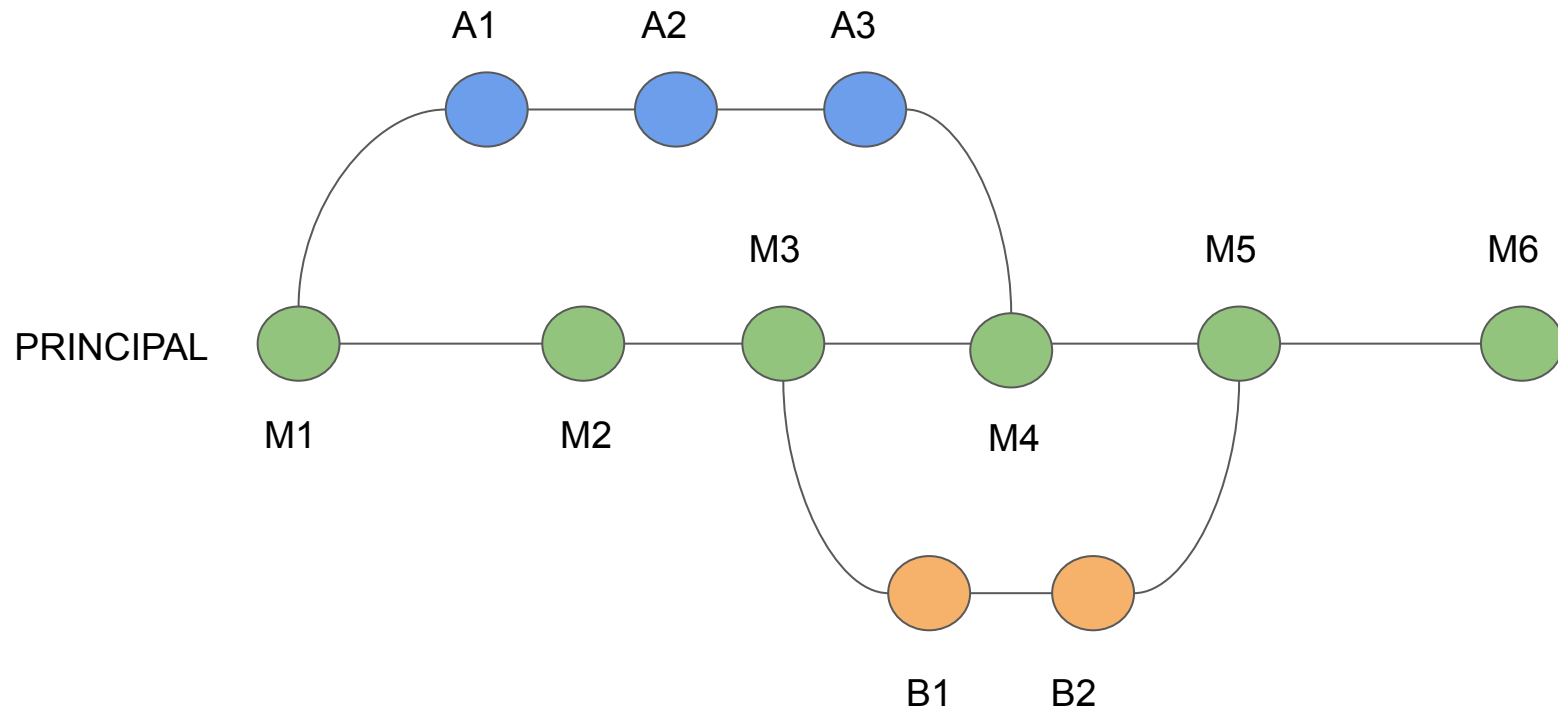
git pull

git branch

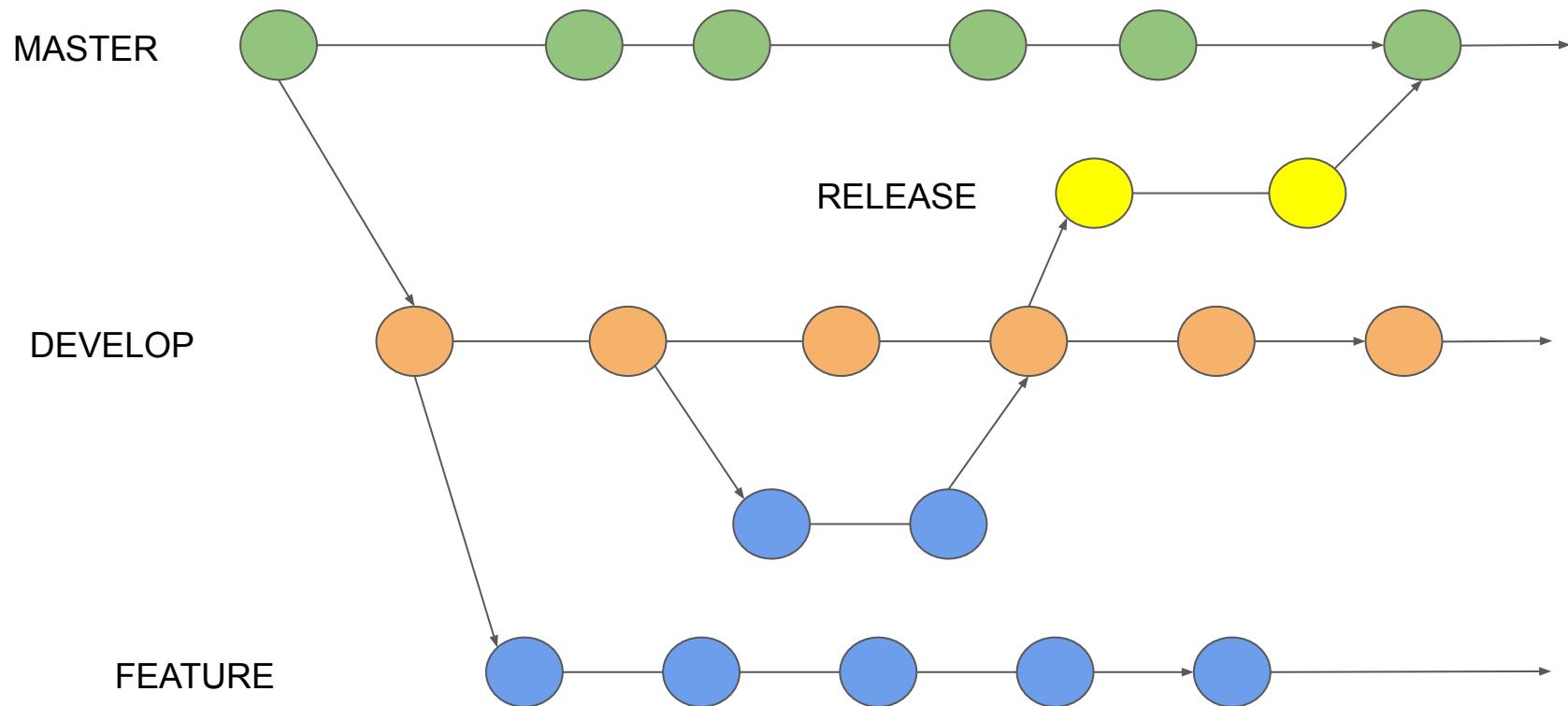
git checkout

git merge

RAMAS



RAMAS



RAMAS

Crear nueva rama e ir a ella: **git checkout -b** mi_rama

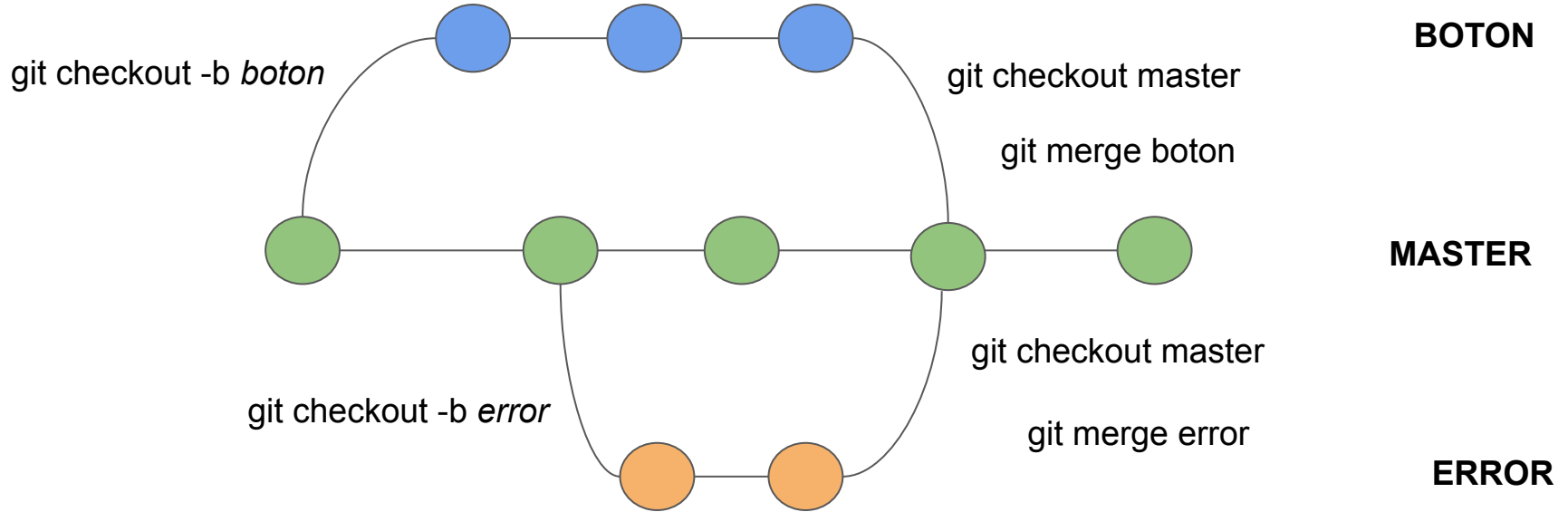
Volver a la rama principal: **git checkout** master

Subir rama al remoto: **git push** origin mi_rama

Traer los cambios a master: **git merge** mi_rama

Comprobar las rama que tenemos: **git branch**

RAMAS

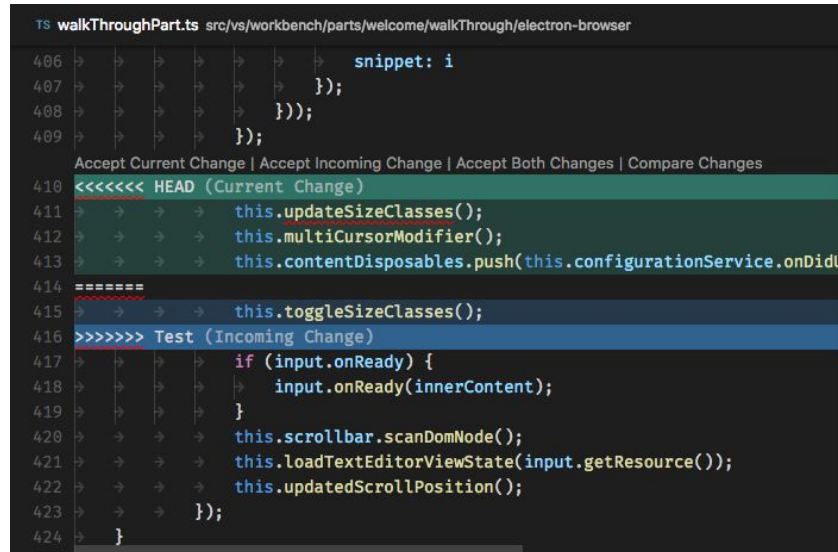


GIT MERGE



Resolución de conflictos

- 1) Git nos avisa de que hay conflictos en ciertos ficheros.
- 2) Tenemos que entrar en cada fichero y resolver los conflictos.
- 3) Finalmente, hay que hacer un commit.



The screenshot shows a code editor with a file named `walkThroughPart.ts` located at `src/vs/workbench/parts/welcome/walkThrough/electron-browser`. The code is in TypeScript and shows a conflict resolution process. The editor has a dark theme. The conflict is between the current HEAD (Current Change) and a Test (Incoming Change). The current HEAD change is highlighted in green, and the incoming change is highlighted in blue. The conflict is resolved by accepting the incoming change, as indicated by the blue highlight on line 416. The code includes a `snippet: i` block and a `toggleSizeClasses()` method. The conflict is resolved by accepting the incoming change, as indicated by the blue highlight on line 416.

```
TS walkThroughPart.ts src/vs/workbench/parts/welcome/walkThrough/electron-browser
406  → → → → → → → snippet: i
407  → → → → → → → });
408  → → → → → → → });
409  → → → → → → → });
Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes
410 <<<<<< HEAD (Current Change)
411  → → → → → → → this.updateSizeClasses();
412  → → → → → → → this.multiCursorModifier();
413  → → → → → → → this.contentDisposables.push(this.configurationService.onDidU
414 =====
415  → → → → → → → this.toggleSizeClasses();
416 >>>>>> Test (Incoming Change)
417  → → → → → → → if (input.onReady) {
418  → → → → → → →   input.onReady(innerContent);
419  → → → → → → → }
420  → → → → → → → this.scrollbar.scanDomNode();
421  → → → → → → → this.loadTextEditorViewState(input.getResource());
422  → → → → → → → this.updatedScrollPosition();
423  → → → → → → → });
424  → → → → → → → }
```

¡YA CONOCES GIT!

VCS

línea de comandos

git add + git commit + git push

git status, git log

git branch / checkout / merge

Github / Gitlab



¿PREGUNTAS?



CRÉDITOS



[Sagar Unagar](#)



[Webalys](#)



[Yannick Lung](#)



[Flatart](#)



[Github](#)



[Visual Pharm](#)



[Candy Design](#)



[Ivan Kurnianto](#)

ENLACES

- Sitio oficial: <http://git-scm.com/>
- Tutorial sencillo: <https://rogerdudler.github.io/git-guide/index.es.html>

