



# Workflow con Git y Source Tree



Sobre atSistemas



 **Confluence**

 **JIRA**



Atlassian Git Party





Source Tree





## Puntos a tener en cuenta

- Solo para Mac (de momento)
- No permite alguna funcionalidad avanzadas:
  - Squasing
  - Rebase interactivos



---

# TRABAJANDO CON GIT



## Trabajando con Git

Git es bueno trabajando con ramas...

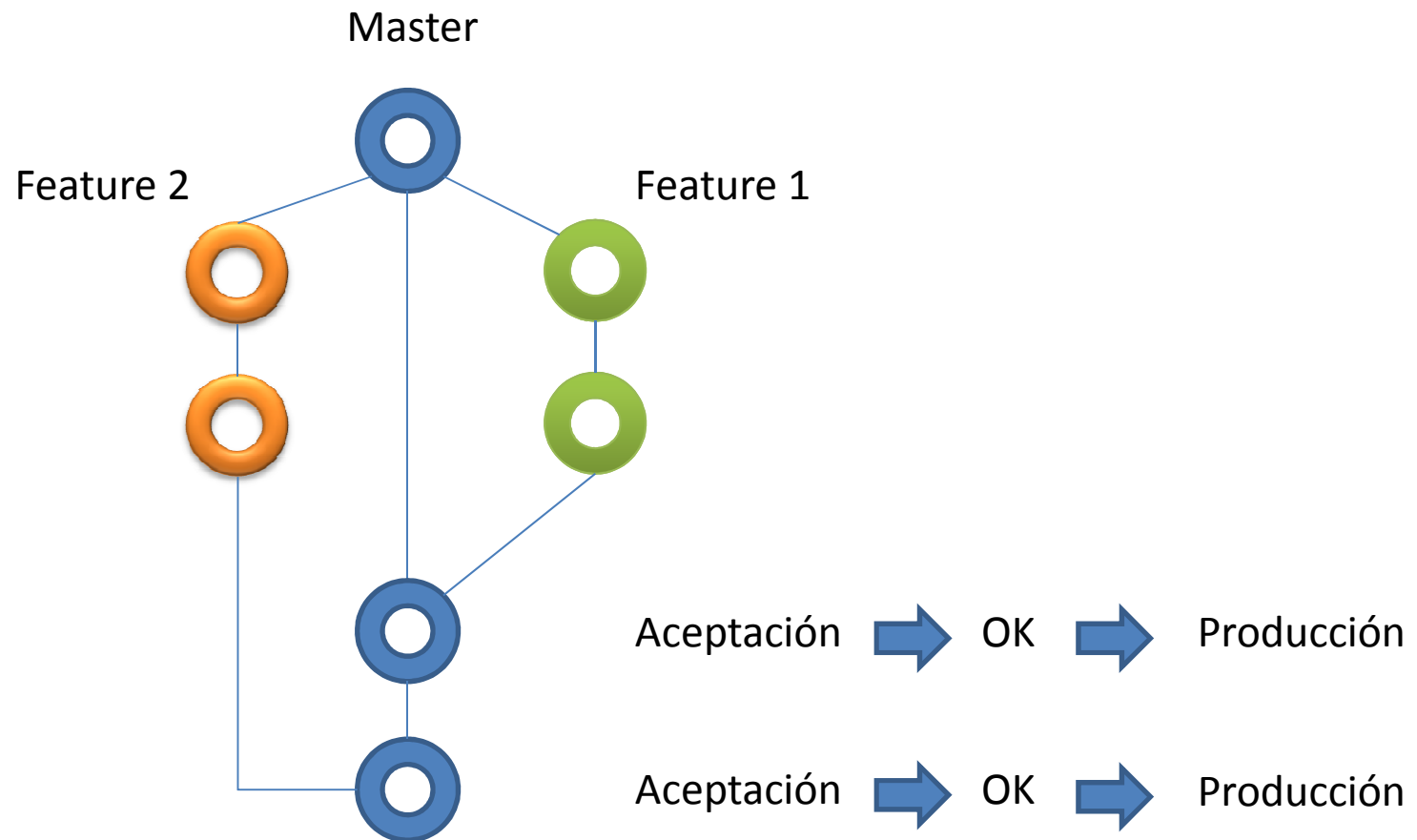


¡ Hagamos ramas para cada funcionalidad / tarea!



# Hagamos ramas

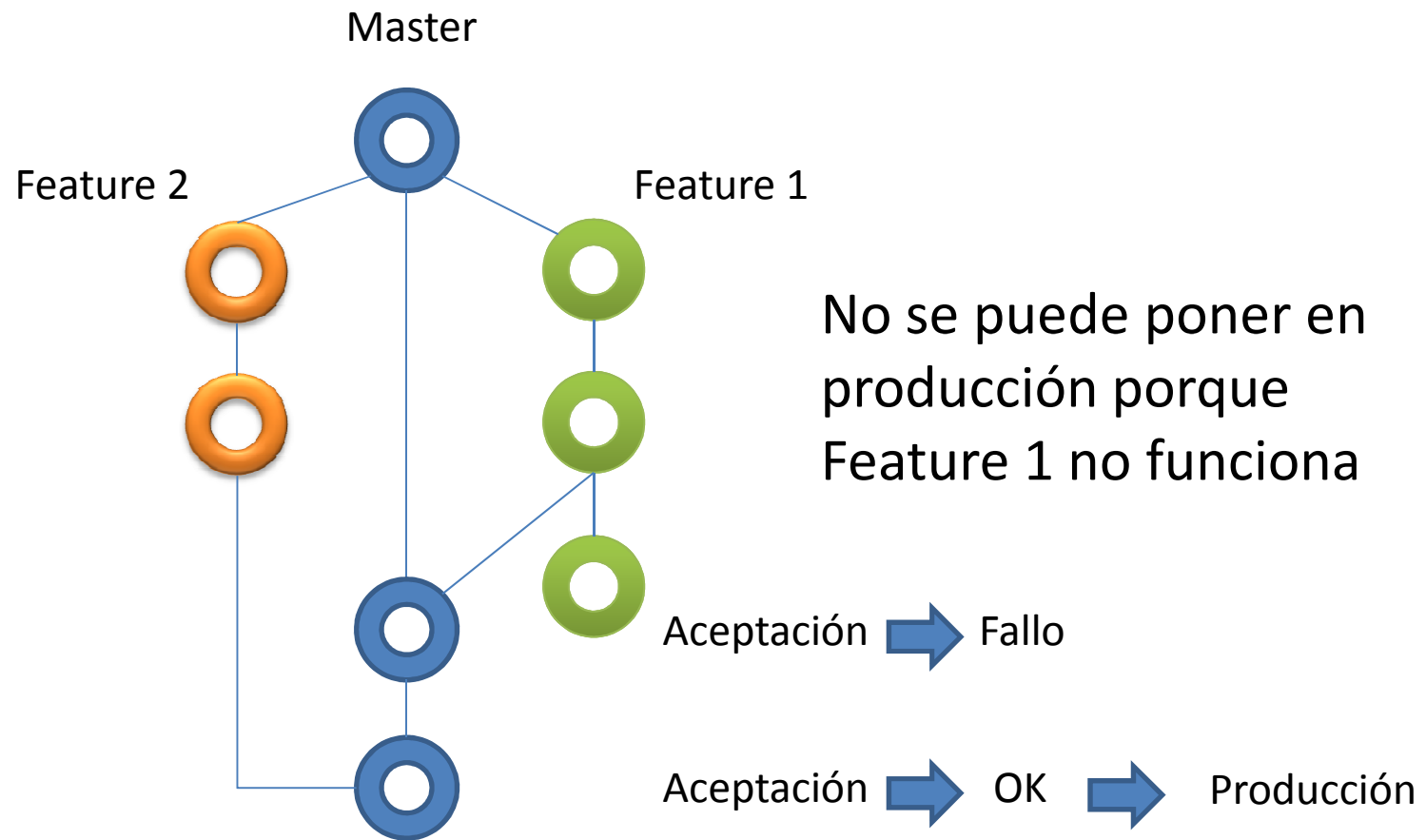
## Escenario A: Funcionalidades correctas





## Hagamos ramas

### Escenario B: Funcionalidad que falla







Hagamos ramas

No basta solo con tener ramas por funcionalidad...

Necesitamos un flujo de trabajo que nos permita trabajar con los diferentes estados de mi desarrollo...

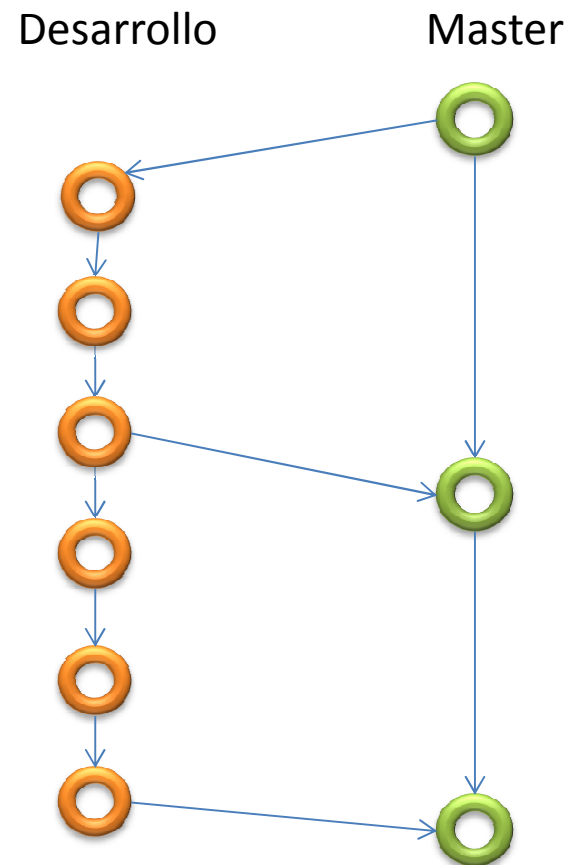
---

# WORKFLOW DE GIT



## Workflow

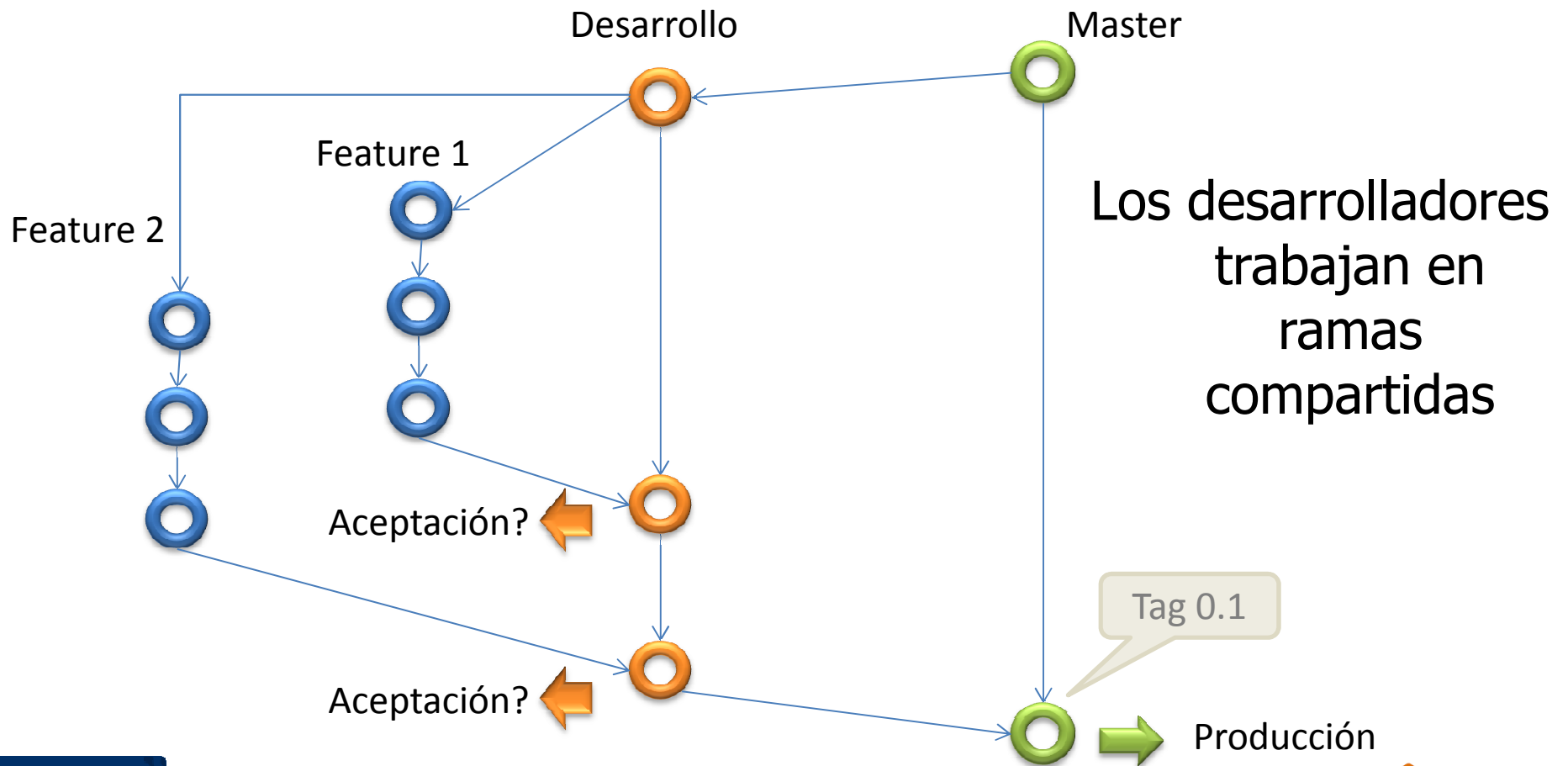
### 1) Dos ramas principales: Desarrollo y Master





## Workflow: Opciones

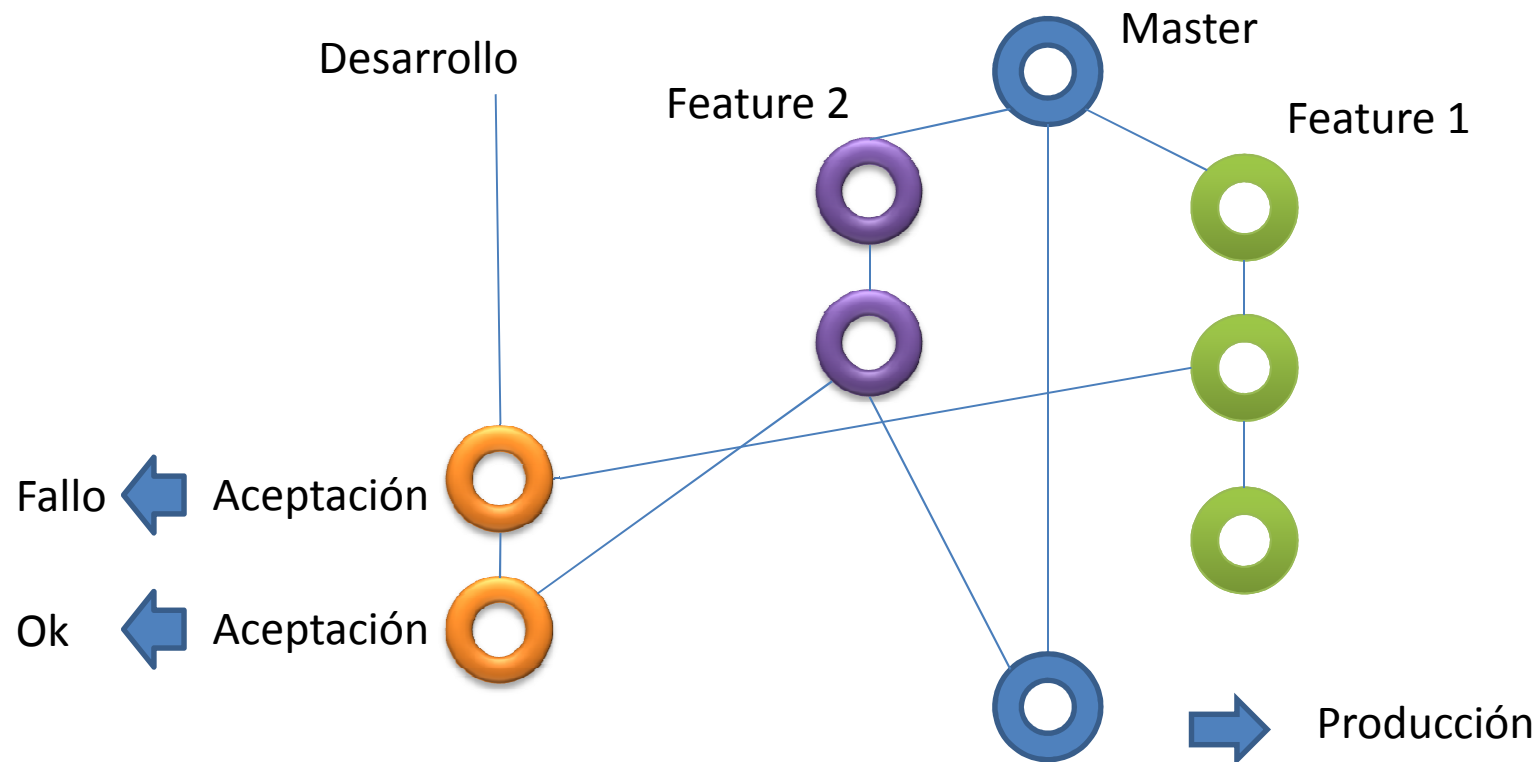
### 2) Una rama por cada funcionalidad / tarea (local)





Hagamos ramas

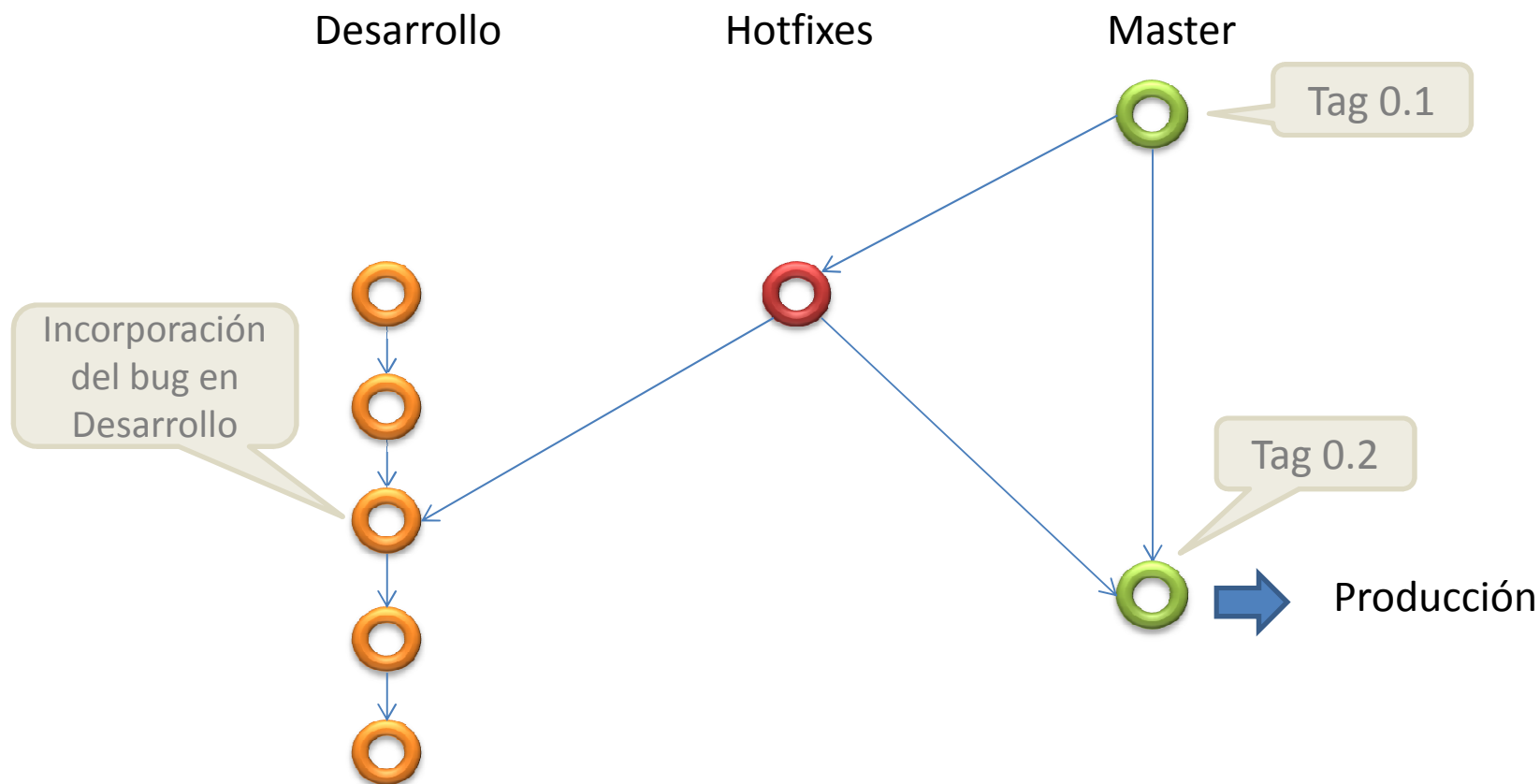
## 2) Una rama por cada funcionalidad / tarea (local)





## Workflow

### 3) Cada bug se resuelve en una rama independiente





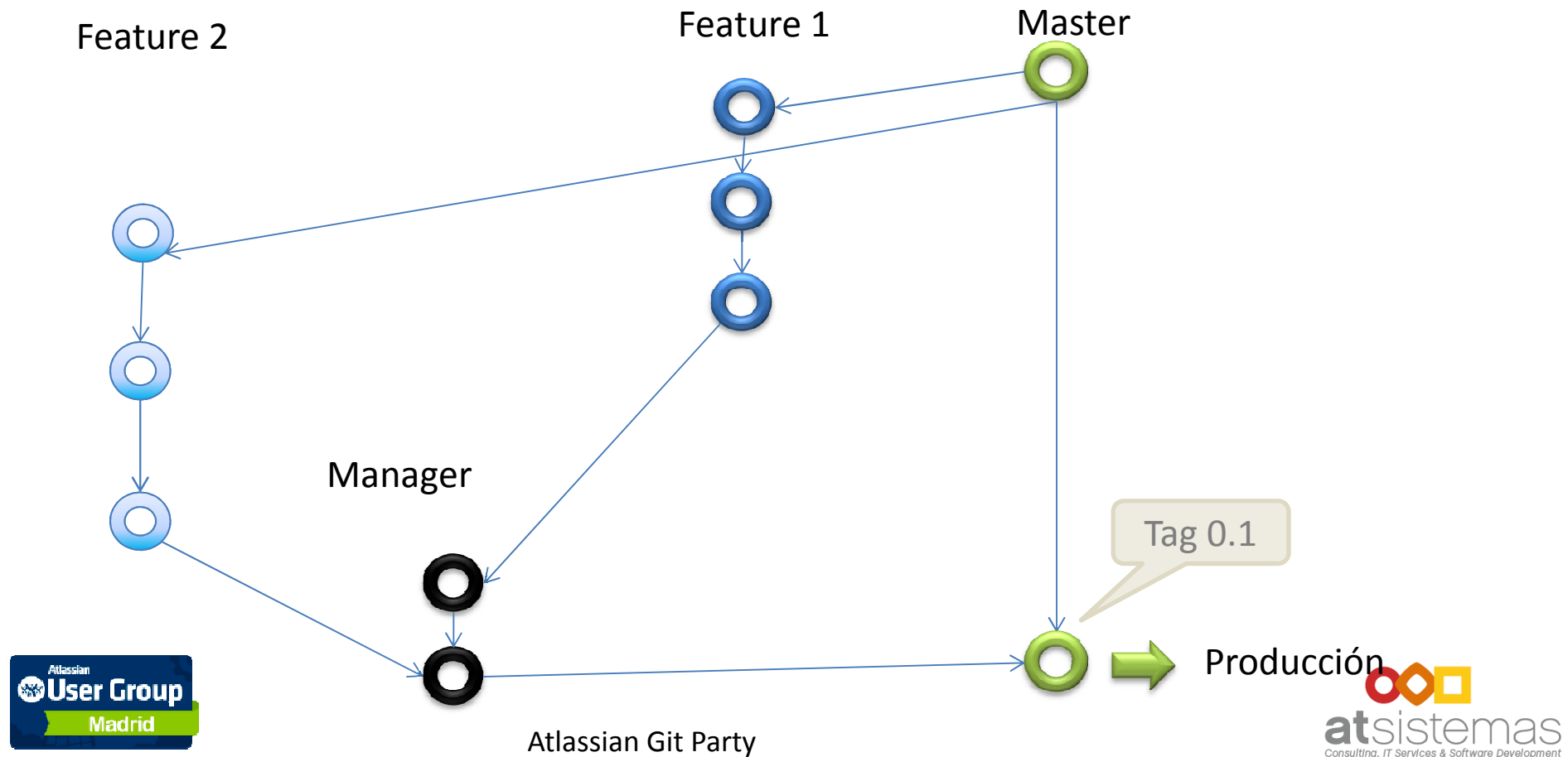
Source Tree

# ¡Manos a la obra!





## Rol responsable de las subidas a master







## Flujo Manager

1. El manager simula la rama de desarrollo.
2. Cada miembro del equipo hace de manager.
3. Solo el manager hace merge a master

---

# RESUMEN



Recuerda

- SIEMPRE crea ramas desde master
- Funcionalidad terminada: Merge a *desarrollo*
- Funcionalidad aceptada: Merge a *master* o *release*
- Nunca merge o rebase *master* con tus ramas.



Puedes encontrar esta presentación en:

<http://goo.gl/5oTUA>



## Recursos

- A successful Git branching model:

<http://nvie.com/posts/a-successful-git-branching-model/>

- Git flow:

<https://github.com/nvie/gitflow>

- Source Tree:

<http://www.sourcetreeapp.com/>



# Preguntas

Israel Alcázar  
ialcazar@atsistemas.com  
@ialcazar



Atlassian Git Party

