

Control de Versiones con



Carlos Monterrosa



@monterrosa_c

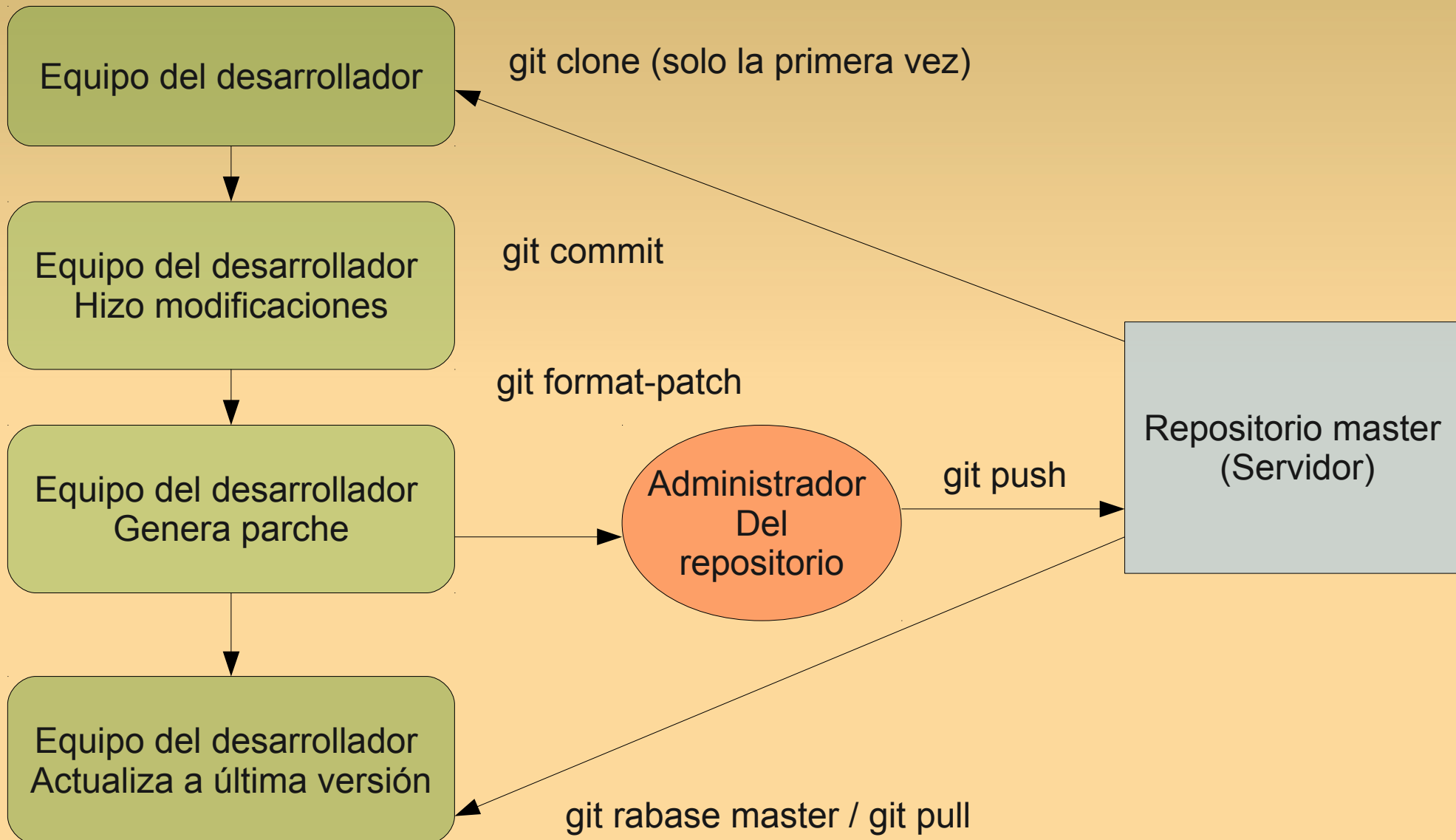
¿Por qué git?

“Git es un sistema de control de versiones escrito por Linus Torvalds (originalmente) y diseñado para manejar proyectos de software con velocidad y eficiencia; es especialmente popular en la comunidad open source, sirviendo como plataforma de desarrollo para proyectos como el Kernel Linux, Ruby on Rails, WINE o X.org, Perl, Gnome, QT, Android, y es utilizado por compañías como: Google, Facebook, LinkedIn, Twitter, entre otras.”

Ventajas

- Tener un control exacto sobre cual es la última versión del código, y quién y cuando la ha cargado.
- Poder comparar versiones, viendo cuales han sido los cambios realizados.
- Regresar atrás (a una versión anterior) cuando lo que hemos desarrollado no nos ha dado los resultados esperados.
- Crear distintas ramas del proyecto. Si llegado a un punto se hace necesario hacer dos aplicaciones con distintas funcionalidades, pero con cosas en común, se pueden separar en dos ramas.

Flujo



Flujo

1. Clonamos repositorio:

```
git clone git.poderjudicialchiapas.gob.mx:/monitorrh
```

2.- Modificamos los archivos que sean necesarios

3.- Actualizamos repositorio

```
git pull origin master
```

4.- Si hay son archivos nuevos, los agregamos al repositorio:

```
git add archivos_nuevos
```

5.- Generamos commit y describimos cambio:

```
git commit -a (Sube todos los cambios)  
git commit archivos_modificados
```

6.- Generamos parche.

```
git format-patch origin master
```

7.- Subimos parche al sistema SIAC como un servicio, adjuntamos el archivo, esto le notificará al administrador del repositorio.

8.- Diariamente el siac actualizará el repositorio y veremos los cambios realizados en el administrador de código fuente.

Comandos básicos

`git init`

Crea un repositorio en el directorio actual

`git clone url`

Clona un repositorio remoto dentro de un directorio

b) Operaciones sobre Archivos:

`git add path`

Adiciona un archivo o un directorio de manera recursiva

`git rm ruta`

Remueve un archivo o directorio del árbol de trabajo

-f : Fuerza la eliminación de un archivo del repositorio

Comandos básicos

`git mv archivo`

Mover un archivo de ruta

`git rebase master`

Actualizar nuestro repositorio local con la última versión del master y mantiene los cambios

`git pull origin master`

Actualiza nuestro repositorio local (conveniente cuando ya hemos empaquetado los cambios)

Comandos básicos

`git checkout [rev] archivo`

Recupera un archivo desde la rama o
revisión actual

-f : Sobre-escribe los cambios locales no
guardados

`git status`

Imprime un reporte del estado actual del
árbol de trabajo local

`git diff [ruta]`

Muestra la diferencia entre los cambios en el
árbol de trabajo local

Donde seguir

- **Git basics**

<http://cworth.org/hgbook-git/tour/>

- **Git inmersión:**

<http://gitimmersion.com/>

- **Tutorial de git en español:**

<http://elweb.co/uncategorized/git-tutorial-1-en-espanol/>

- **Git desde cero:**

<http://xenodesystems.blogspot.mx/2012/05/tutorial-git-desde-cero.html>