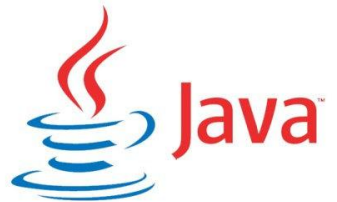


Paradigmas y Lenguajes III

UNIDAD II - INTRODUCCIÓN A GIT



Terminología

Repositorio: El repositorio es el lugar en el que se almacenan los datos actualizados e históricos de cambios

Revisión: Una revisión es una versión determinada de la información que se gestiona. Hay sistemas que identifican las revisiones con un contador (Ej. subversion). Hay otros sistemas que identifican las revisiones mediante un código de detección de modificaciones (Ej. git usa SHA1).

Terminología

Commit: Confirmar es escribir o mezclar los cambios realizados en la copia de trabajo del repositorio. Los términos 'commit' y 'checkin' .

Branch: Un conjunto de archivos puede ser ramificado o bifurcado en un punto en el tiempo de manera que, a partir de ese momento, dos copias de esos archivos se pueden desarrollar a velocidades diferentes o en formas diferentes de forma independiente el uno del otro.

Terminología

Change: un cambio (o diff, o delta) representa una modificación específica de un documento bajo el control de versiones.

Checkout: Es crear una copia de trabajo local desde el repositorio.

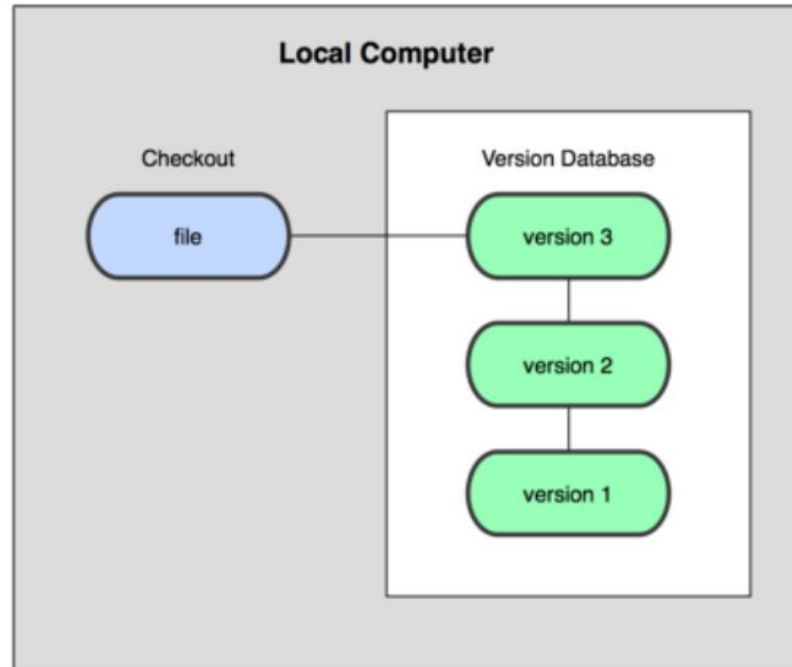
Merge: Una fusión o integración es una operación en la que se aplican dos tipos de cambios en un archivo o conjunto de archivos.

Clasificación de los Tipos de Repositorios

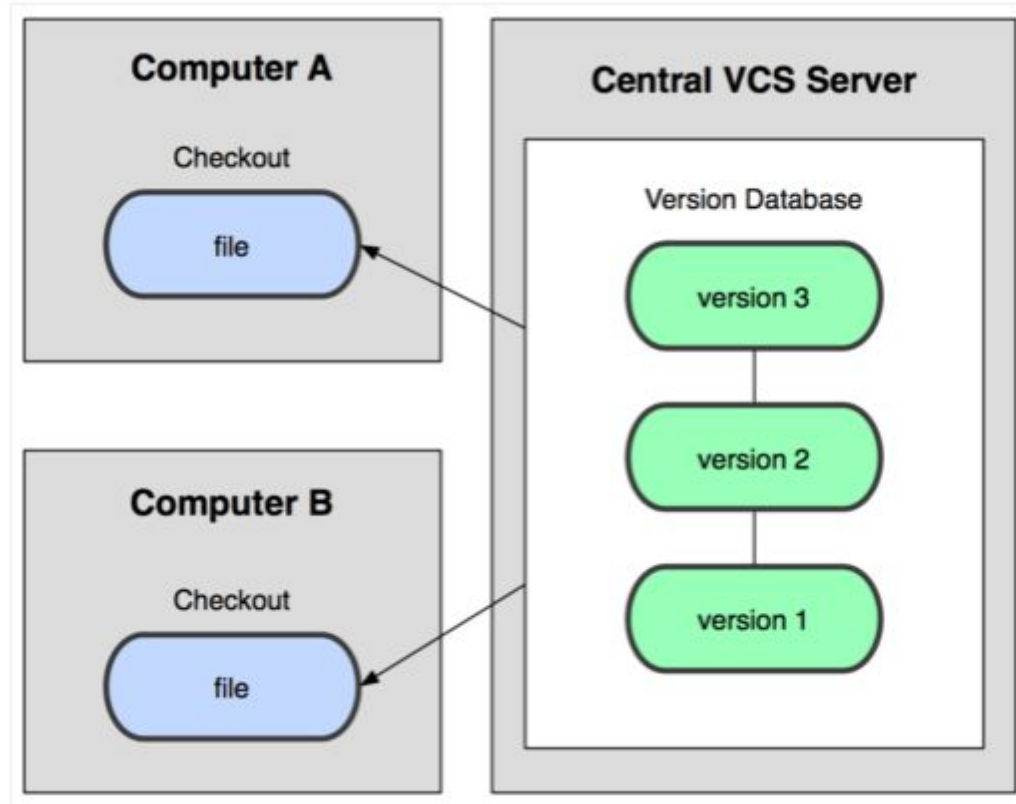
- Locales
- Centralizados
- Distribuidos

Locales

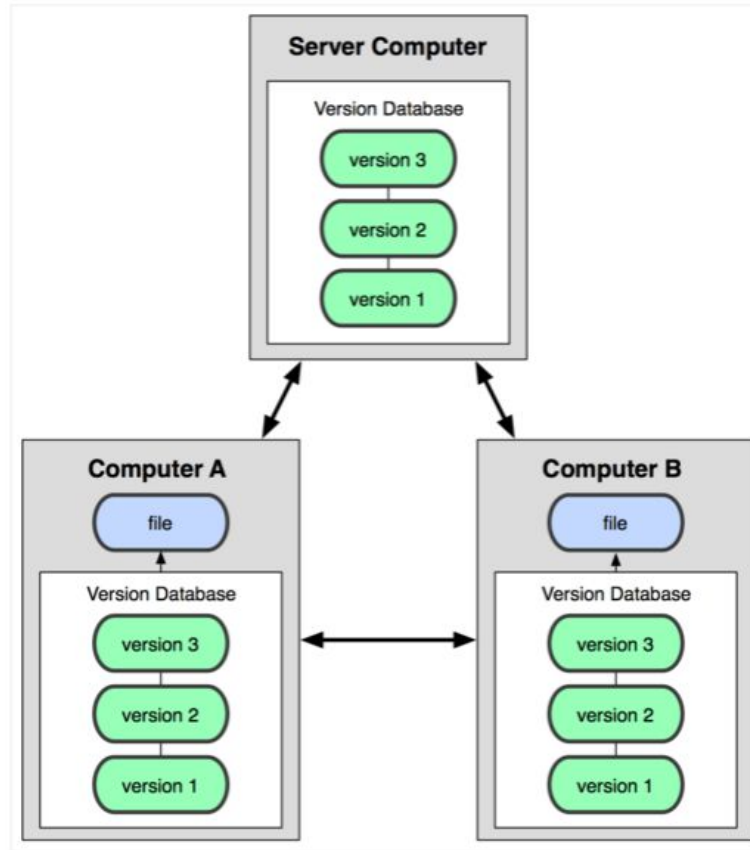
Los cambios son guardados localmente.



Centralizados



Distribuidos



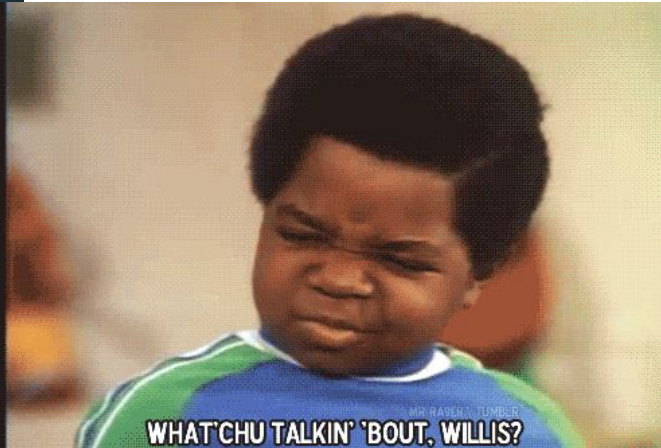
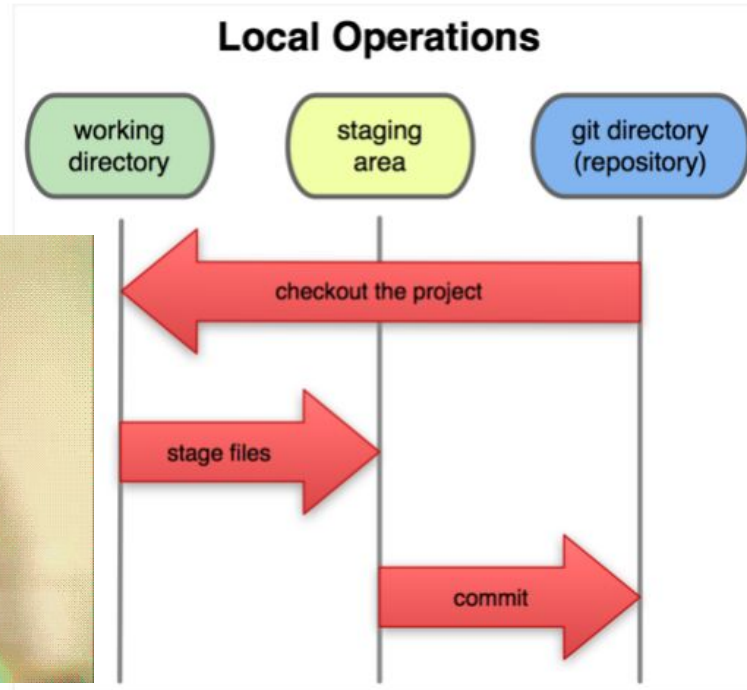
Funcionamiento de Git

Git tiene tres estados principales en los que se pueden encontrar los archivos en proyecto:

- committed
- modified
- staged

Funcionamiento de Git

Todo proyecto en Git tiene tres secciones:



Instalación de Git

Para descargar la última versión del cliente de Git ingresar a:

<https://git-scm.com/downloads>



Comenzando con Repositorios Git

#Ingresar al directorio de mi proyecto

```
$ cd /c/user/unidad2
```

#Creando un proyecto git

```
$ git init
```

Comenzando con Repositorios Git

#Creamos un archivo

```
$ echo "you should read me ">README
```

#ver el estado del repositorio

```
$git status
```

#podemos ver los cambios efectuados

```
$git add README
```

Comenzando con Repositorios Git

#Llegamos al commit

\$ git commit -m "my first commit"

#podemos ver el log del repositorio

y ver si quedo registrado nuestro commit

\$git log



Git - Configuración

```
$ git config --global user.name "paradigmasIII"
```

```
$ git config --global user.email abc@example.com
```

#Ayuda en consola

```
$ git help <verb>
```

Comenzando con Repositorios Git

#Ingresar al directorio de mi proyecto

\$ echo "unidad2" > README

#comprobar estado del repositorio

\$ git status

#agregar archivo

\$ git add README

Añadir un Repositorio Remoto

#para ver tus repositorios remotos

```
$ git remote
```

#Añadir Repositorio Remoto

```
#git remote add [nombre] [url]
```

```
$ git remote add unidad2
```

```
https://github.com/paradigmasylenguajes3/unidad2
```

Clonar un Repositorio Remoto

#Ahora podemos traer todos los archivos

\$ git clone

<https://github.com/paradigmasylenguajes3/unidad2.git>

Otras formas de obtener los archivos de un repositorio remoto

\$ git fetch

Git - Branchs

¿ Que es

- Git no
- (guar
- Git al
- (copia
- tal y c



ental

antáneas
npletos,
nento).

Git - Branchs

¿ Que es una Rama ?

- En cada confirmación de cambios (commit), Git almacena una instantánea de tu trabajo preparado.

Git - Branchs



- Vamos a la práctica
1. Supongamos que tenemos 3 archivos que preparar (stage)
 2. Luego confirmamos los archivos (commit)
 3. Git cuando prepara los archivos realiza una suma de control (SHA-1)

Git - Branchs

- Vamos a la práctica

4. Almacena una copia de los archivos en el repositorio, a las que denomina **BLOBS**

En código sería:

```
$git add README test.rb License
```

```
$git commit -m "initial commit"
```

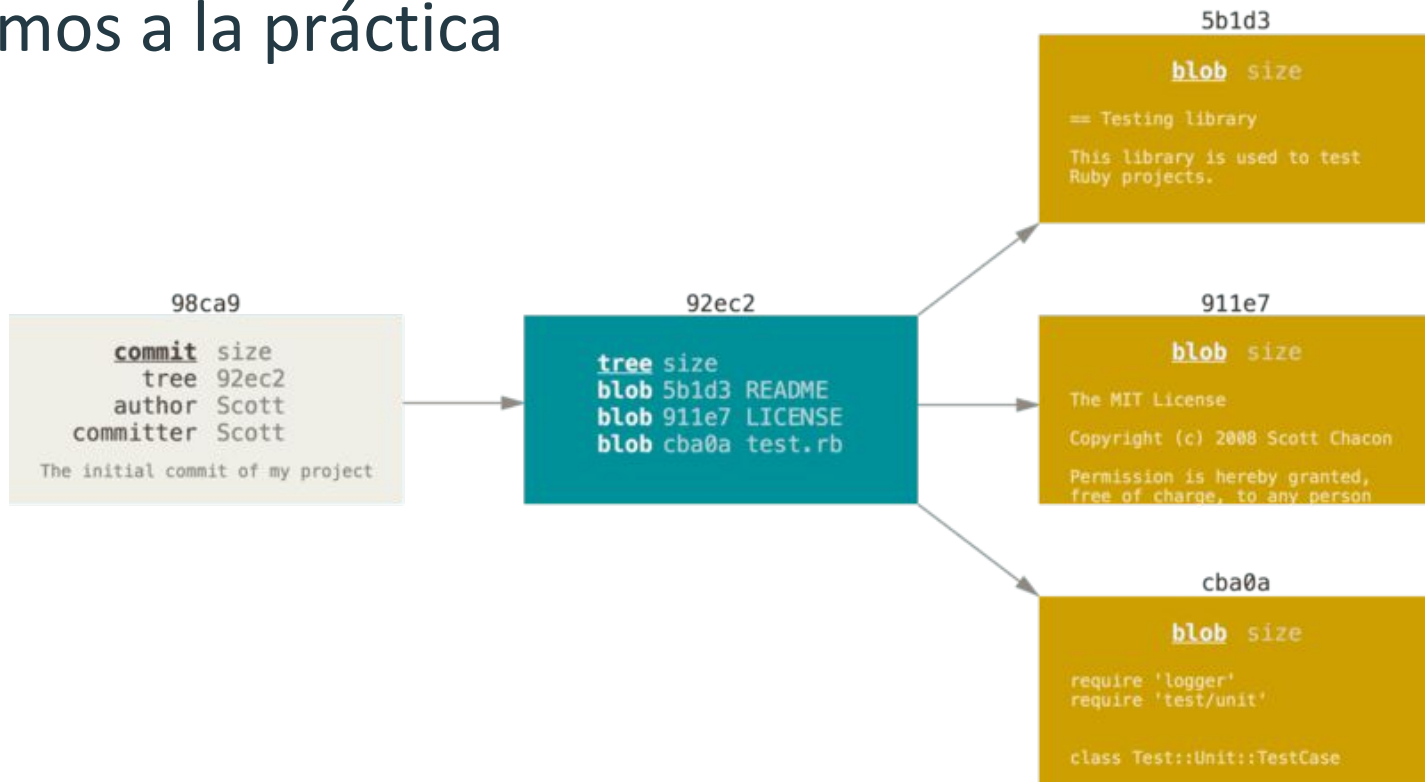
Git - Branchs

- Vamos a la práctica

5. Por último Git realiza las sumas de control y los guarda como objetos del árbol del repositorios

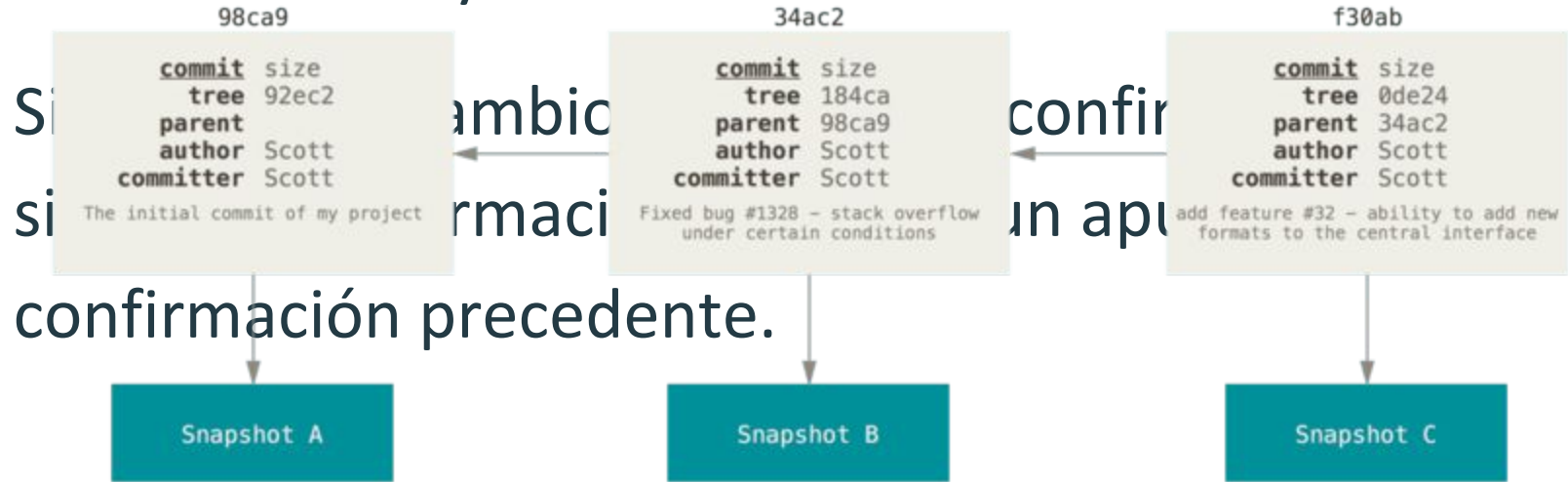
Git - Branchs

- Vamos a la práctica



Git - Branchs

- Un commit y sus árboles



Git - Branchs

→ Commits y sus predecesores

- ◆ Una rama Git es simplemente un apuntador móvil apuntando a una de esos commits

- ◆ La rama por defecto de Git es la rama **master**

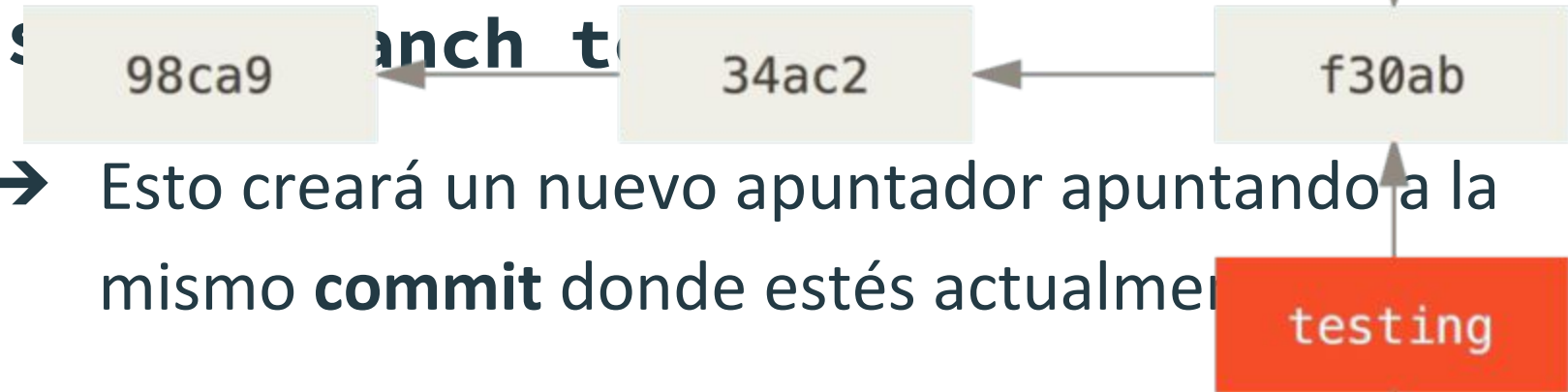
- ◆ Con el primer commit de cambios que realicemos, se creará esta rama principal



Git - Branchs

→ Crear una nueva Branch o Rama

◆ ¿Qué sucede cuando creas una nueva rama?



→ Esto creará un nuevo apuntador apuntando a la mismo **commit** donde estés actualmente