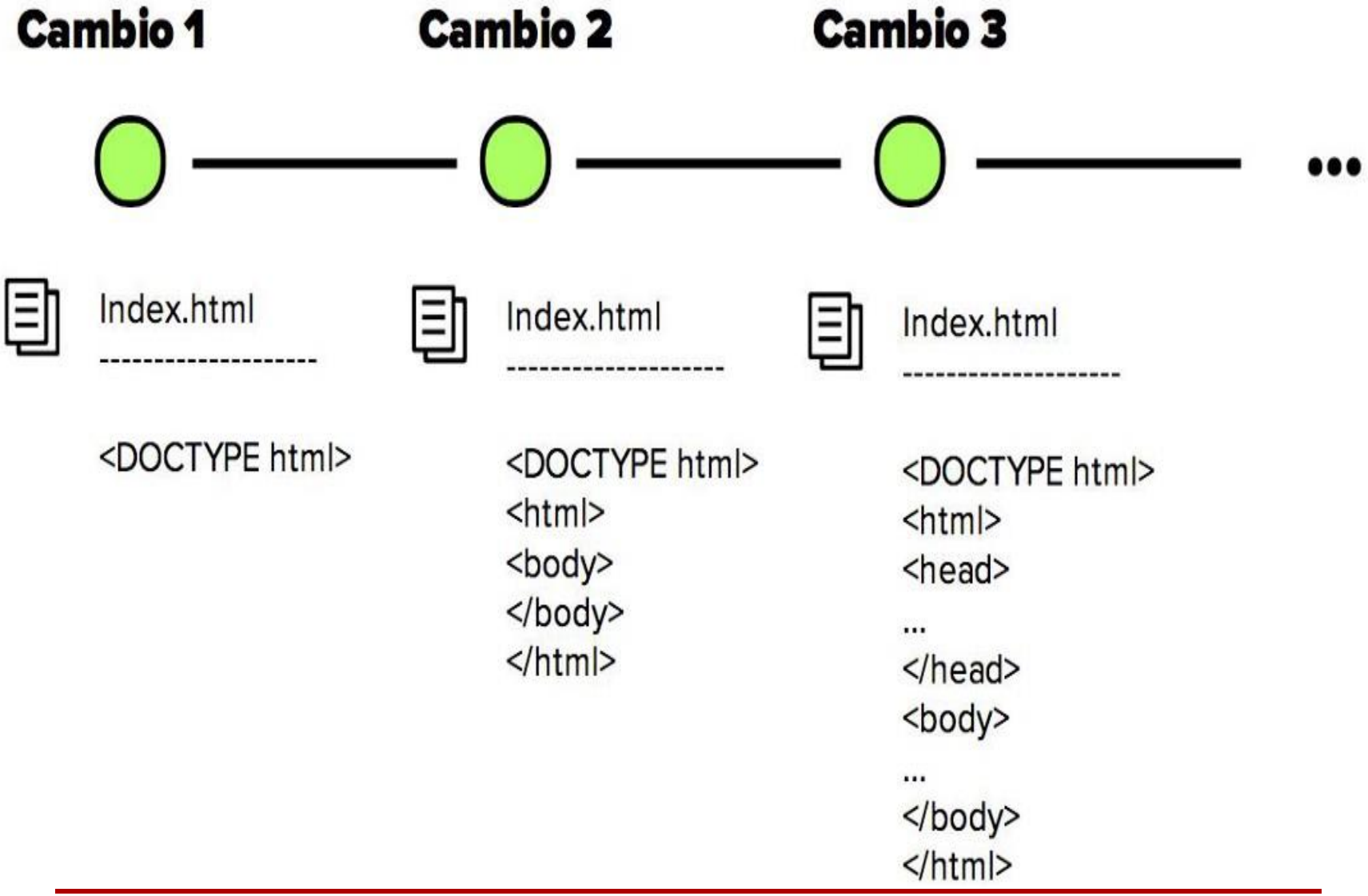
+

Presentado por:

Maryon Torres

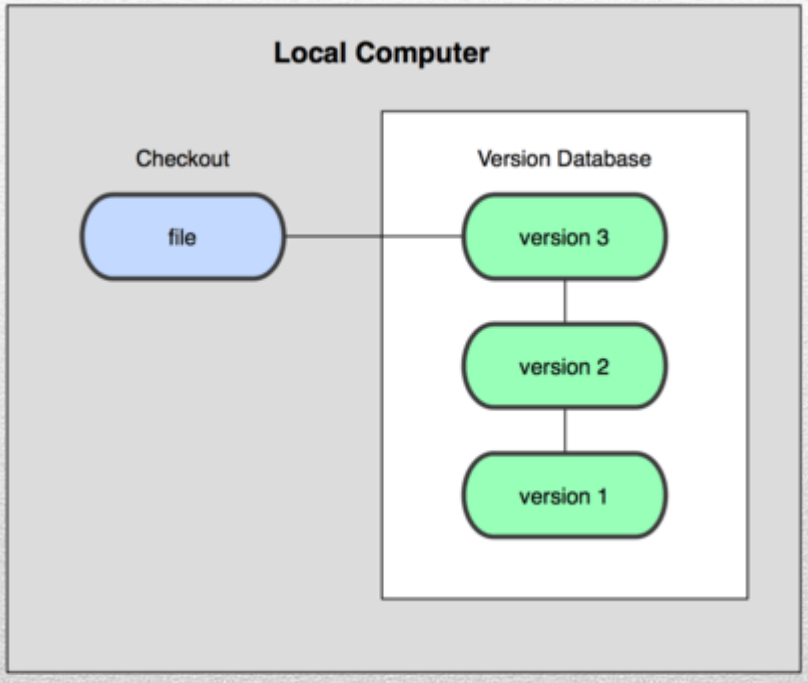
Acerca de Git:

- Sistema que registra los cambios realizados sobre un archivo o conjunto de archivos a lo largo del tiempo
- Es útil para cualquier profesión, no únicamente desarrolladores
- Permite revertir archivos específicos a su estado anterior, o bien todo el proyecto entero
- Compara cambios a lo largo del tiempo, permitiendo ver quién modificó por última vez un archivo


Cambio 1
Cambio 2
Cambio 3
Index.html
<DOCTYPE html>
Index.html
<DOCTYPE html>
<html>
<body>
</body>
</html>
Index.html
<DOCTYPE html>
<html>
<head>
...
</head>
<body>
...
</body>
</html>

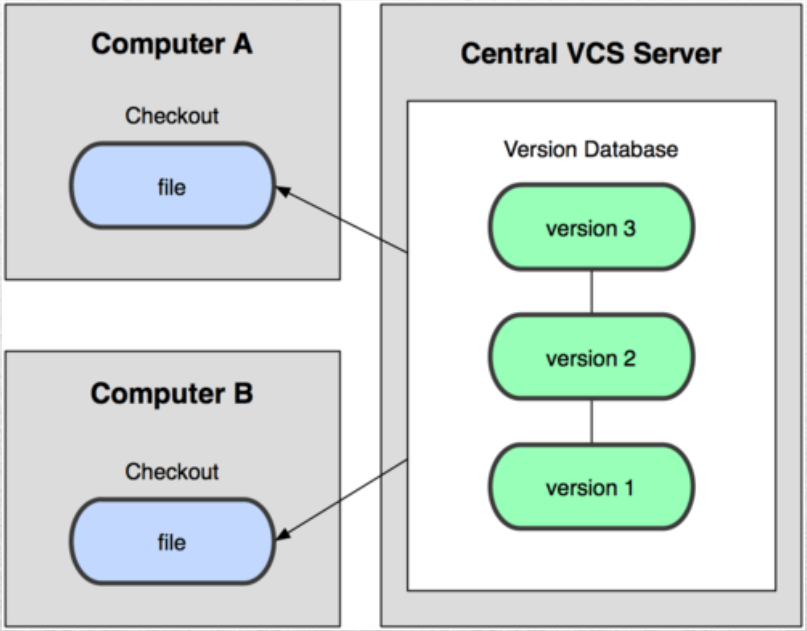
Sistemas de control de versiones local

- El método más primitivo, utilizado por mucha gente, es copiar los archivos a otro directorio
- Este enfoque es muy común por su simpleza, pero está propenso a muchos errores: perder la ubicación del directorio, sobrescribir archivos no deseados



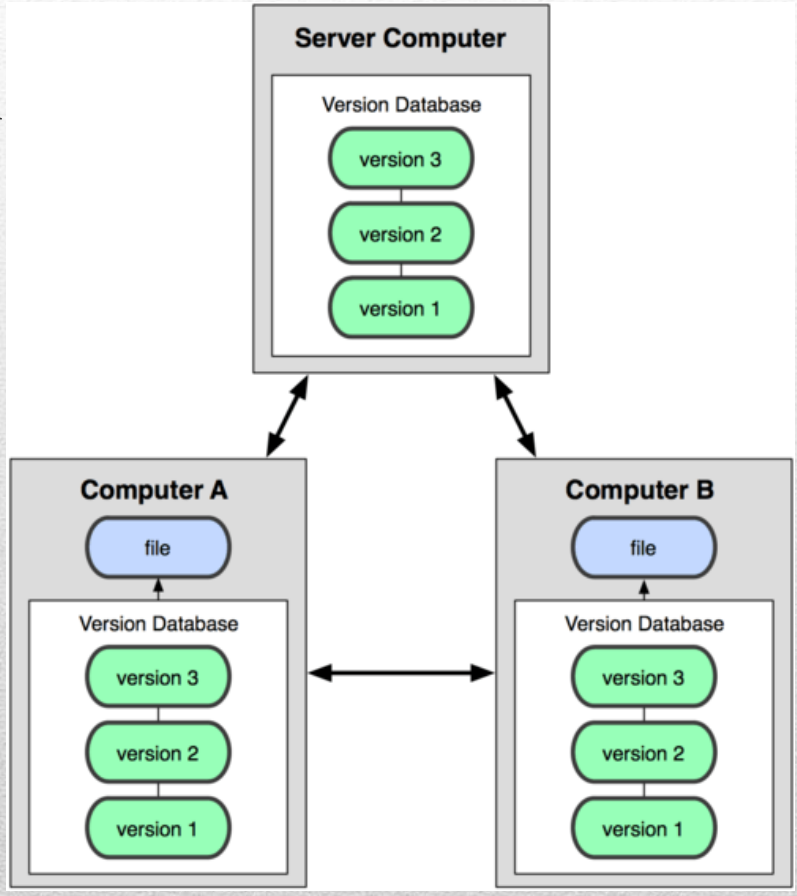
Sistemas de control de versiones centralizado

- Inicia por la necesidad de colaborar en otros sistemas
- El servidor contiene todos los archivos centralizados y los clientes descargan del servidor
- Durante muchos años éste ha sido el estándar para el control de versiones
- Unas de sus desventaja alta dependencia del servidor.



Sistemas de control de versiones distribuida

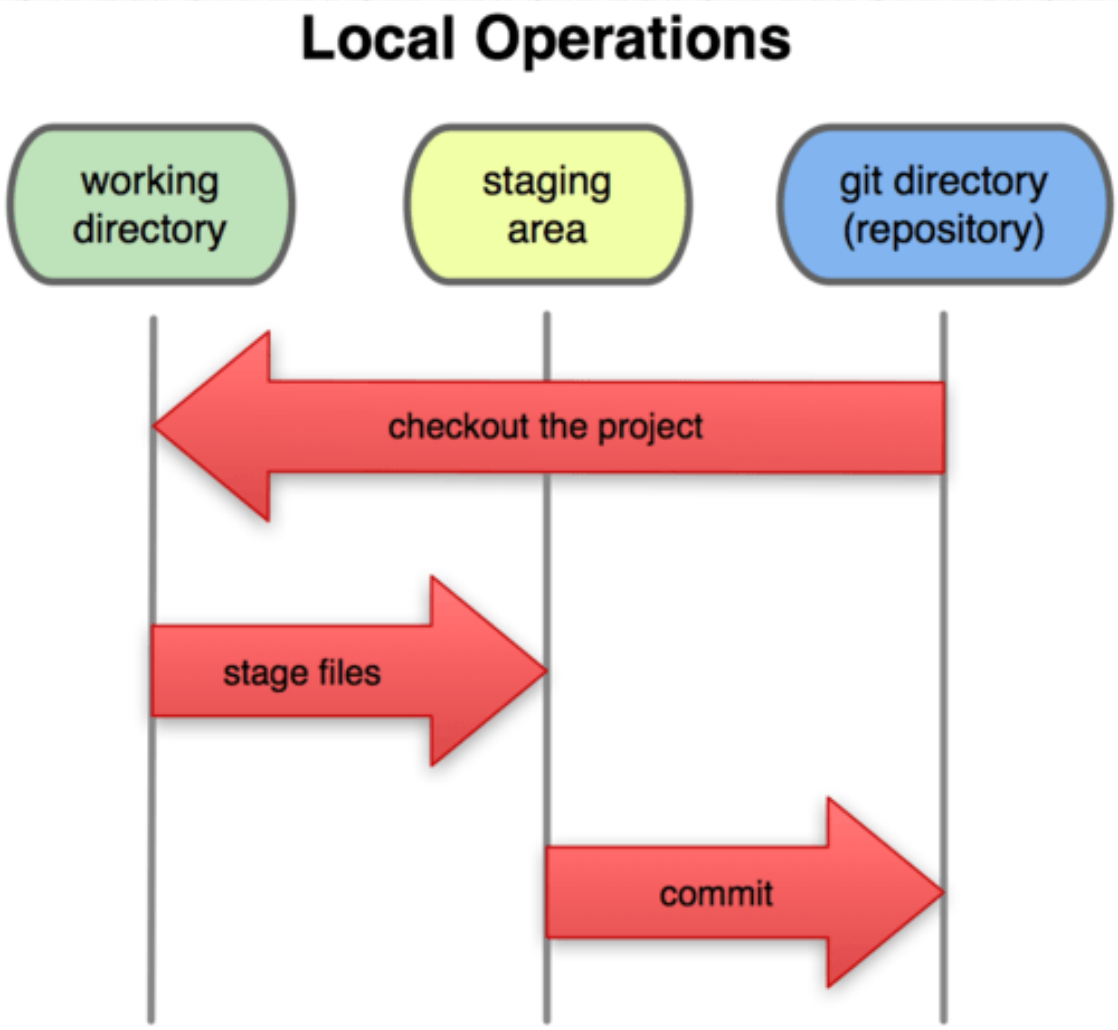
- No sólo descargan la última versión de los archivos, replican completamente el repositorio
- Se elimina la dependencia total al servidor primario
- Se puede trabajar de forma simultánea en distintos grupos
- Permite a los usuarios trabajar de forma productiva cuando no están conectados a la red.
- Estándar de Git y Mercurial



GIT

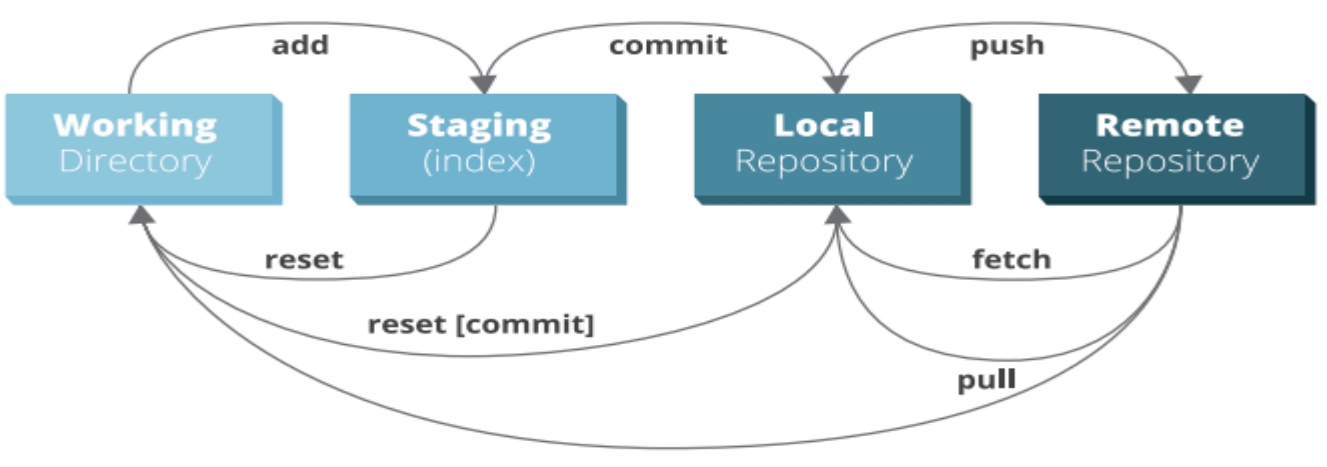
- Diseñado por **Linus Torvalds**
- Comenzó de la mala relación entre la comunidad desarrolladora del núcleo de Linux y la compañía que desarrollaba BitKeeper.
- Git ha evolucionado y madurado, desde 2005, para ser fácil de usar, ser tremendamente rápido, eficiente con grandes proyectos y tener un sistema de ramificación increíble para el desarrollo no lineal

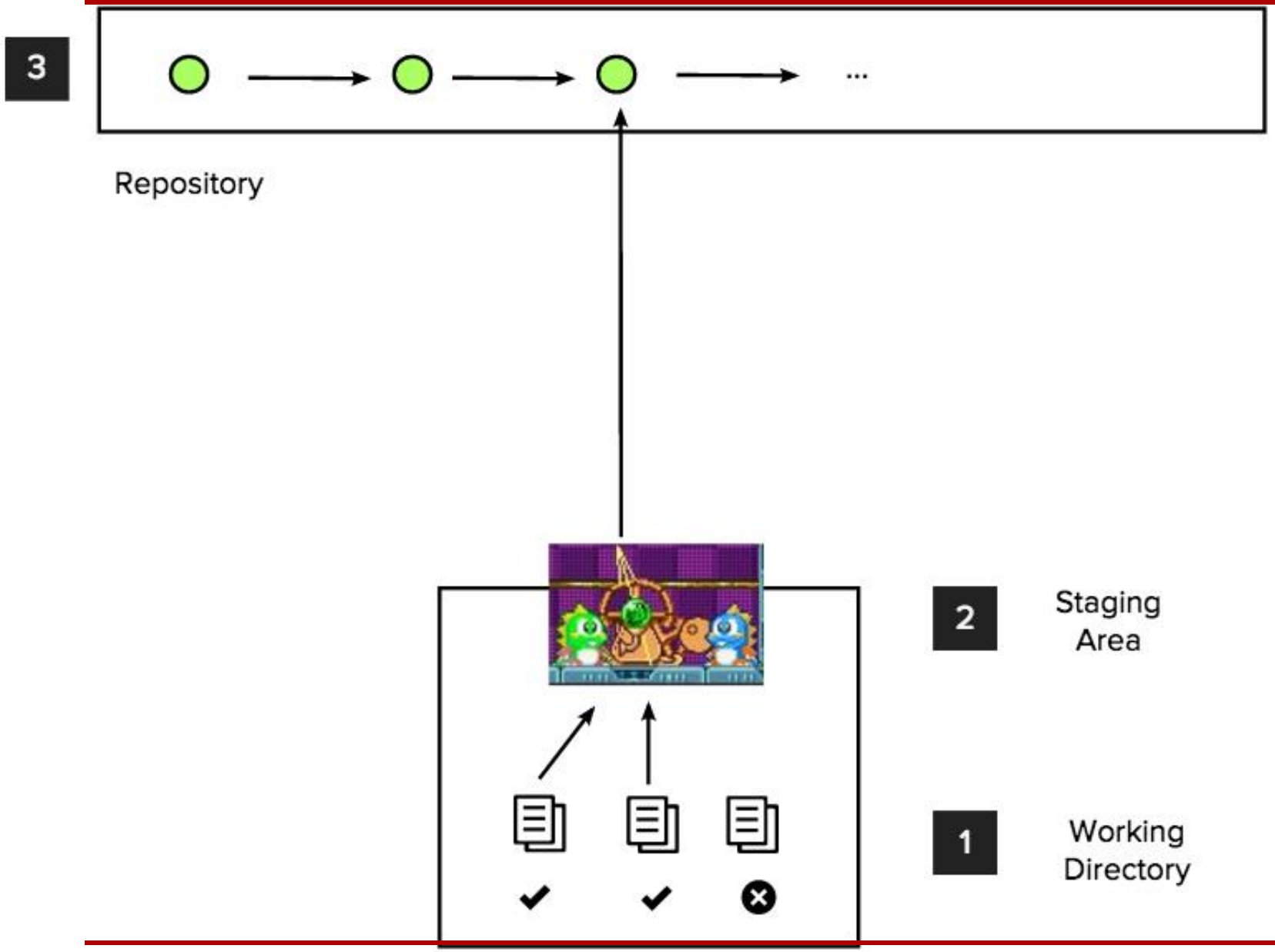
Los tres estados



Flujo de trabajo en Git

1. **Modificar** una serie de archivos en tu directorio de trabajo.
2. **Preparas los archivos**, añadiendo los a tu área de preparación.
3. **Confirmas los cambios**, lo que toma los archivos tal y como están en el área de preparación.
4. **Subir** al repositorio.




3
...
Repository
2
Staging
Area
1
Working
Directory

Instalación

- GNU/Linux
  - Fedora utilizando el comando **yum install git-core**
  - Ubuntu utilizando el comando **apt-get install git**
- Mac
  - http://sourceforge.net/projects/git-osx-installer/
- Windows
  - http://msysgit.github.com/

COMANDOS BASICAS

Crear un nuevo repositorio local
$ **git init**

Clonación de repositorio existente
$ git clone https://domain.com/user/repo.git

Ver los archivos cambiados en el directorio de trabajo
$ git status

Ver los cambios en los archivos monitorizados
$ git diff

Agregar todos los cambios actuales en el siguiente **commit**
$ git add .

Agregar algunos cambios en **<archivo>** en el siguiente **commit**
$ git add <archivo>

Entregar todos los cambios locales en los archivos monitorizados
$ git commit -a

Cambiar el último commit
$ git commit –amend

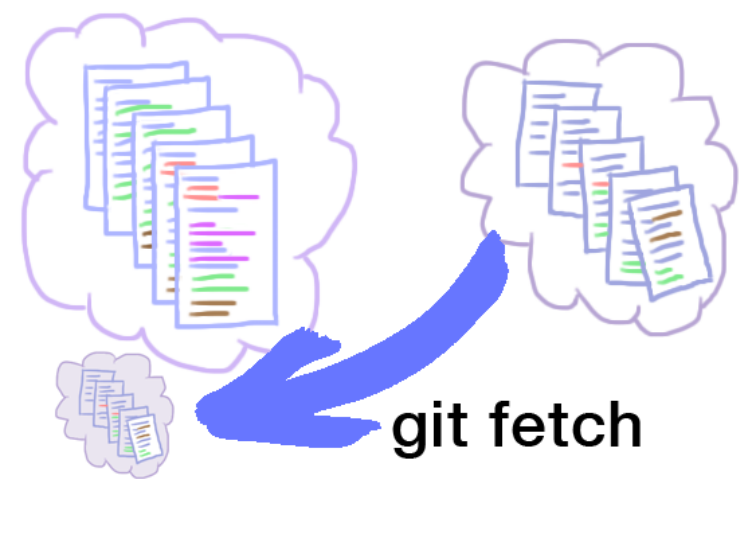
Mostrar todos los commits, iniciando con el más nuevo
$ git log

Mostrar los cambios en el tiempo de un archivo específico
$ git log –p

Ver quién cambió qué y cuándo en un archivo específico
$ git blame

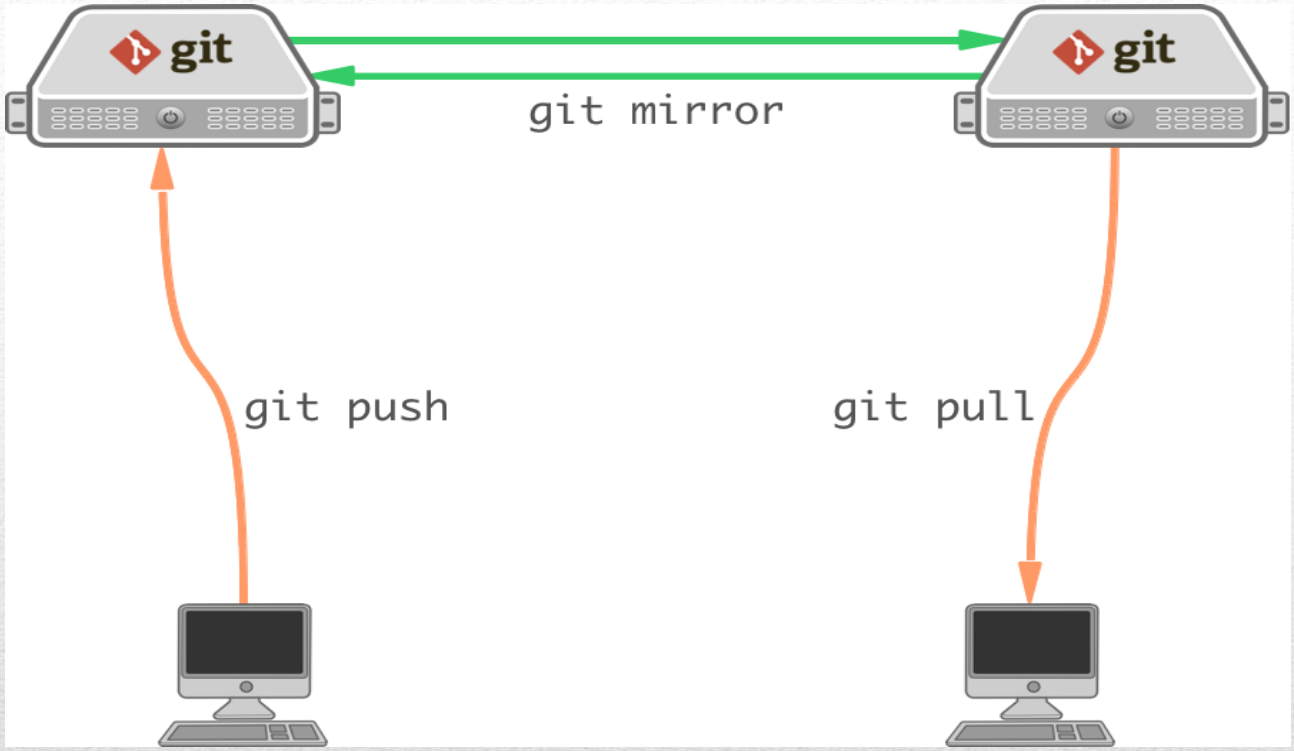
Lista todas las ramas existentes
$ git branch –av

Cambiar rama
$ git checkout nombre_de_rama



git fetch:
Descarga los cambios realizados en el repositorio remoto.

git merge <nombre_rama>:
Impacta en la rama en la que te encuentras parado, los cambios realizados en la rama “nombre_rama”.



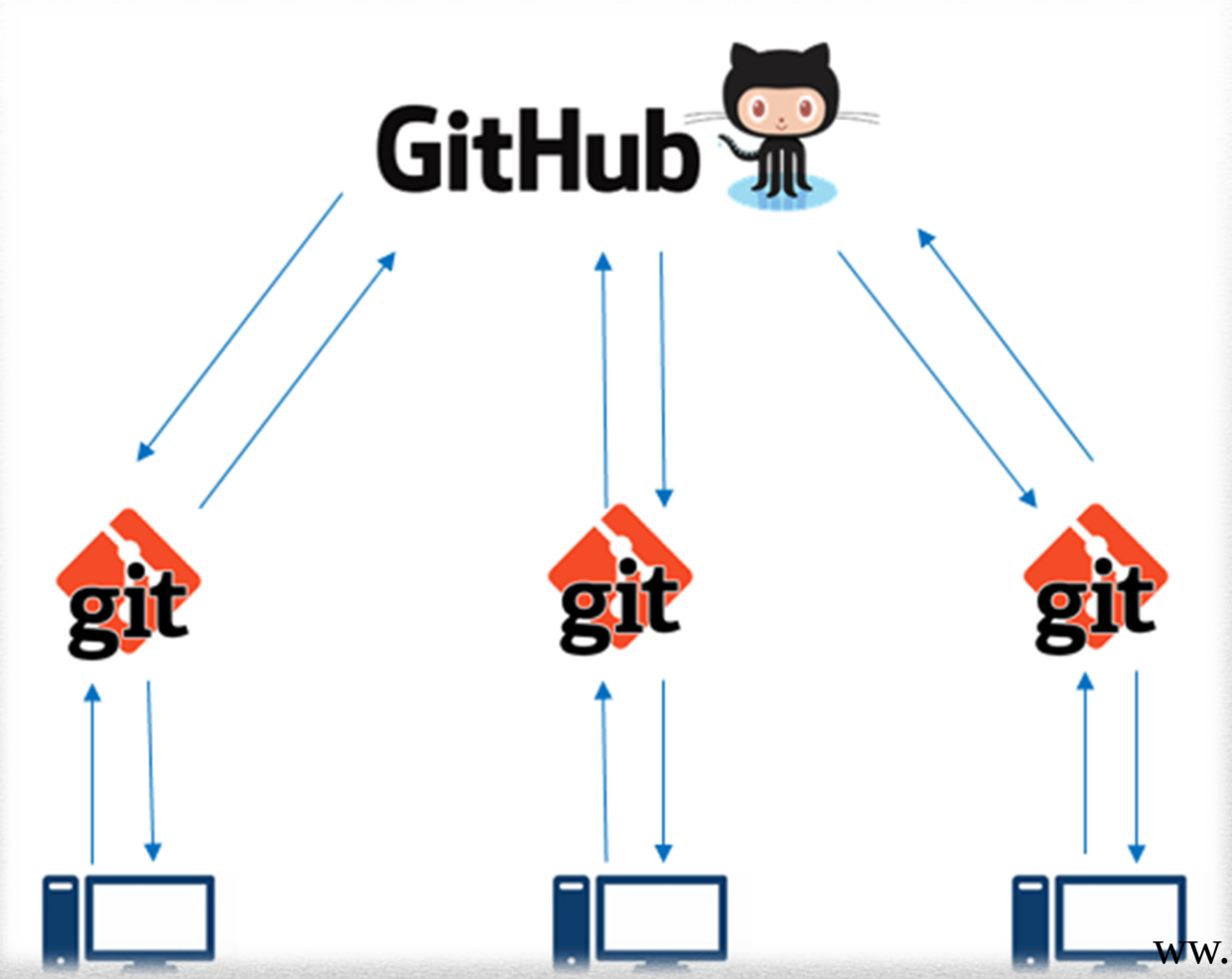
git push origin <*nombre_rama*>:
Sube la rama “nombre_rama” al servidor remoto.

git pull:
Unifica los comandos ***fetch*** **y** ***merge*** en un único comando.

RESUMEN

Modelo de desarrollo	Software libre
Desarrollador(es)	Linus Torvalds, Junio Hamano y Software Freedom Conservancy
Autor(es)	Linus Torvalds
Lanzamiento inicial	7 de abril de 2005
Última versión estable	2.17.1 (info) 22 de mayo de 2018 (24 días)
Género	Control de versiones
Programado en	C, Bourne Shell, Perl[1]
Sistema operativo	Unix-like, Windows
Licencia	GNU GPL v2
Idiomas	inglés

GITHUB



ww.quora.com

CONFIGURE TOOLING

Configure user information for all local repositories

```
$ git config --global user.name "[name]"
```

Sets the name you want attached to your commit transactions

```
$ git config --global user.email "[email address]"
```

Sets the email you want attached to your commit transactions

```
$ git config --global color.ui auto
```

Enables helpful colorization of command line output

CREATE REPOSITORIES

Start a new repository or obtain one from an existing URL

```
$ git init [project-name]
```

Creates a new local repository with the specified name

```
$ git clone [url]
```

Downloads a project and its entire version history

MAKE CHANGES

Review edits and craft a commit transaction

`$ git status`
Lists all new or modified files to be committed
`$ git diff`
Shows file differences not yet staged
`$ git add [file]`
Snapshots the file in preparation for versioning
`$ git diff --staged`
Shows file differences between staging and the last file version
`$ git reset [file]`
Unstages the file, but preserve its contents
`$ git commit -m "[descriptive message]"`
Records file snapshots permanently in version history

GROUP CHANGES

Name a series of commits and combine completed efforts

$ git branch
Lists all local branches in the current repository
$ git branch [branch-name]
Creates a new branch
$ git checkout [branch-name]
Switches to the specified branch and updates the working directory
$ git merge [branch]
Combines the specified branch's history into the current branch
$ git branch -d [branch-name]
Deletes the specified branch


GRUPO A
Rusia
Arabia Saudí
Egipto
Uruguay
GRUPO B
Portugal
España
Marruecos
Irán
GRUPO C
Francia
Australia
Perú
Dinamarca
GRUPO D
Argentina
Islandia
Croacia
Nigeria
TALLER
GRUPO E
Brasil
Suiza
Costa Rica
Serbia
GRUPO F
Alemania
México
Suecia
Corea Sur
Bélgica
Panamá
Túnez
Inglaterra
GRUPO H
Polonia
Senegal
Colombia
Japón
2018
FIFA
WORLD
CUP
RUSSIA

FUENTES

- https://git-scm.com/book/es/v1/
- https://www.hostinger.es/tutoriales/comandos-de-git
- https://medium.com/@david25lo/gesti%C3%B3n-de-proyectos-con-git-github-c046412f5bb0
- https://github.com/ampotty/uip-pc3/blob/master/01.Manejo.Repositorios.Git/manejogit.pdf
- https://es.wikipedia.org/wiki/Git#Caracter%C3%ADsticas
- https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet#images

GRACIAS……