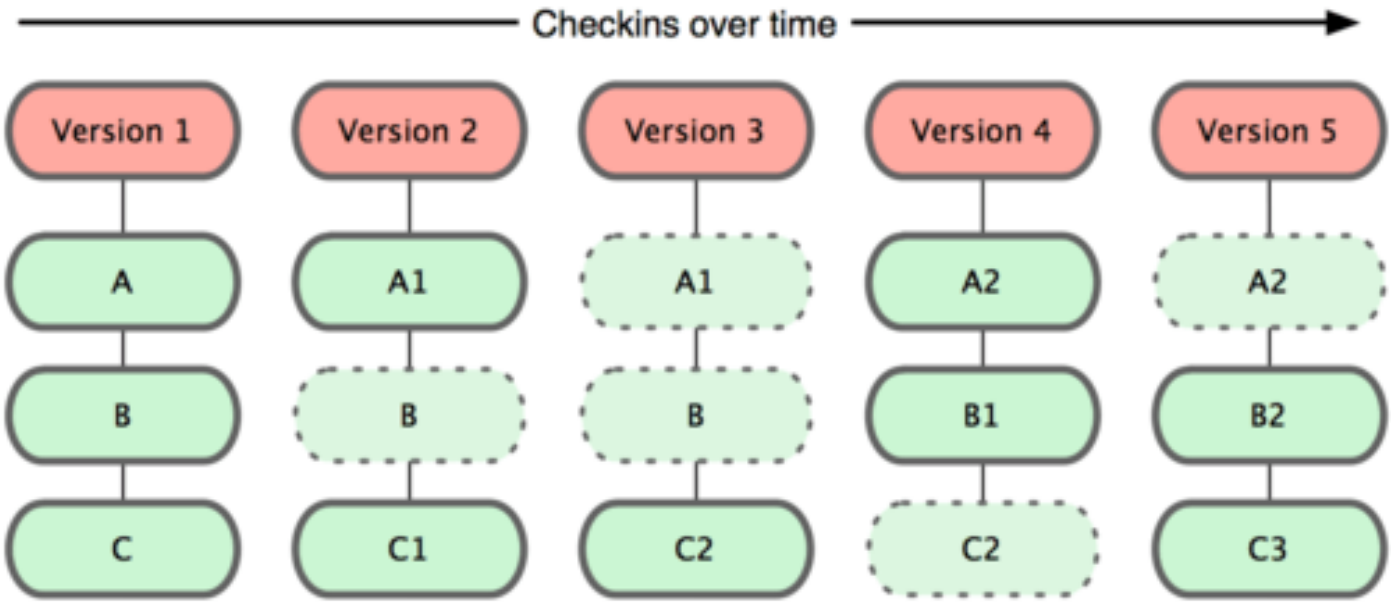
Introducción a Git

¿Qué es git?

Control de versiones

Seguimiento de uno o varios archivos dentro de un repositorio.



Configurando git

Antes de usar git hay que hacer una configuración inicial. Desde la terminal:

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

$ git config --global user.email "a@s.es"

$ git config --global core.editor nano
```

Creación de un repositorio

1 Crear 'Nueva carpeta'

```
$ mkdir <new_dir>
```

2 Entramos dentro de la carpeta

```
$ cd <new_dir>
```

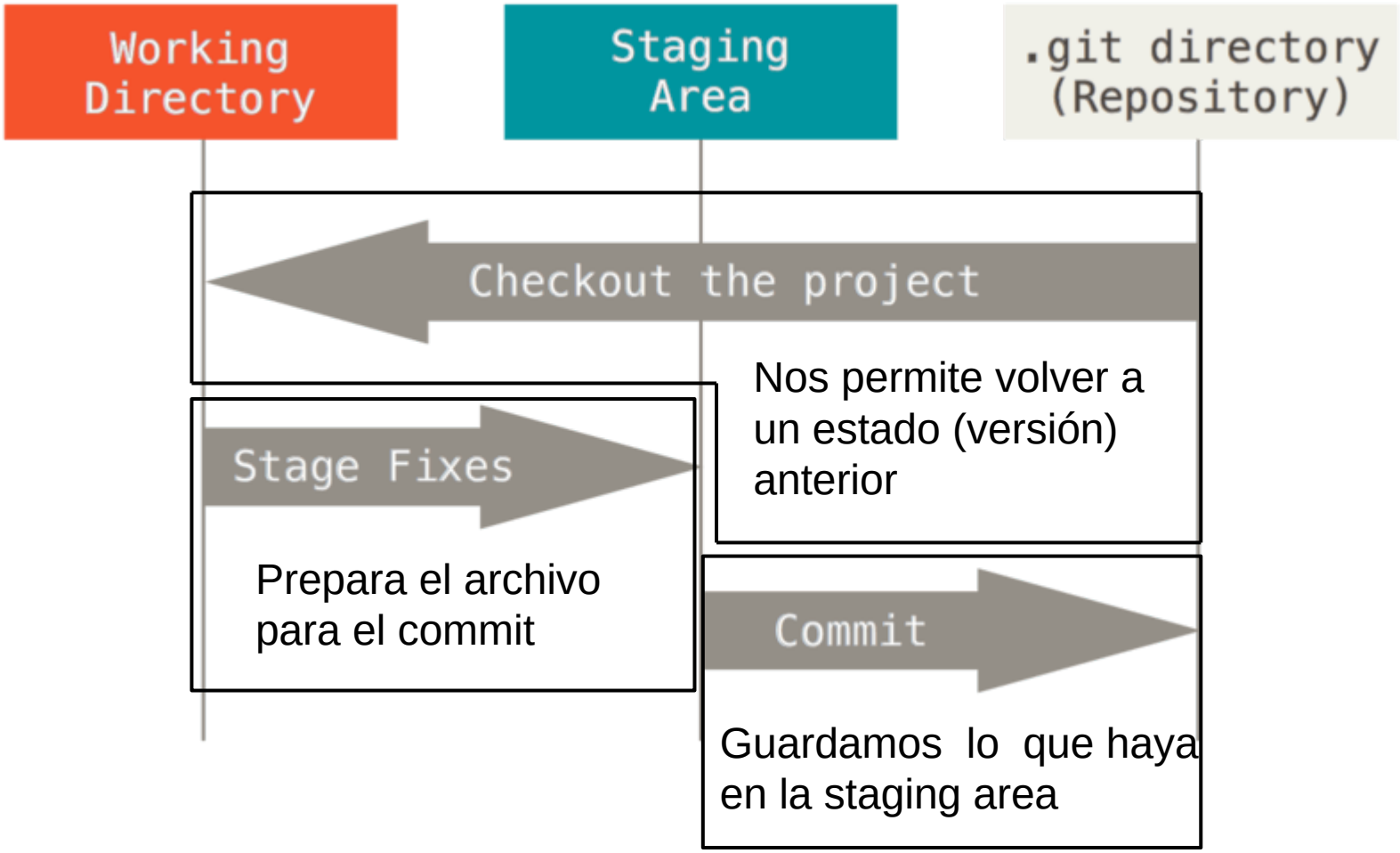
3 Creamos el repositorio

```
$ git init
```

O si queremos clonar un repositorio remoto (sin crear una carpeta nueva)

```
$ git clone <URL_repositorio>
```

Cómo funciona git



Estado del repositorio

```
$ git status
```

Nos da información del estado del repositorio. Si hay archivos sin seguimiento, modificados, en el ‘staging area’...

Seguimiento de archivos

Seleccionamos los archivos que queremos seguir con

```
$ git add <nombre archivo>
```

Con git add hacemos una "imagen" del archivo como esté en ese momento. Si hacemos cambios en ese archivo debemos repetir el comando.

Nuestro archivo está ahora en la "staging area"

Commit

El paso final para guardar nuestra "imagen"

```
$ git commit –m 'message'
```

El ‘mensaje’ debe ser lo más descriptivo posible para futuras búsquedas de versiones anteriores

Historial del repositorio

```
$ git log
```

Nos muestra información de los commits que hayamos realizado desde el más reciente. ('q' para salir si es un log largo)

```
commit 2d2af513688bc90c041d4cb5c6079b69c88ad2e2
 Author: miguel_windows <m.omullony@ceab.csic.es>
 Date:   Thu Dec 7 13:06:01 2017 +0100

 chapter 1
```

Volviendo a una versión anterior de un archivo

Sabiendo el número de commit (con los 6 o 7 primeros números) viendo el historial podemos volver a una "imagen" anterior del archivo.

```
$ git checkout <nº commit> <file>
```

Podemos ir moviéndonos entre versiones del archivo que queramos

Volviendo a una versión anterior del repositorio

```
$ git checkout <nº commit>
```

Ahora todos los archivos de la carpeta están en la versión del commit que queramos.

Puede ser útil si hemos eliminado un archivo o se ha quedado corrupto. (Perderemos los últimos cambios, pero no habremos perdido todo lo demás).

Para volver de nuevo a nuestra rama principal, es decir, la versión más avanzada:

```
$ git checkout master
```

Cambio de nombre de un archivo

Para que git detecte que ha habido un cambio de nombre en un archivo (que ya esté en seguimiento) y no lo vea como un archivo nuevo.

```
$ git mv <file> <new_file>
```

En resumen

Una vez que hayamos terminado de editar un archivo hacemos una imagen del archivo con `git add` y después guardamos la imagen en el repositorio con

`git commit –m 'text'`

El mensaje que escribamos al hacer el commit deberá ser lo más descriptivo posible para que encontremos más fácilmente una versión anterior del archivo con

`git checkout <commit> <file>`

Repositorio remoto

Nos permite tener una copia de nuestro repositorio en un servidor externo, como Github, Gitlab...

Además varias personas pueden trabajar en el mismo repositorio.

Una vez hecho un commit actualizamos el repositorio remoto:

```
$ git push
```

Configuración para usar rep. remotos

Necesitamos una llave ssh para que el servidor y nuestro ordenador se reconozcan y poder hacer

https://help.github.com/articles/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent/#platform-mac

https://help.github.com/articles/adding-a-new-ssh-key-to-your-github-account/

Más información

https://services.github.com/on-demand/downloads/github-git-cheat-sheet.pdf
(Un pdf con los comandos de git más usados)

https://git-scm.com/book/en/v2
(documentación sobre git)