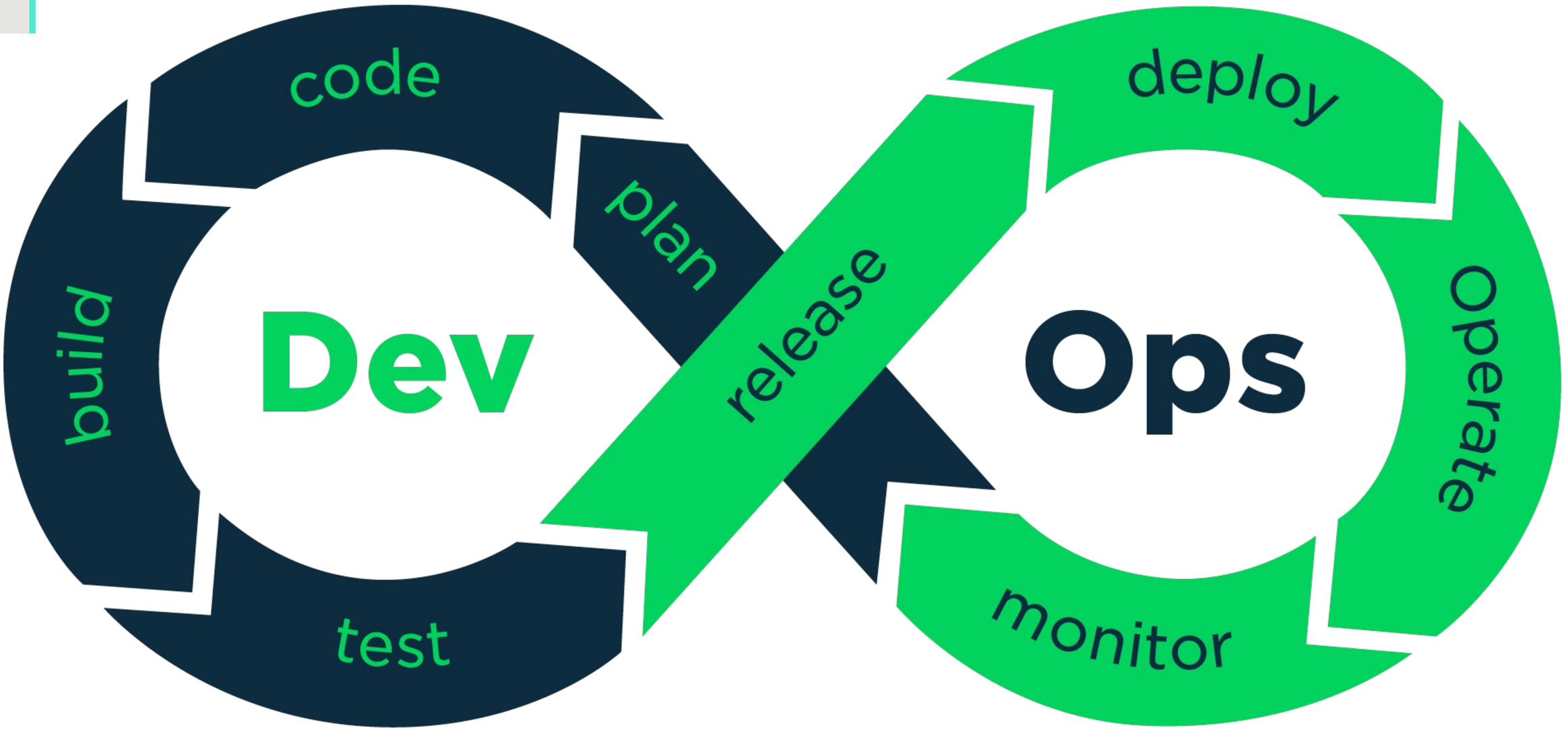
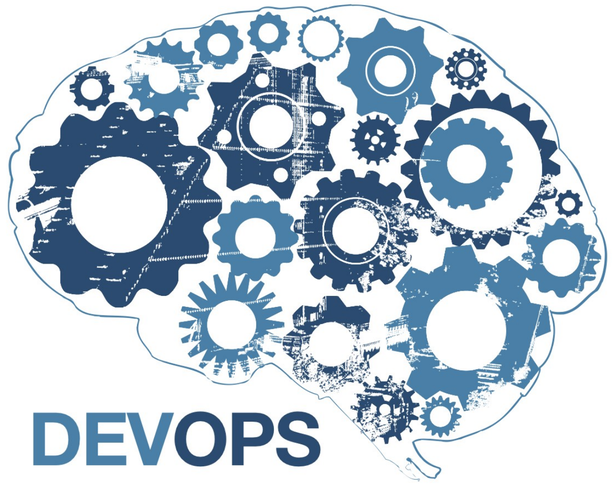

code
plan
build
Dev
test
release
deploy
Ops
operate
monitor



INTERMEDIO

Unidad 03

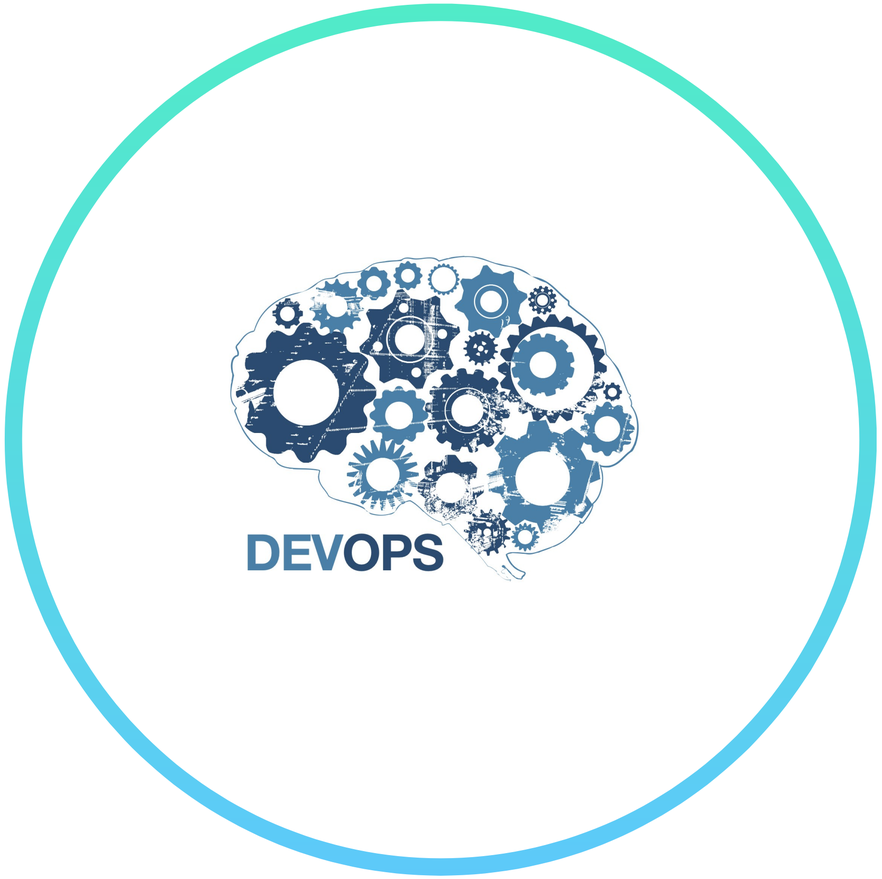
Git

CURSOS DE DESARROLLO

O

Objetivos

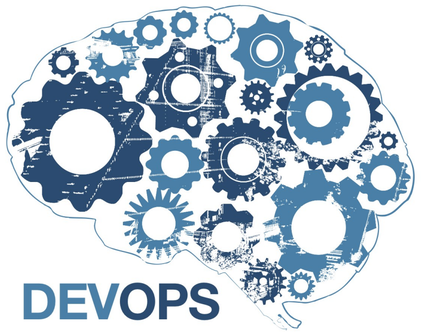
¿Qué voy a a aprender?



- Saber crear un repositorio
- Saber manejar un repositorio

Contenidos

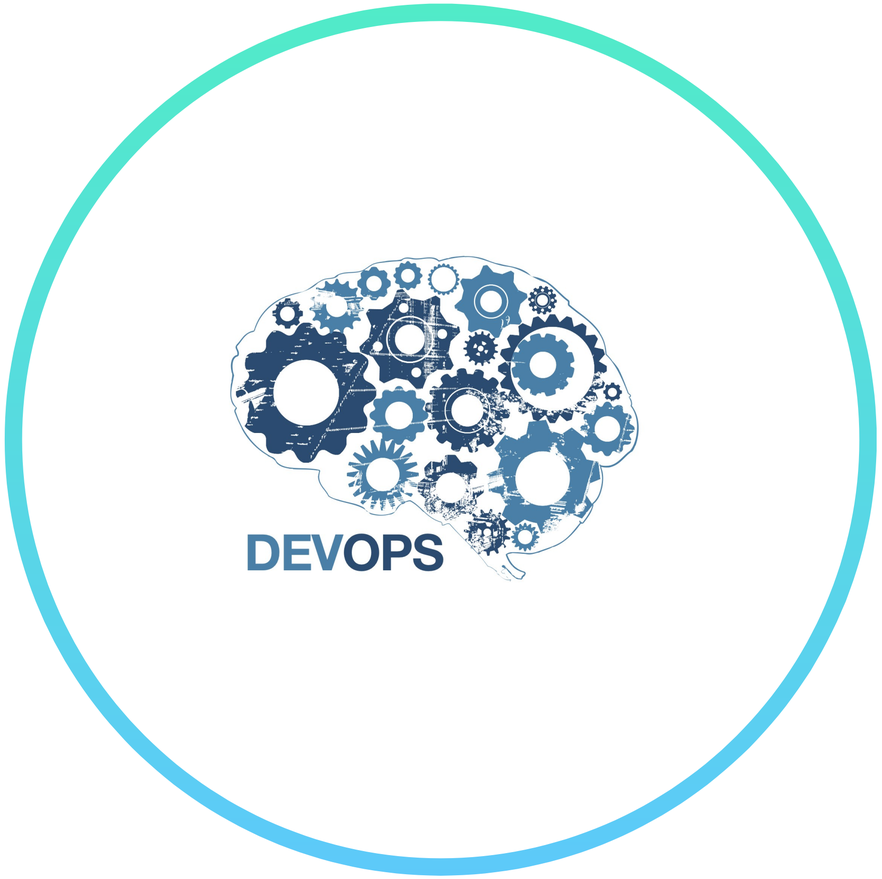
¿Cómo voy a aprenderlo?



1. Introducción
2. Instalación
3. Gestión de un repositorio
4. GUI’s
5. Github
6. Gitlab
7. Demo
8. Conclusiones

Introducción

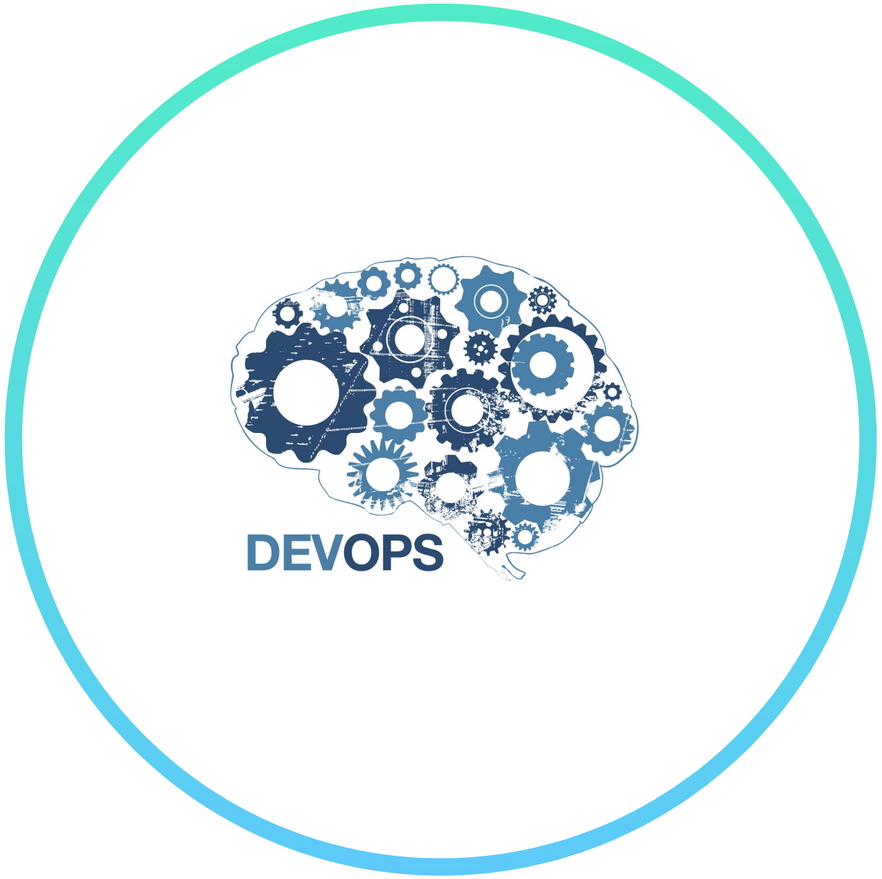
¿Por dónde empezamos?



¿Qué es Git?

Introducción

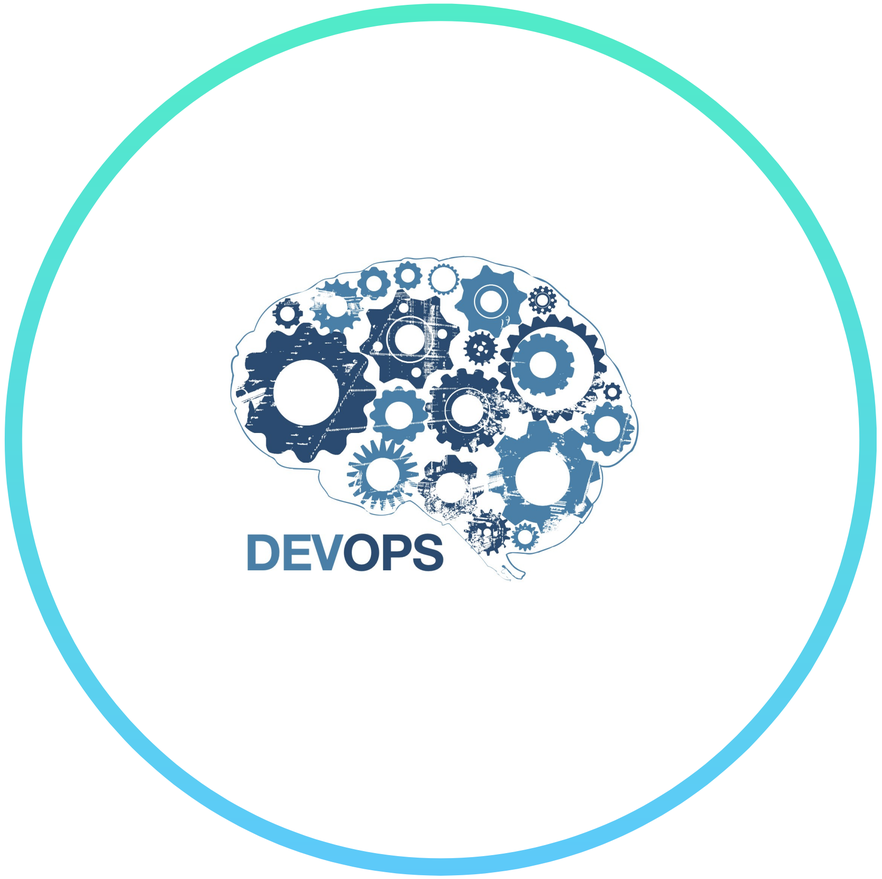
¿Por dónde empezamos?



Git es un sistema de control de versiones distribuido de carpetas y ficheros

Introducción

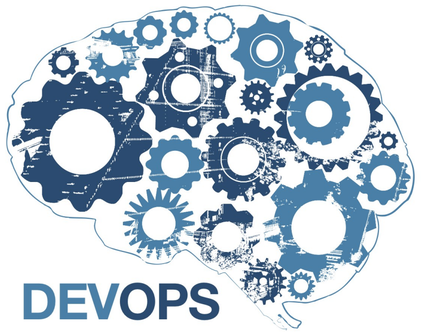
¿Por dónde empezamos?



¿En qué consiste un sistema de versiones?

Introducción

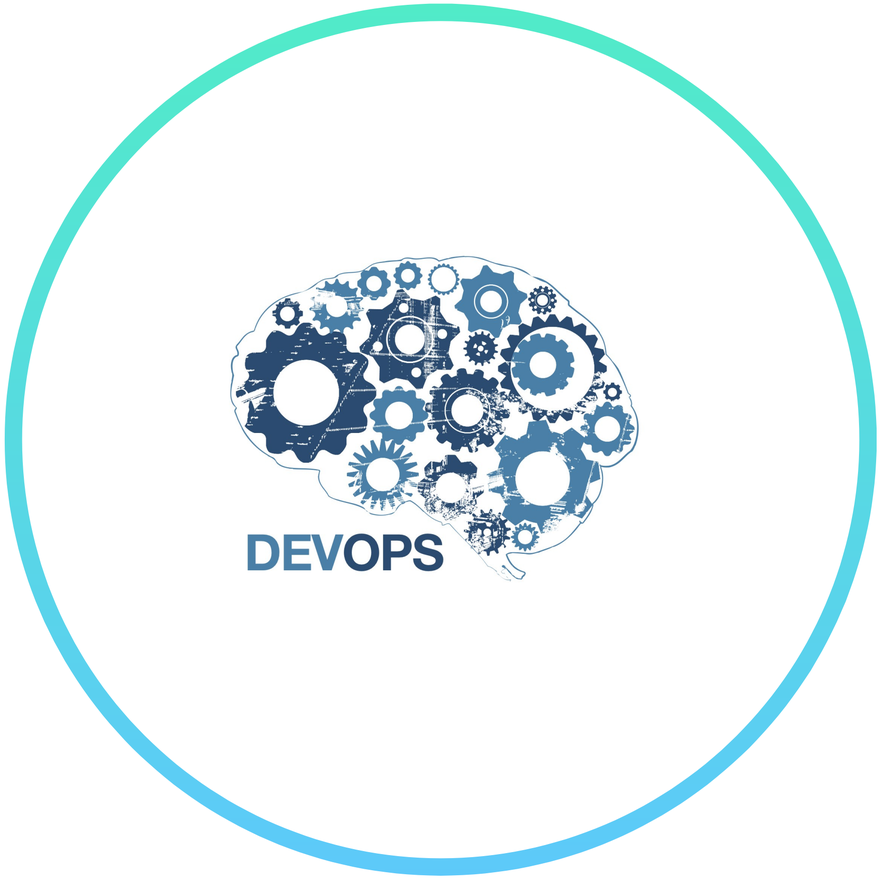
¿Por dónde empezamos?



Maneja un histórico de los ficheros y carpetas que pertenecen a un proyecto pudiendo recuperar el estado de un proyecto dado

Introducción

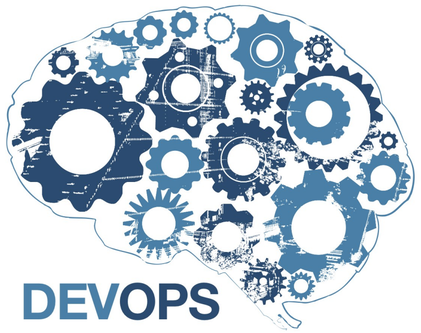
¿Por dónde empezamos?



¿Cómo se crea un estado?

Introducción

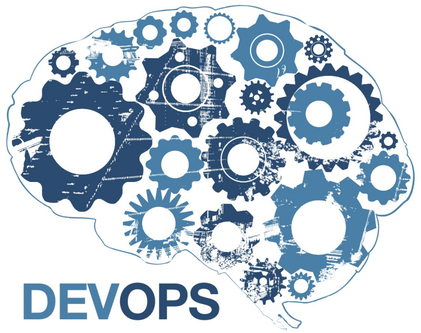
¿Por dónde empezamos?



Se crea un estado nuevo cada vez que se hace un commit, es decir, se guardan los cambios en un repositorio

Introducción

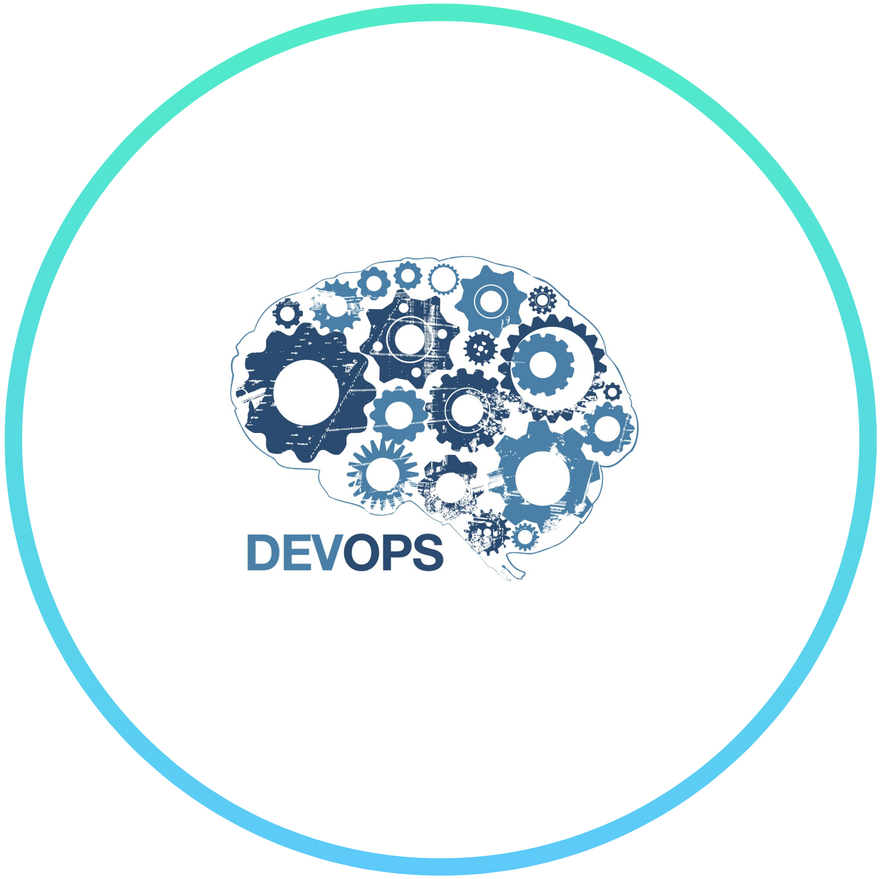
¿Por dónde empezamos?



Se establecen distintas versiones del proyecto en total

Introducción

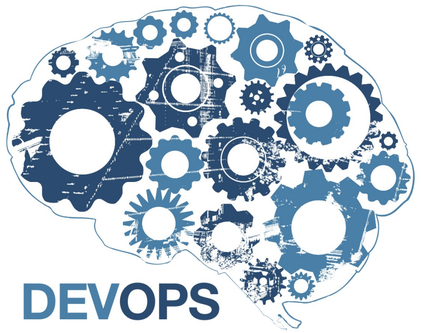
¿Por dónde empezamos?



En cada versión del repositorio se pueden ver los cambios provocados en cada fichero o carpeta

Introducción

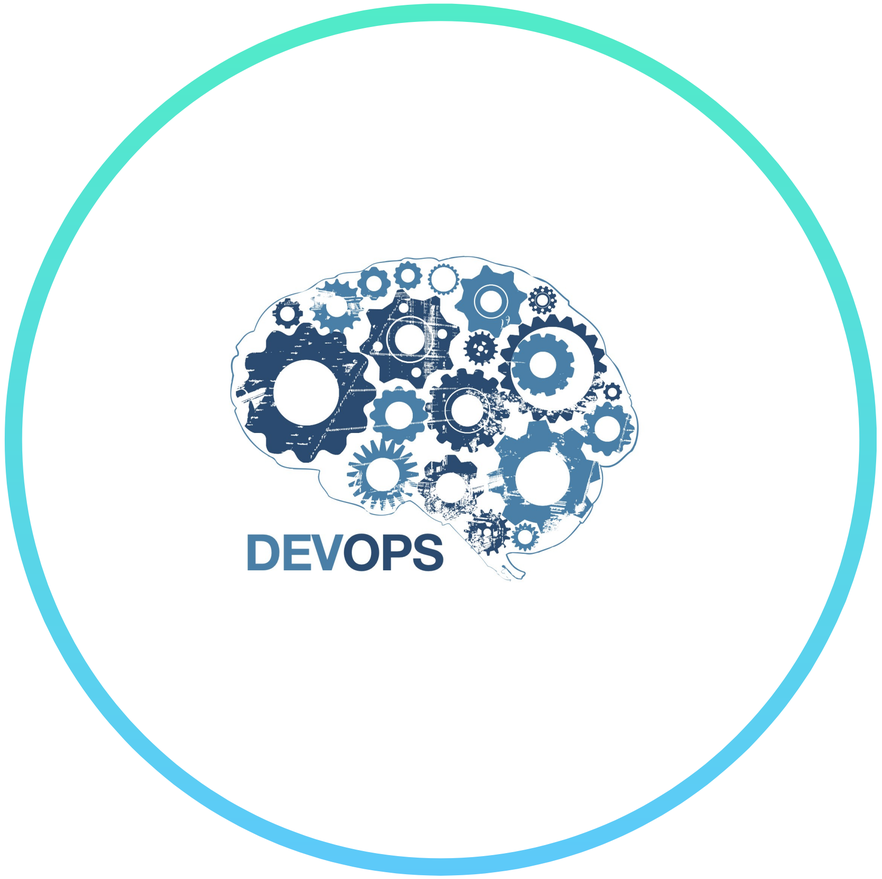
¿Por dónde empezamos?

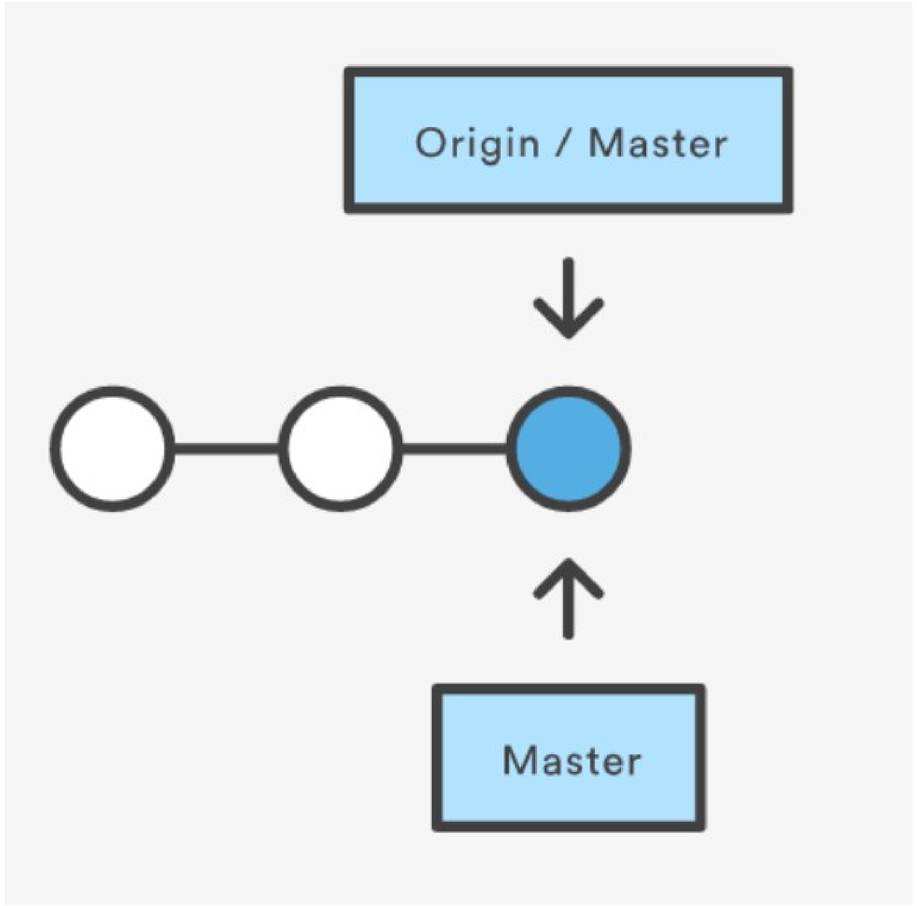


Estos cambios son las diferencias entre una versión y otra

Introducción

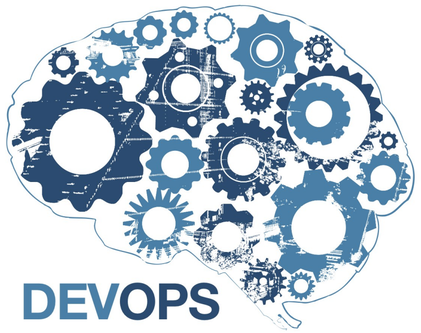
¿Por dónde empezamos?





Introducción

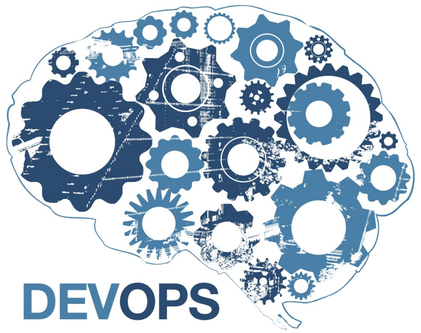
¿Por dónde empezamos?



Estos cambios suelen llevar asociado un determinado mensaje que contextualiza el cambio, arreglo de un fallo, nueva funcionalidad, etc…

Introducción

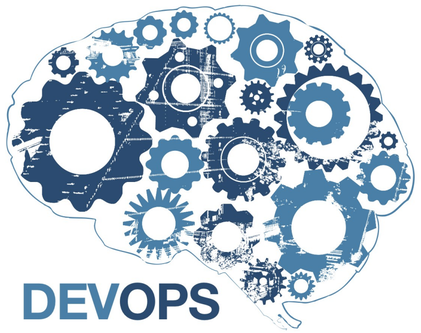
¿Por dónde empezamos?



Hasta aquí sería parecido a otros sistemas de versiones como SVN o CVS

Introducción

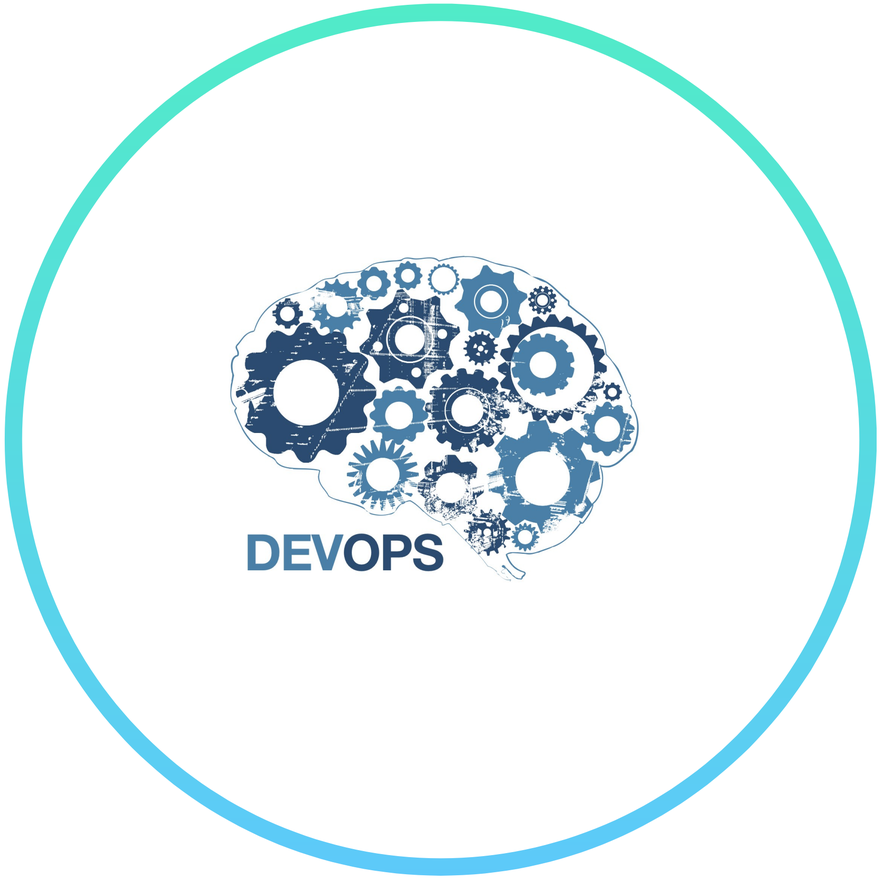
¿Por dónde empezamos?



Entonces, ¿Porqué usar Git?

Introducción

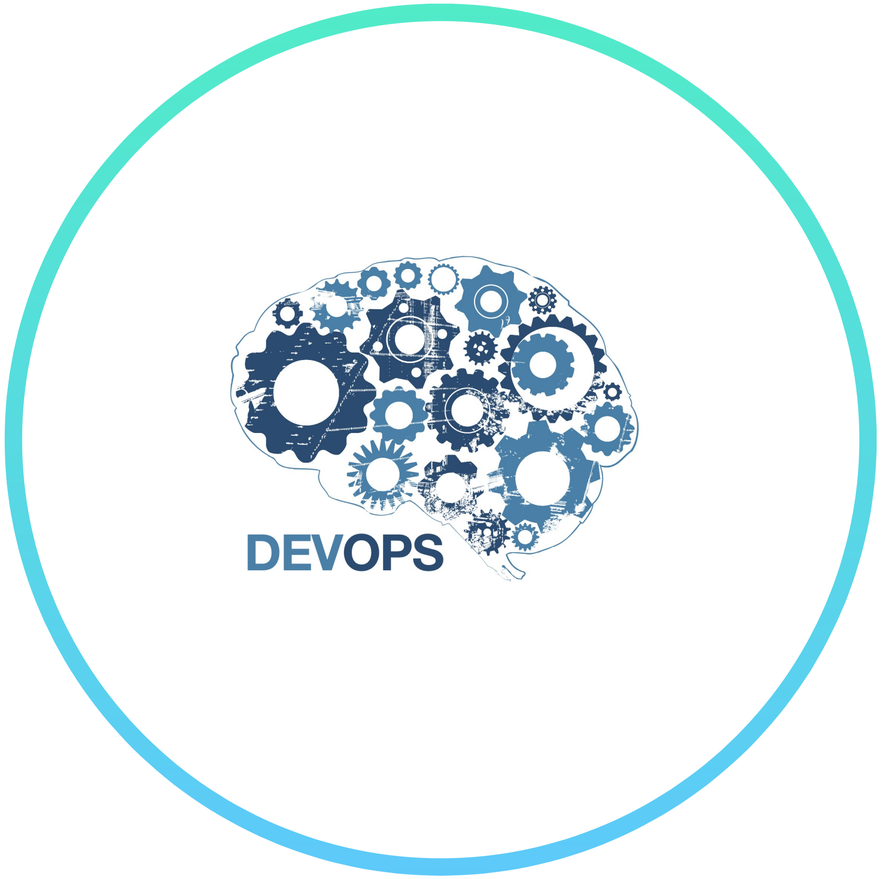
¿Por dónde empezamos?



Ventajas que tiene Git sobre otros sistemas de control de versiones

Introducción

¿Por dónde empezamos?

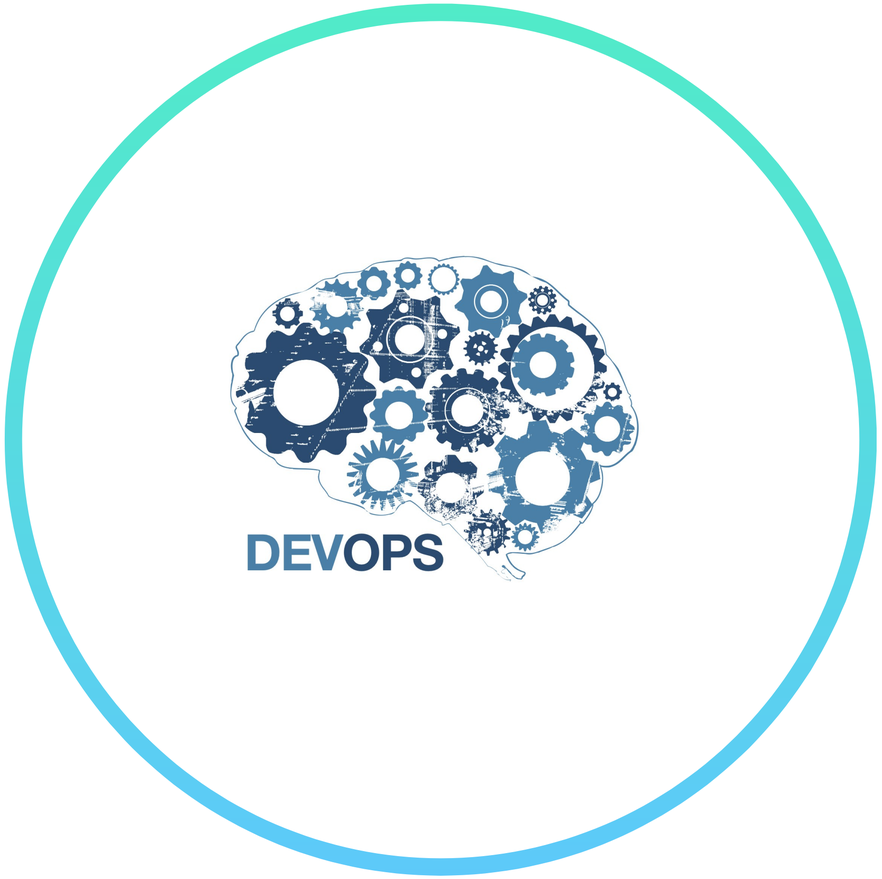


Ramificación

Permite crear distintas ramas para el código a través del cual podemos hacer el seguimiento del desarrollo de un proyecto

Introducción

¿Por dónde empezamos?

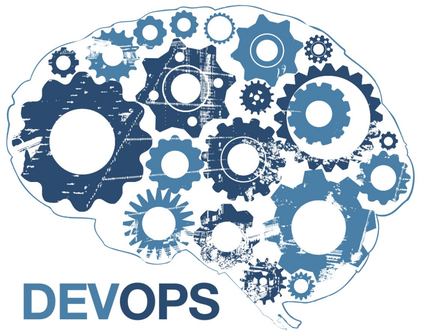


Convergencia

Se pueden fusionar ramas para hacer una convergencia de código

Introducción

¿Por dónde empezamos?

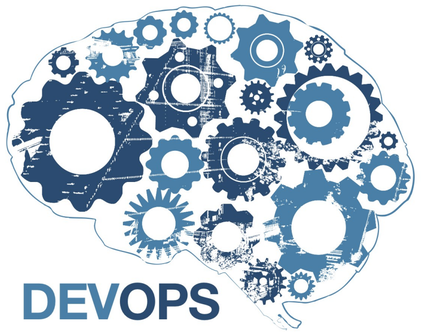


Distribución

Cada persona tiene la posibilidad de duplicar el repositorio y tener un repositorio local para que cada repositorio sea independiente y los cambios no afecten a los demás necesariamente

Introducción

¿Por dónde empezamos?

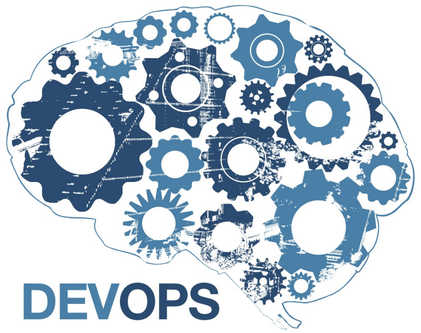


Adaptabilidad

Según las necesidades de cada proyecto la estructura de versiones se puede ajustar a cada equipo y cada proyecto

Introducción

¿Por dónde empezamos?

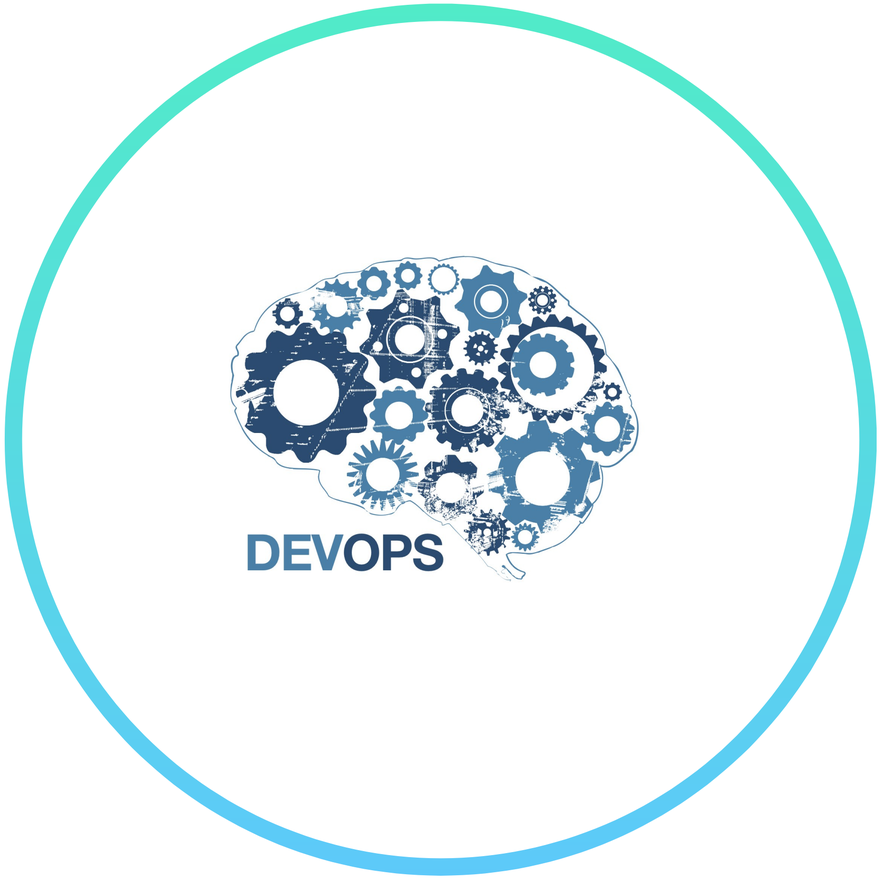


Pull Request

Permite que otras personas puedan llegar a querer hacer commits en nuestro repositorio

Instalación

¿Cómo lo instalamos?

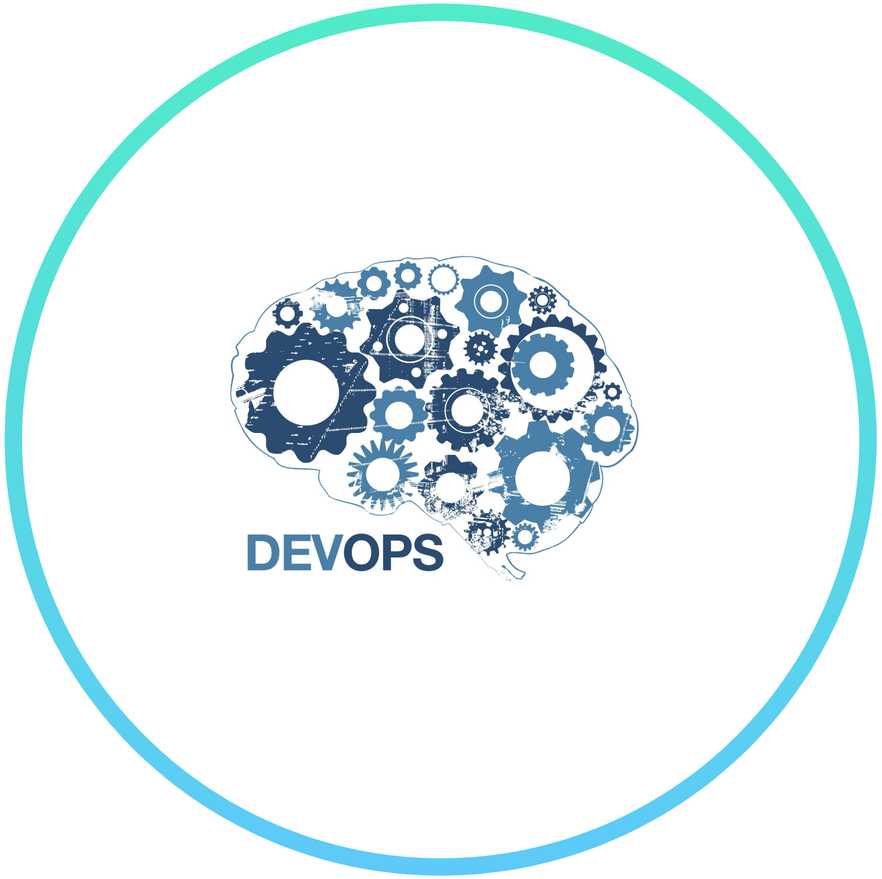


Sitio Oficial

https://git-scm.com/

Instalación

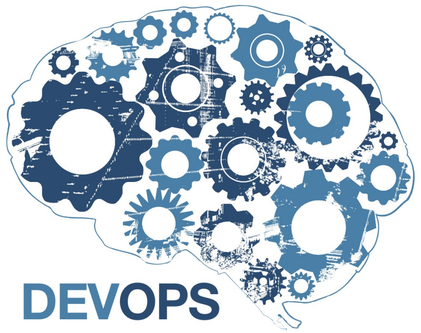
¿Cómo lo instalamos?



Descarga
https://git-scm.com/downloads

Creación Repositorio

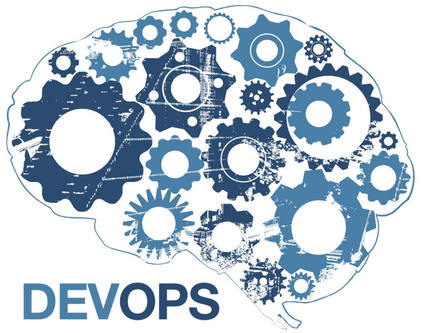
¿Cómo creamos un repositorio?



Sobre una carpeta en la que queremos gestionar un proyecto ejecutamos:
git init

Creación Repositorio

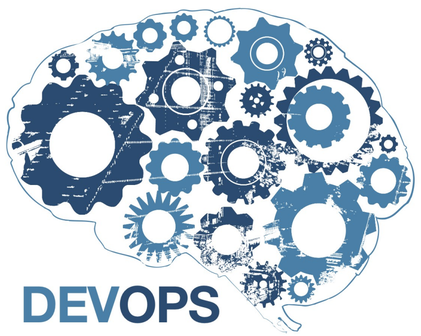
¿Cómo creamos un repositorio?



Con esto creamos un repositorio en esa carpeta

Creación Repositorio

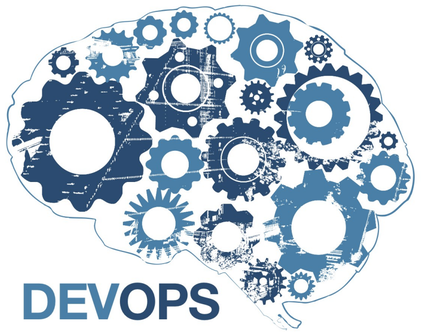
¿Cómo creamos un repositorio?



Si queremos descargarnos un repositorio público, debemos clonarlo:
git clone (URL_REPO) (Directorio destino)

Creación Repositorio

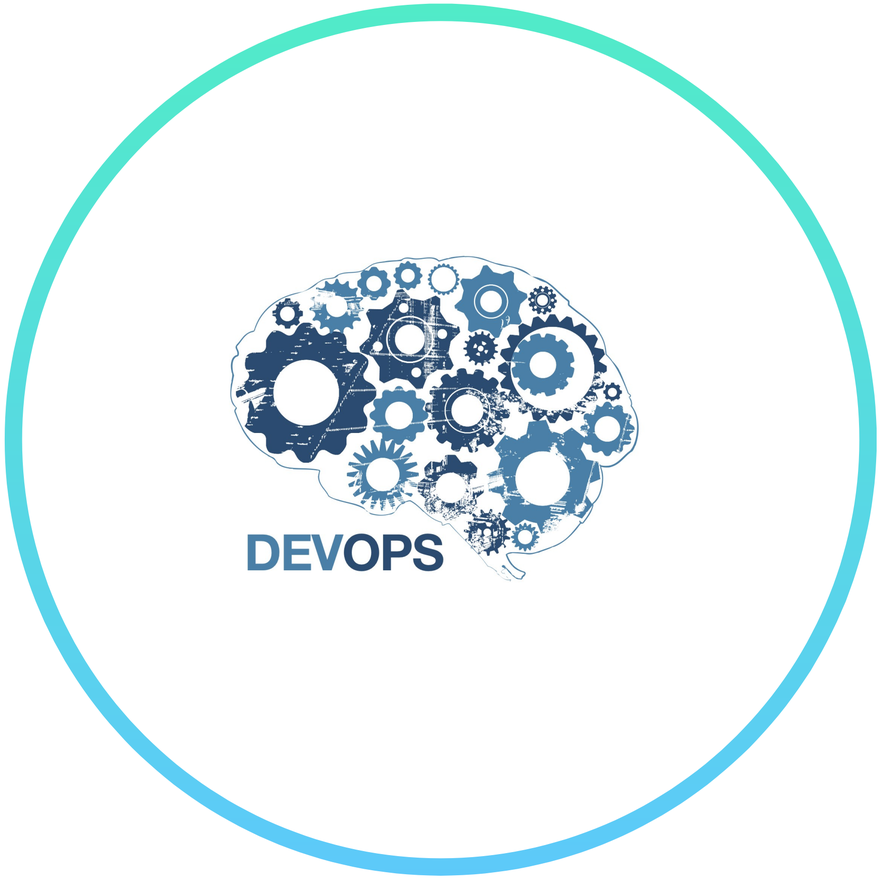
¿Cómo creamos un repositorio?



En ambos casos podemos gestionar ya los ficheros que hay en ese repositorio

Creación Repositorio

¿Cómo creamos un repositorio?



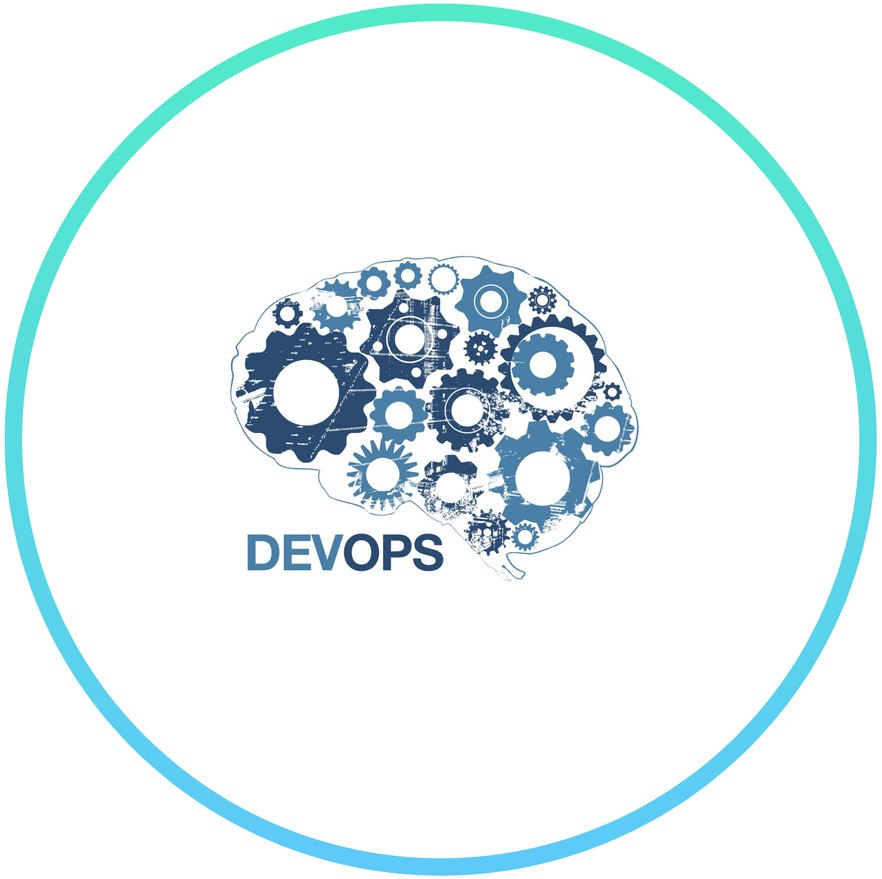
Siempre está bien configurar los datos de nuestro usuario del repositorio, sobre todo de cara a luego compartir dicho repositorio

git config --global user.name <name>

git config --global user.email <email>

Creación Repositorio

¿Cómo creamos un repositorio?



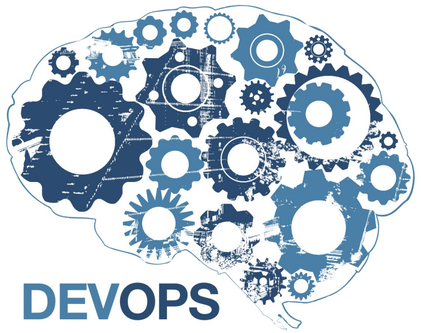
Para añadir un archivo o conjunto de archivos al repositorio ejecutamos:

git add <file>

git add <directory>

Creación Repositorio

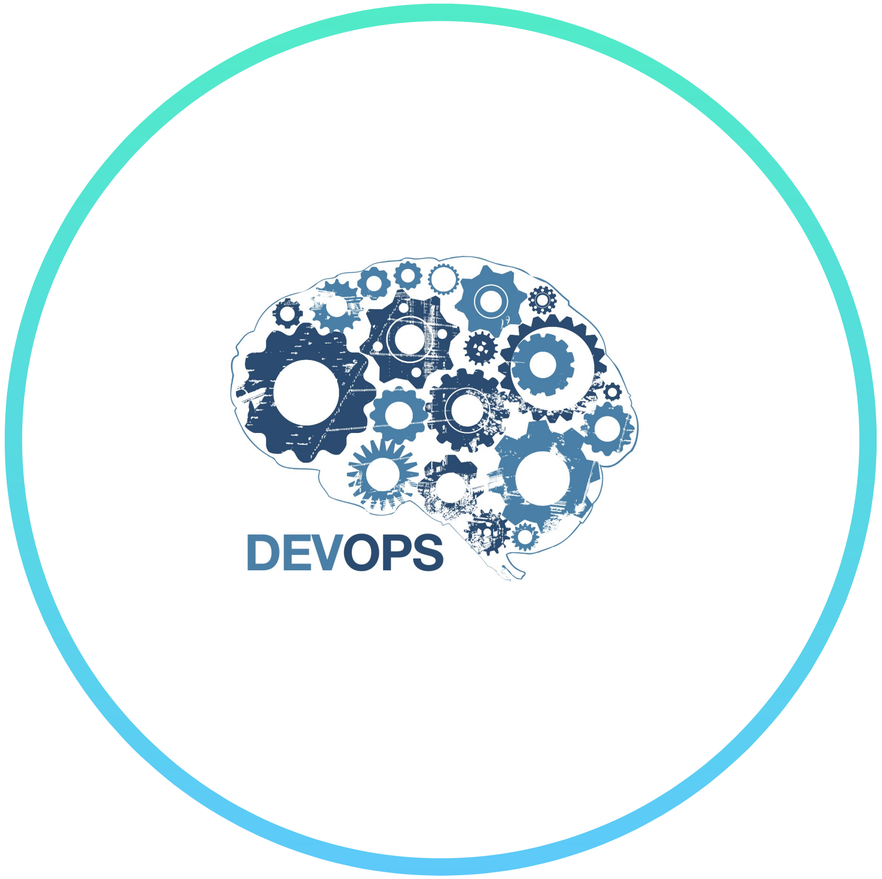
¿Cómo creamos un repositorio?



Al añadir los archivos es como decirle al repositorio que queremos gestionar los archivos con dicho repositorio, pero si queremos guardar esos cambios como una nueva versión es necesario crear un nuevo commit

Creación Repositorio

¿Cómo creamos un repositorio?

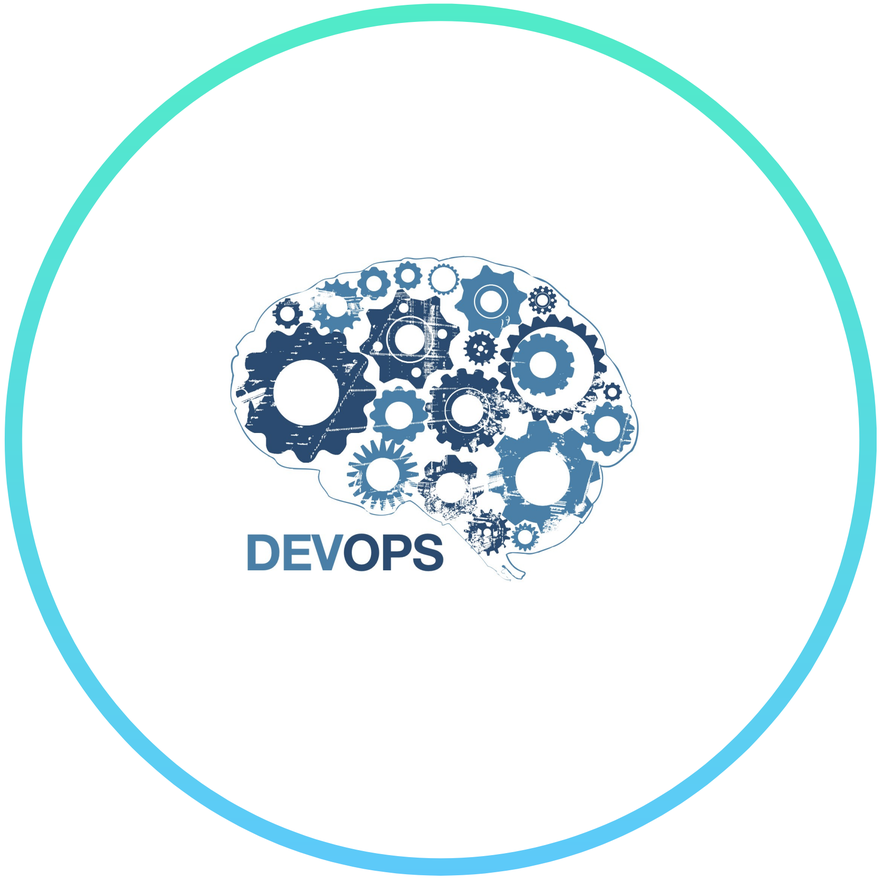


Para crear un nuevo commit (versión estable):

git commit -m ‘datos del cambio’

Creación Repositorio

¿Cómo creamos un repositorio?

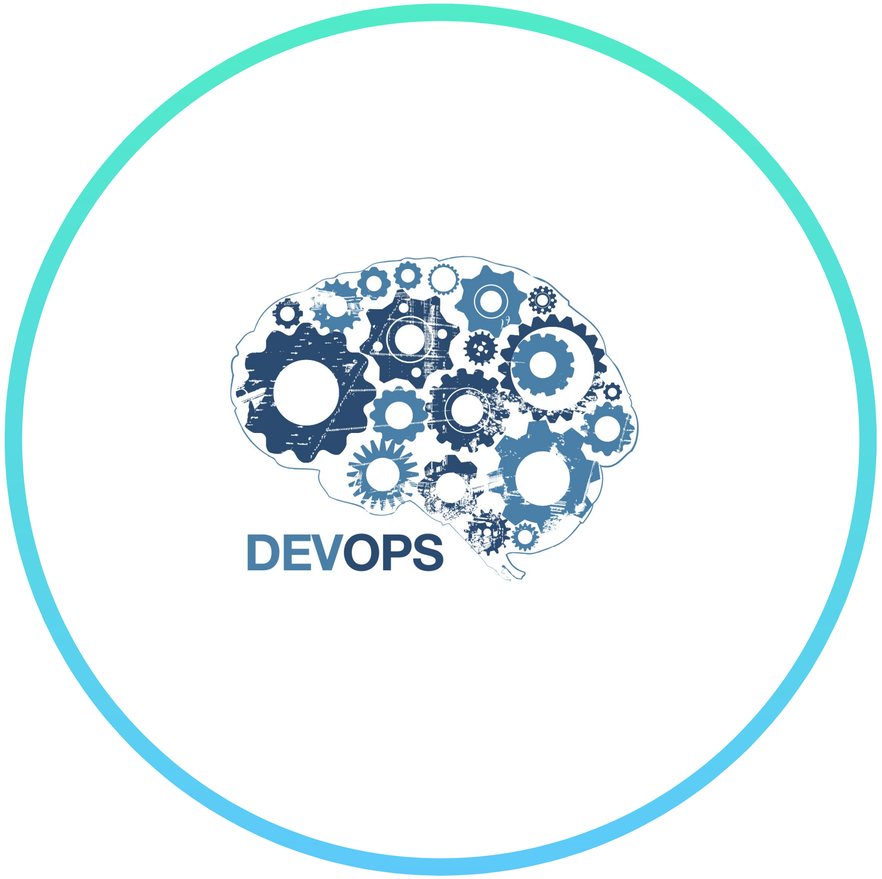


Para ver los estado en un repositorio:

git status

Creación Repositorio

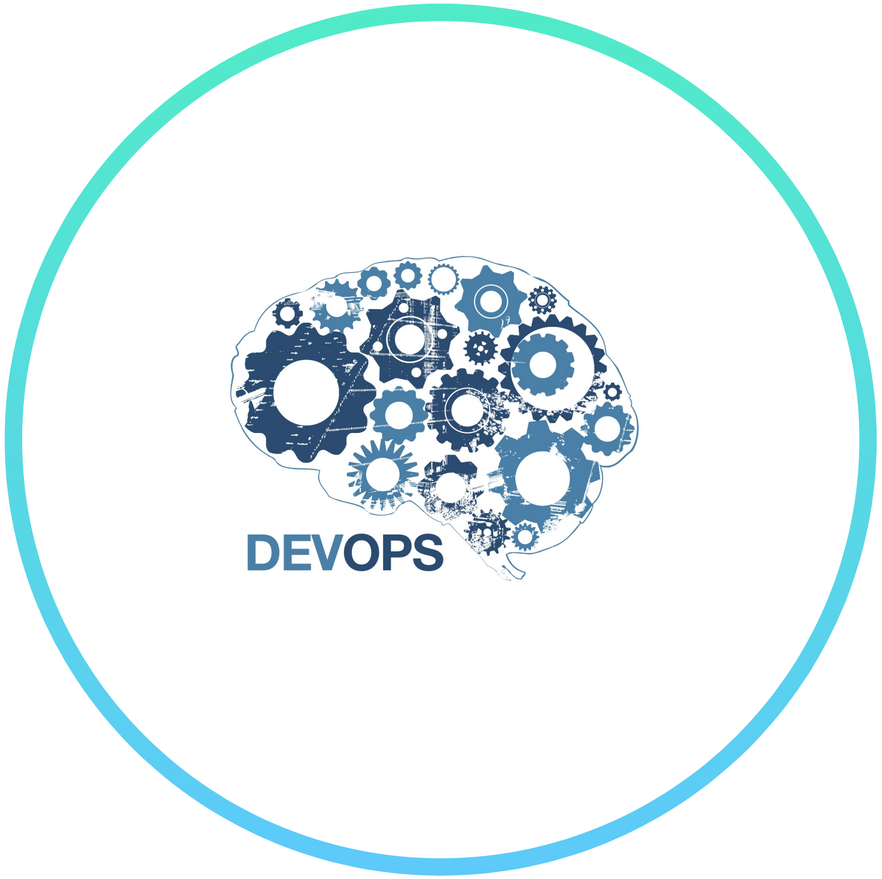
¿Cómo creamos un repositorio?



Es posible que no queramos que algunos ficheros
sean gestionados por el repositorio
para ello
podemos gestionar el fichero
.gitignore

Creación Repositorio

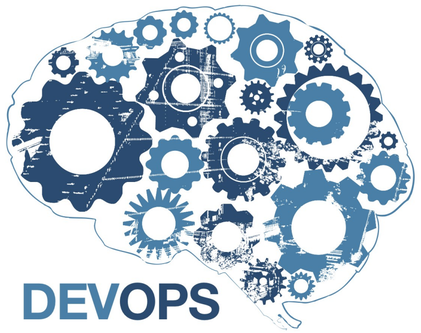
¿Cómo creamos un repositorio?



También podemos ver las distintas versiones delrepositorio:
git log

Creación Repositorio

¿Cómo creamos un repositorio?



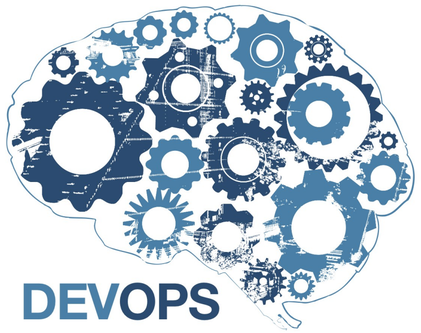
Con checkout podemos volver el proyecto a una determinada versión de commit:

git checkout <commit>

git checkout <commit> <file>

Creación Repositorio

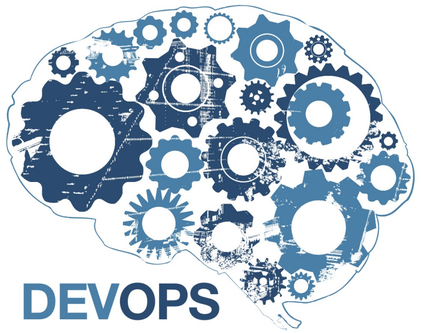
¿Cómo creamos un repositorio?



Con revert podemos volver atrás una serie de cambios en el proyecto con un nuevo commit:
git revert <commit>

Creación Repositorio

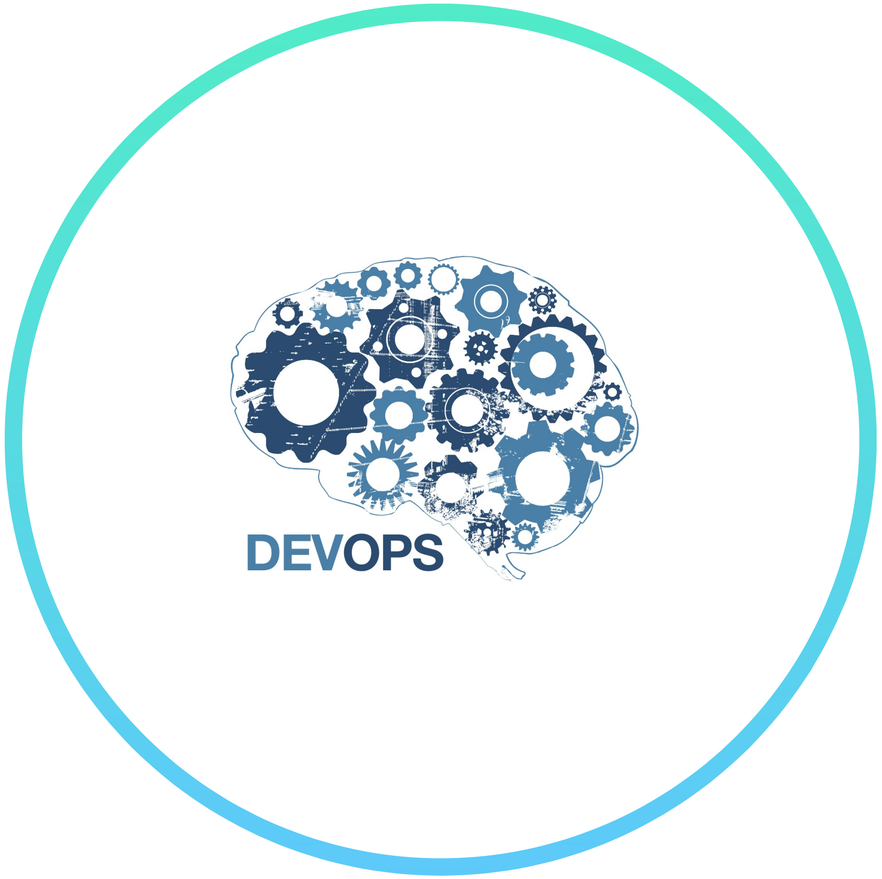
¿Cómo creamos un repositorio?

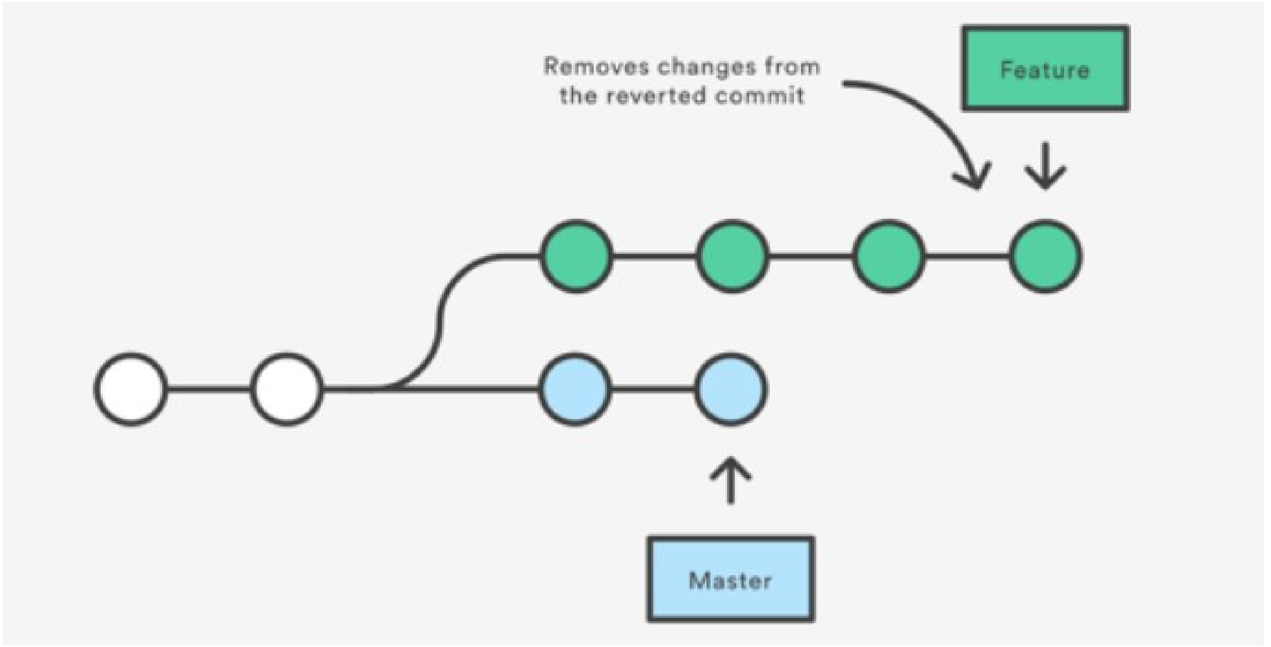


De esta manera mantenemos los commit anteriores y hacemos uno nuevo. Es decir, no borramos commits

Creación Repositorio

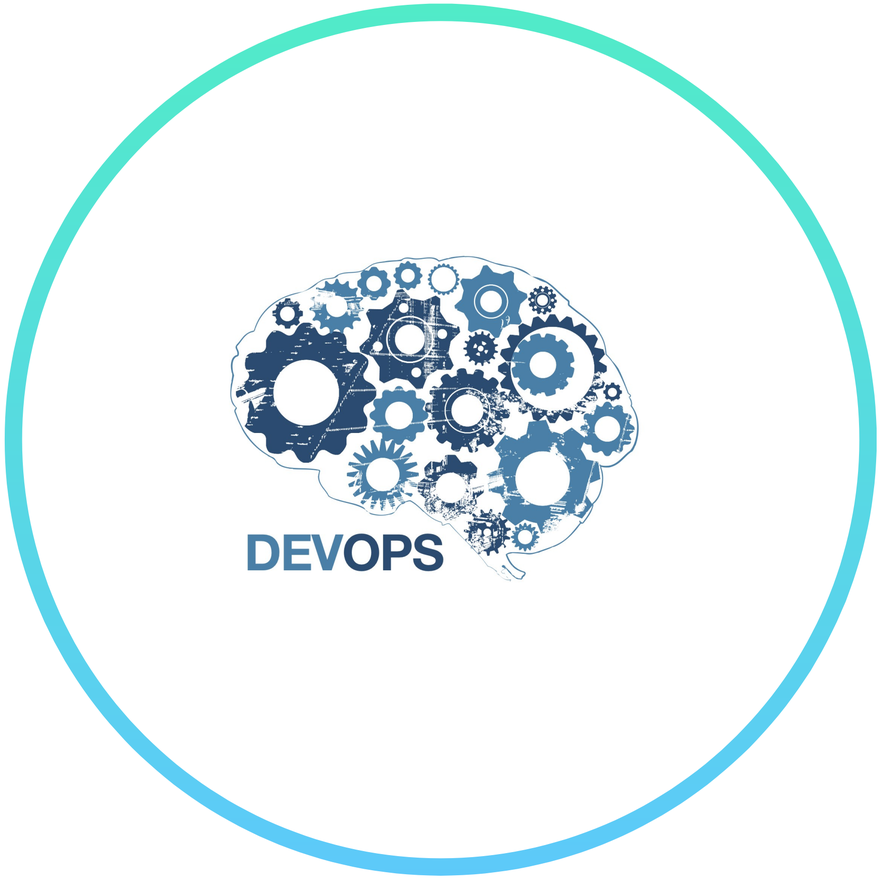
¿Cómo creamos un repositorio?





Creación Repositorio

¿Cómo creamos un repositorio?

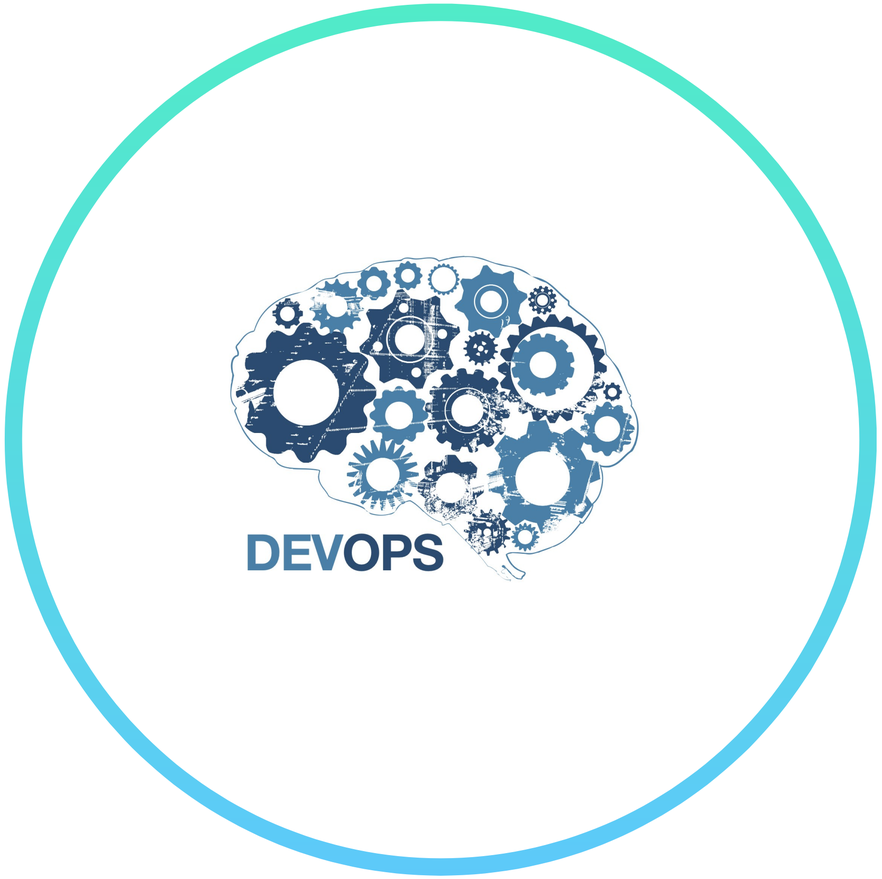


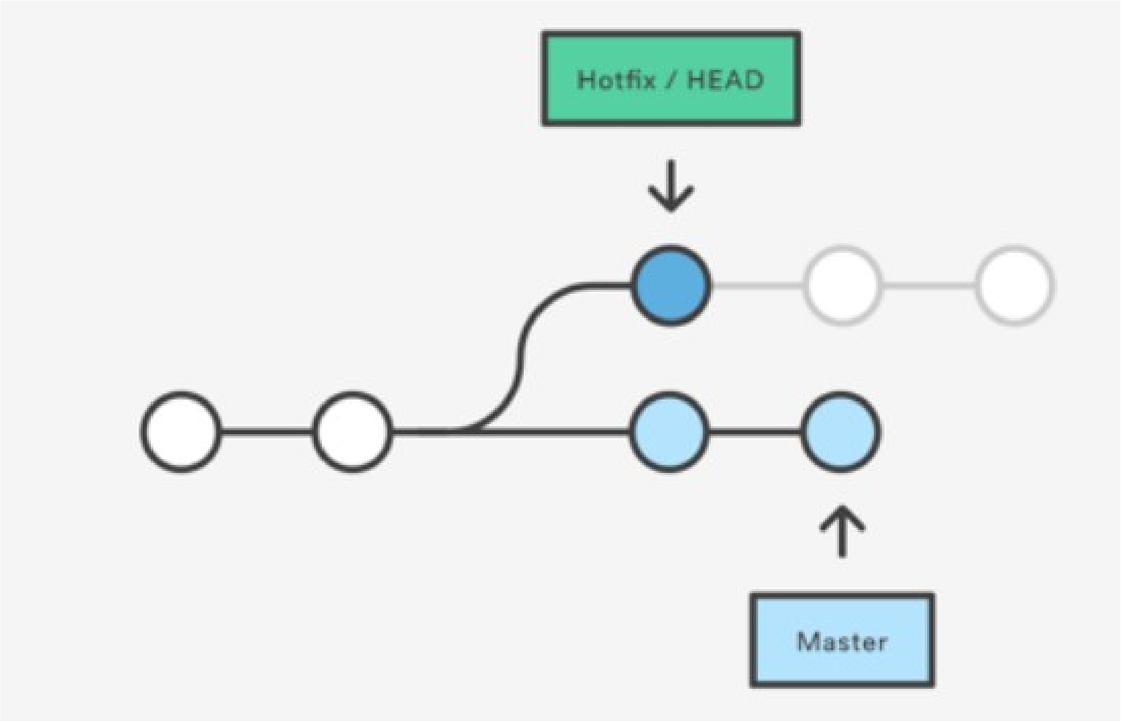
Con reset podemos volver atrás una serie de cambios borrando commits:

git reset <commit>

Creación Repositorio

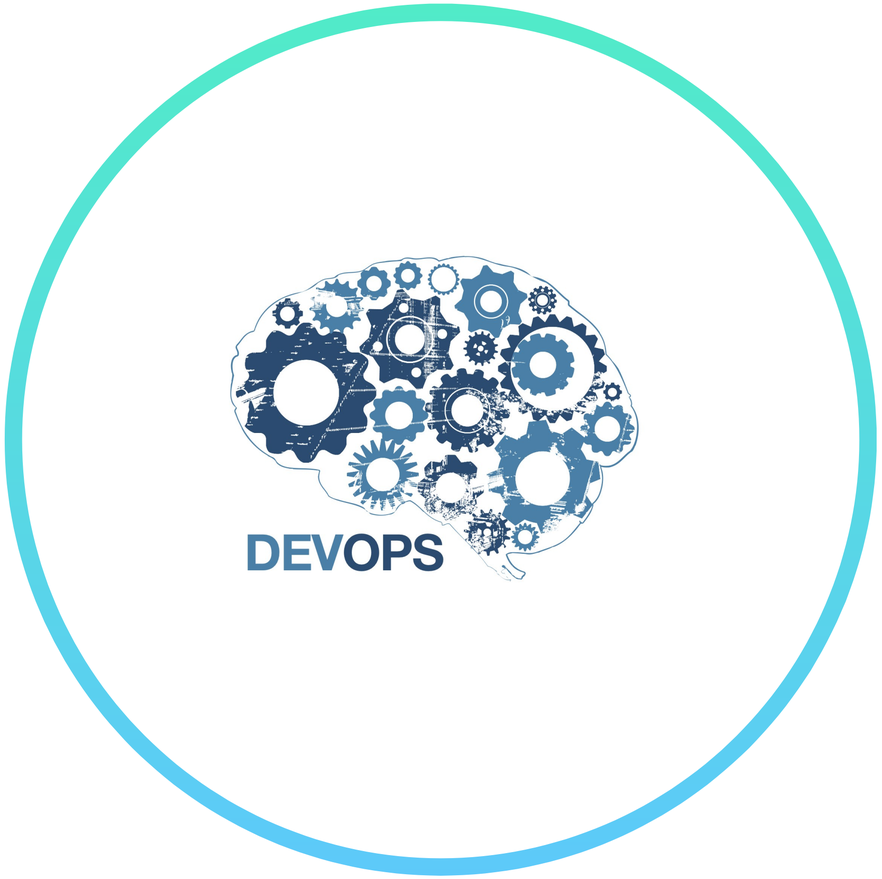
¿Cómo creamos un repositorio?





Creación Repositorio

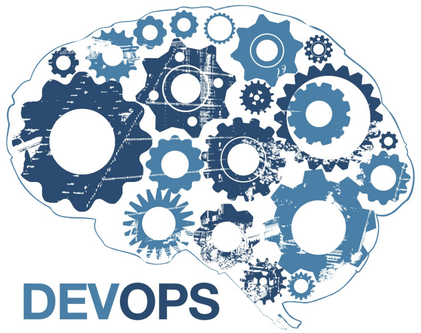
¿Cómo creamos un repositorio?



Command	Scope	Common use cases
`git reset`	Commit-level	Discard commits in a private branch or throw away uncommited changes
`git reset`	File-level	Unstage a file
`git checkout`	Commit-level	Switch between branches or inspect old snapshots
`git checkout`	File-level	Discard changes in the working directory
`git revert`	Commit-level	Undo commits in a public branch
`git revert`	File-level	(N/A)

Gui’s

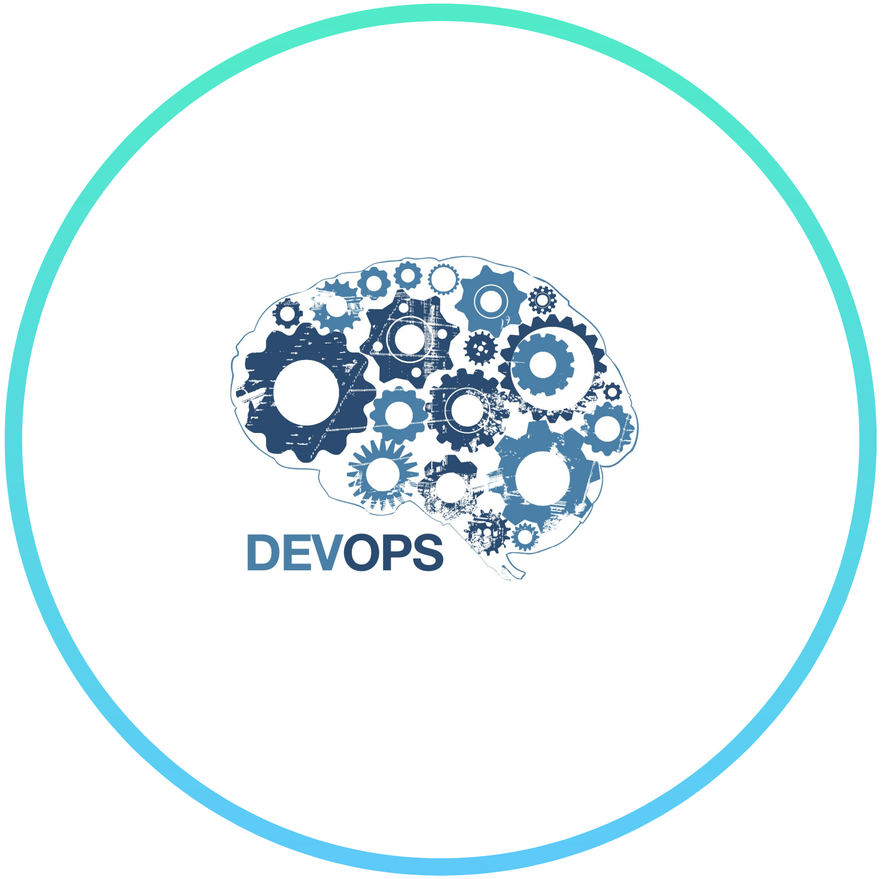
¿Cómo manejamos visualmente el repo?



Como esta gestión por consola suele ser algo
tediosa es preferible disponer de alguna herramienta
gráfica que ayude a esta gestión
https://git-scm.com/downloads/guis

Gui’s

¿Cómo manejamos visualmente el repo?

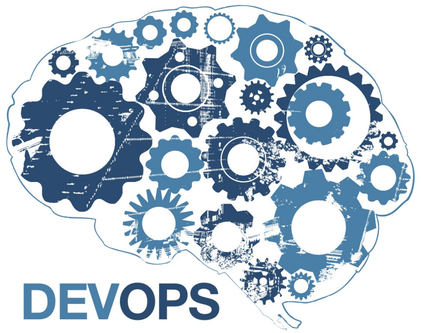


GitHub Desktop

https://desktop.github.com/

Gui’s

¿Cómo manejamos visualmente el repo?

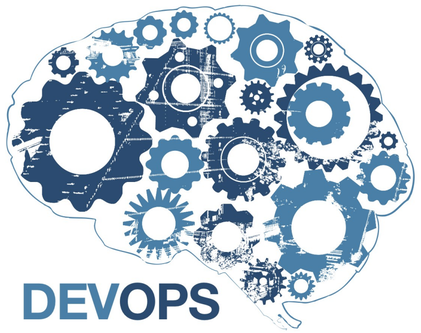


Sourcetree

https://www.sourcetreeapp.com/

Gui’s

¿Cómo manejamos visualmente el repo?

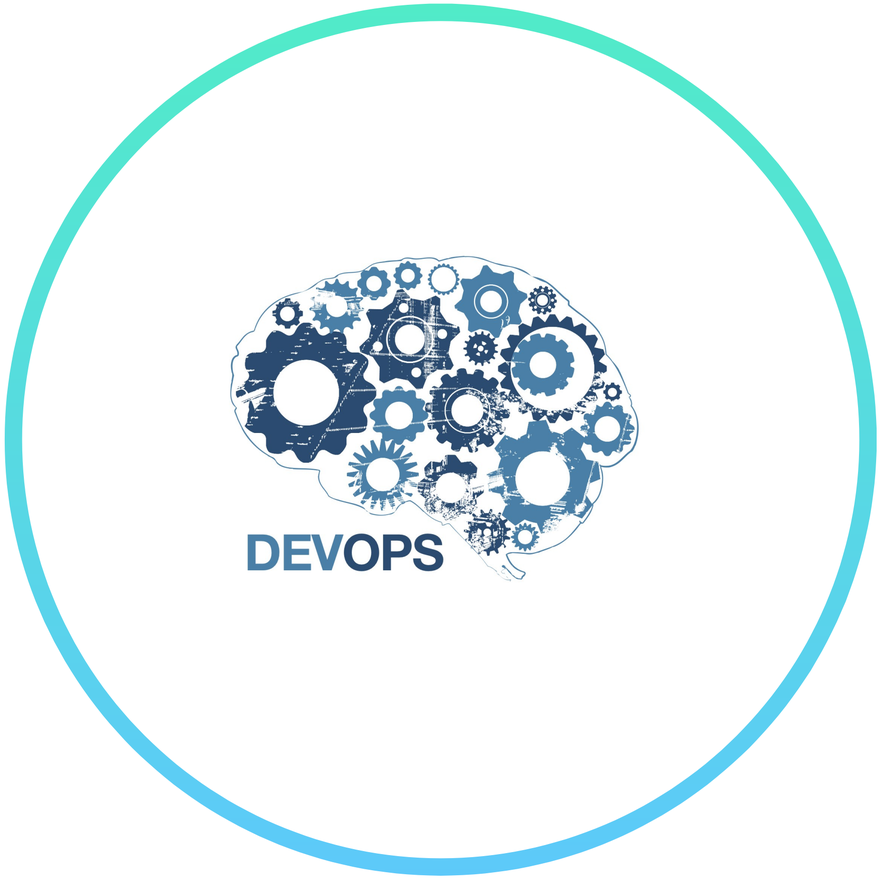


GitKraken

https://www.gitkraken.com/

Gui’s

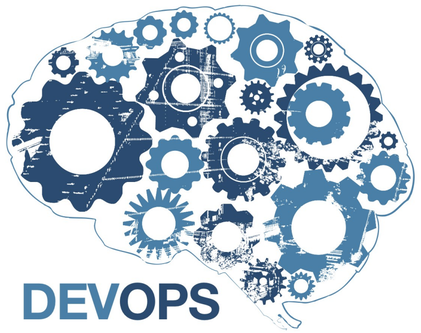
¿Cómo manejamos visualmente el repo?



La mayor parte de los IDE’s actuales disponen tanto de integración con Git como de Github

Github

¿Qué servicios tenemos para repos?

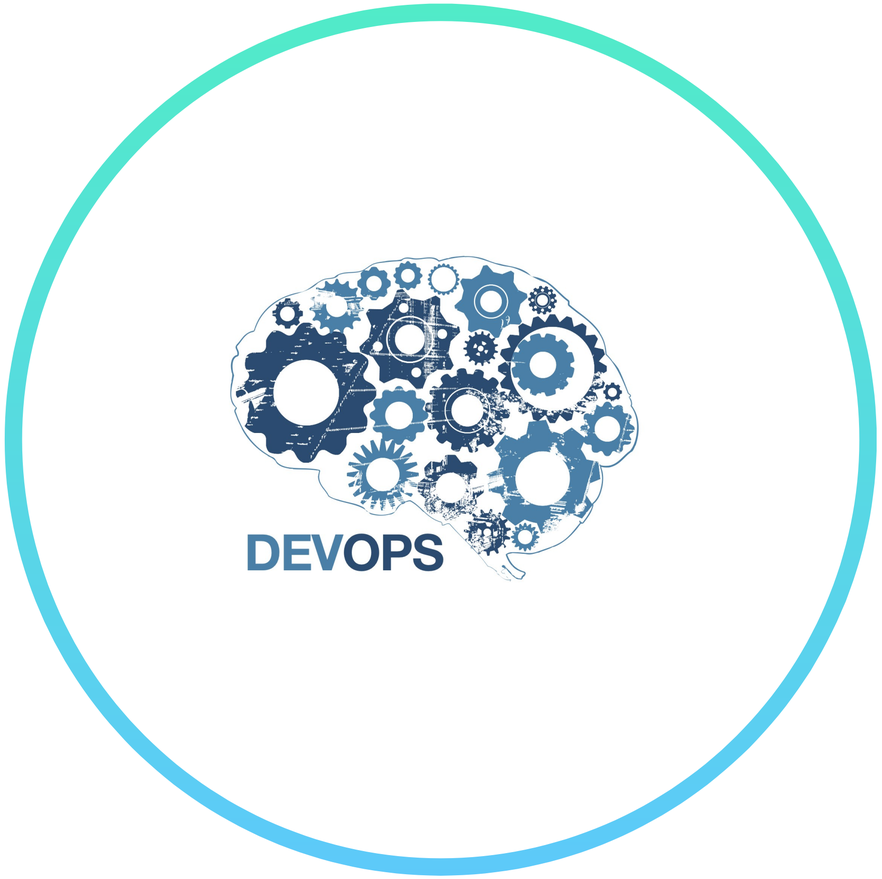


Github es un servicio disponible a través de la web en el que se pueden compartir repositorios Git con otras personas sin la necesidad de disponer de un ordenador encendido 24/7

https://github.com/

Github

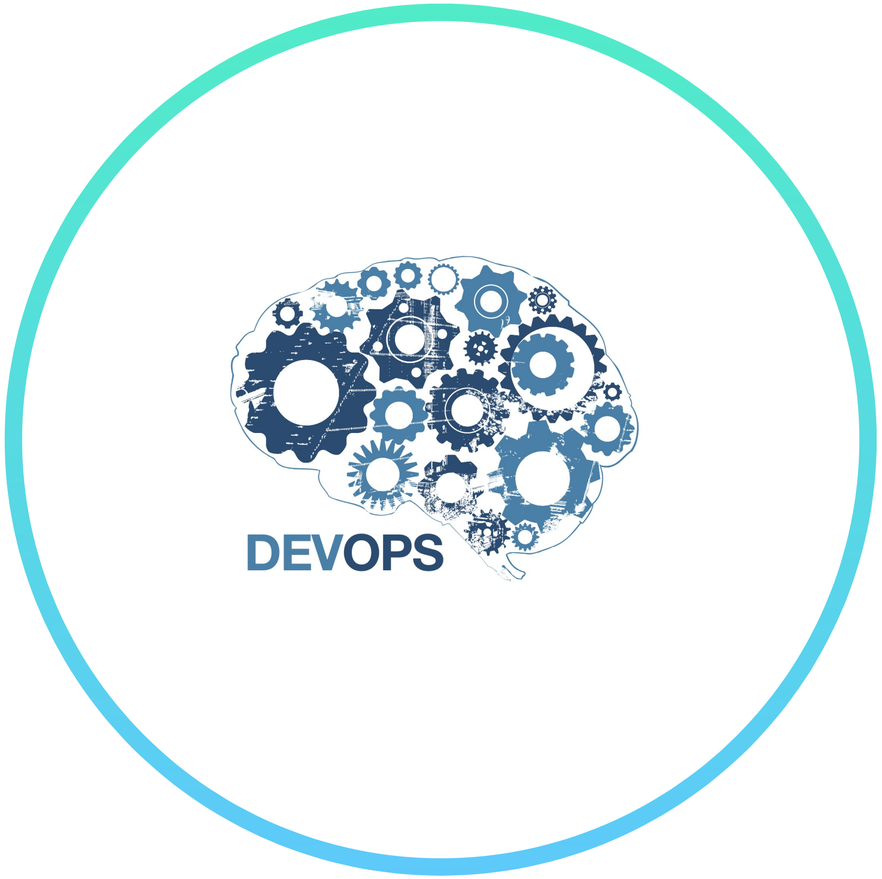
¿Qué servicios tenemos para repos?



Ejemplo de Repositorios abiertos:
https://github.com/pepesan

Gitlab

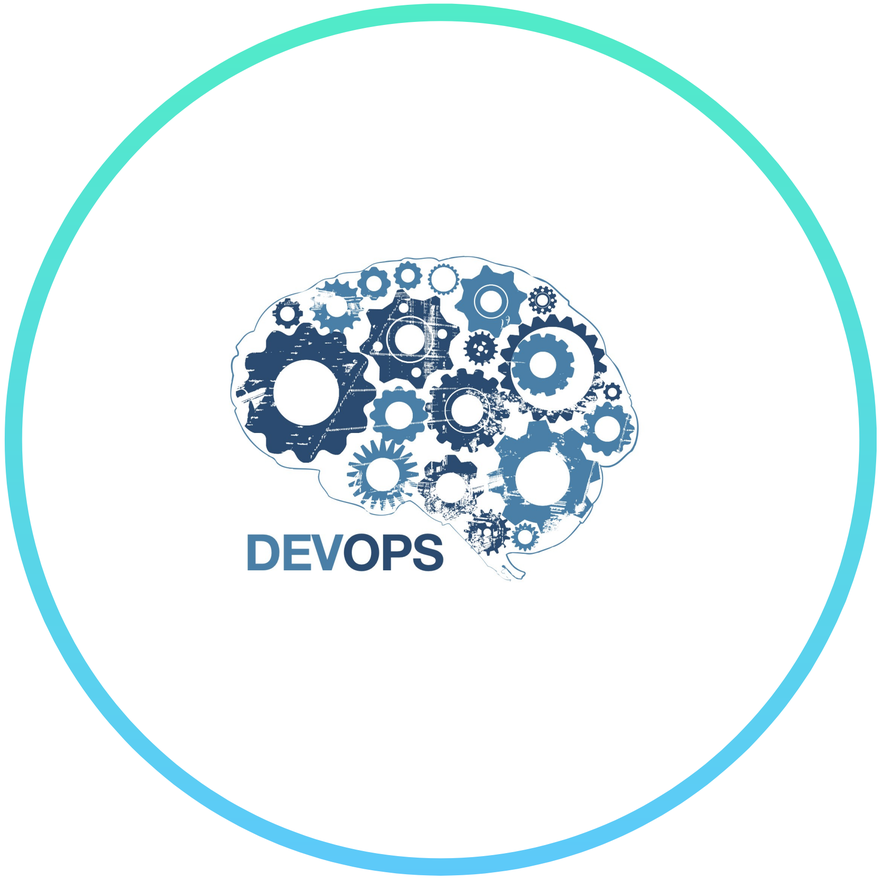
¿Qué servicios tenemos para repos?



Gitlab es la manera de disponer de tu propio Github instalado en tus propios servidores: https://about.gitlab.com/

Gitlab

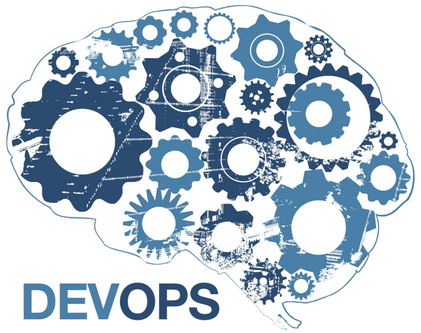
¿Qué servicios tenemos para repos?



La ventaja principal sobre GitHub es que tienes el control de los servidores y todo el contenido puede estar alojado dentro de tu empresa y no en los servidores de Github

Demo

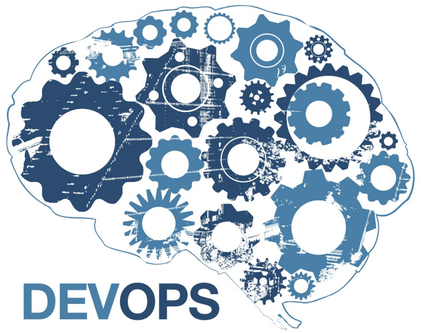
¿Veamos cómo va esto?



Show Time

Conclusiones

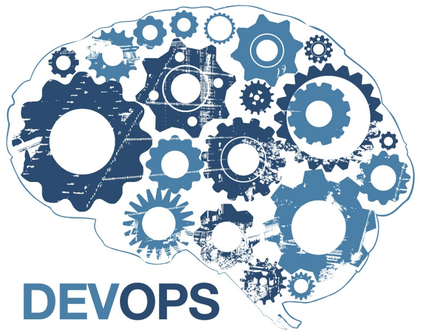
¿Qué podemos sacar en claro?



- Hemos visto cómo montar un servidor jenkins
- Hemos visto como importar un repositorio de código
- Hemos visto cómo integrar tareas a realizar en un proyecto
- Hemos visto cómo extraer informes de pruebas

Referencias

¿Fuentes de información?



Cosecha propia :)

CURSOS DE DESARROLLO
DAVID VAQUERO
LICENCIA CC-BY-SA-NC 4.0

info@cursosdedesarrollo.com

http://cursosdedesarrollo.com