

```
$ gitclick
```

Usage:

```
gitclick create [<repository>] [on <account>] [--no-issues] [--no-wiki] [--private]
```

Creates a public repository with issues and wiki by default.

<repository> default to the name of the current folder.

<account> default to the current account.

```
gitclick list
```

```
gitclick use <account>
```

```
gitclick add
```

```
gitclick remove <account>
```



```
gitclick -v, --version
```

Output version number

```
gitclick -h, --help
```

Output usage information

```
$ |
```

MIND  
HUB.

# ¿Qué es Git?

Git es un sistema de control de versiones distribuido, lo que significa que un clon local del proyecto es un repositorio de control de versiones completo. Estos repositorios locales plenamente funcionales permiten trabajar sin conexión o de forma remota con facilidad.

# Características de Git:

**Git realiza un seguimiento de cambios, no de versiones**

**Open Source, es decir es Gratis**

**Multiplataforma**

**Historia completa**

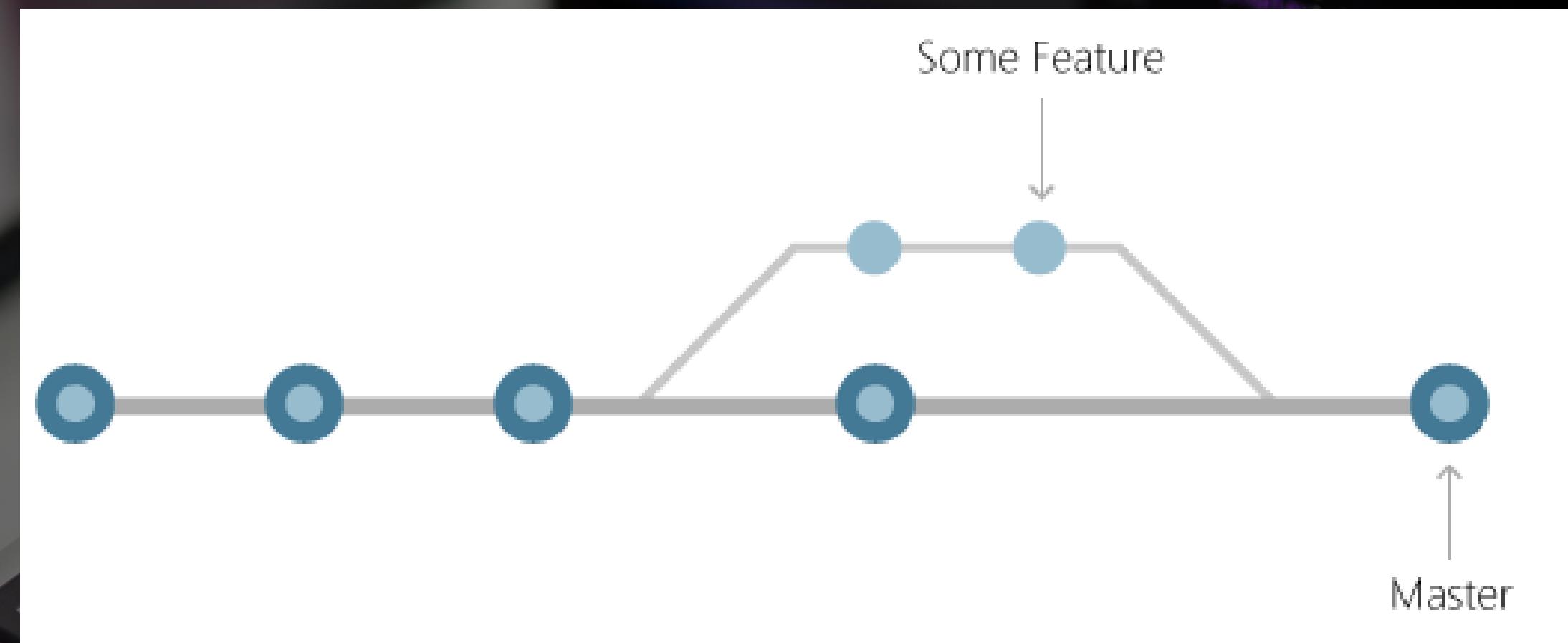
# Componentes de un proyecto GIT

- El directorio de trabajo
- El área de preparación.
- El directorio git



# Ramas GIT

Git proporciona herramientas para aislar los cambios y volver a combinarlos posteriormente. Las ramas, que son punteros ligeros para trabajar en curso, administran esta separación. Una vez finalizado el trabajo creado en una rama, se puede combinar de nuevo en la rama principal (o troncal) del equipo.



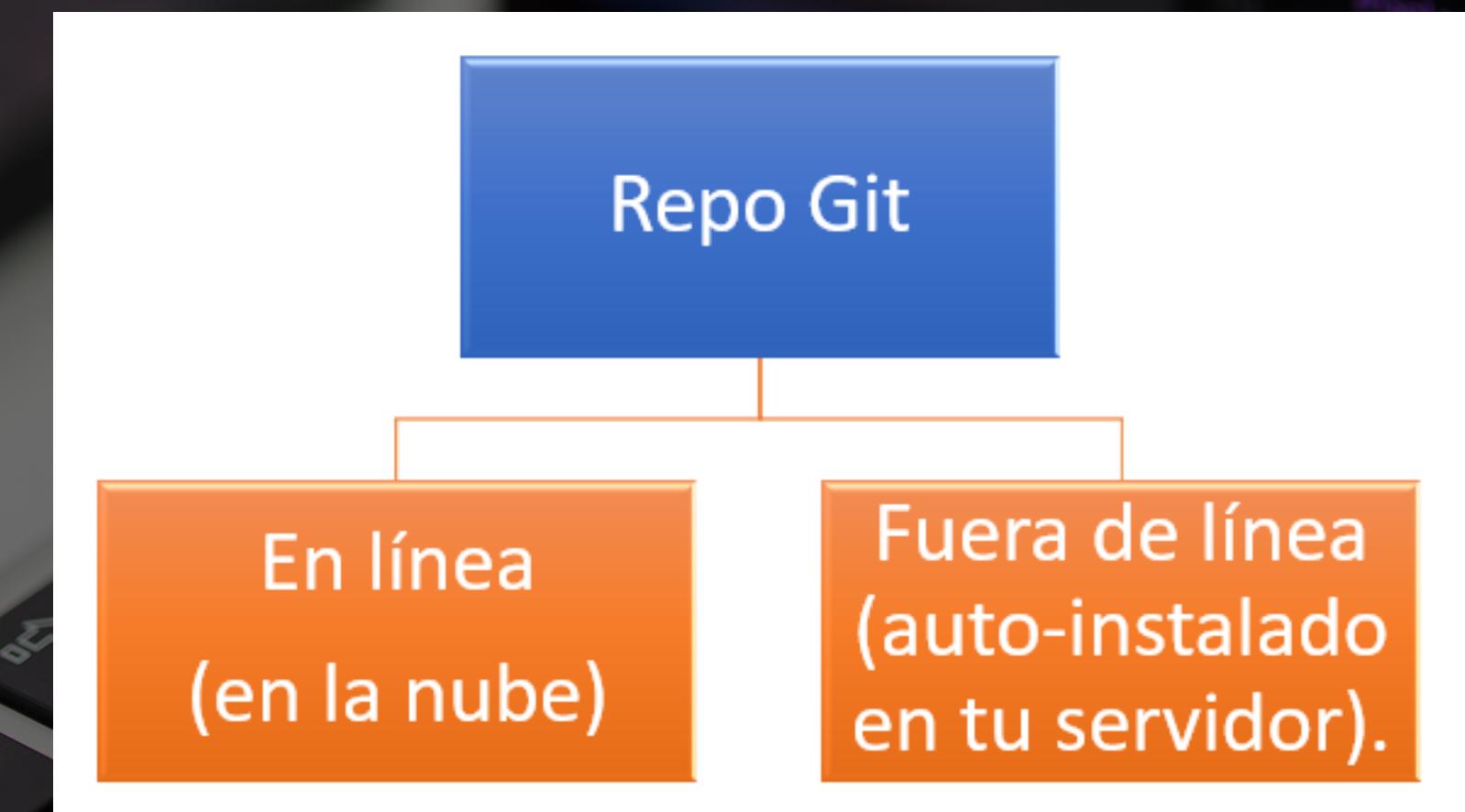
# Archivos y confirmaciones:

- Los archivos de Git se encuentran en uno de los tres estados: **Modificados**, **Almacenados provisionalmente** o **Confirmados**. Cuando se modifica un archivo por primera vez, los cambios solo existen en el directorio de trabajo. Todavía no forman parte de una confirmación o del historial de desarrollo. El desarrollador debe almacenar provisionalmente los archivos modificados que se incluirán en la confirmación. El área de almacenamiento provisional contiene todos los cambios que se van a incluir en la siguiente confirmación.
- El almacenamiento provisional permite a los desarrolladores elegir qué cambios de archivo guardar en una confirmación para desglosar los cambios grandes en una serie de confirmaciones más pequeñas. Al reducir el ámbito de las confirmaciones, es más fácil revisar el historial de confirmaciones para buscar cambios de archivo específicos.



# Repositorios GIT:

Git utiliza lo que llamamos "repositorios" o "Repo" si queremos abbreviar. Un Repositorio es un proyecto que contiene múltiples archivos.  
Existen dos maneras de alojar tus repositorios:



# Beneficios de utilizar GIT:

1. Git facilita la contribución a proyectos de código abierto
2. Documentación: Utilizando GitHub, facilitas la obtención de excelente documentación
3. Opciones de Integración: GitHub puede integrarse con plataformas comunes como Amazon y Google Cloud, con servicios como Code Climate para rastrear tus comentarios y puede resaltar la sintaxis en más de 200 lenguajes de programación diferentes.
4. Rastrea cambios en tu código a través de versiones



# Comandos basicos de Git:

Git init

Git status

Git log

Git add

Git commit

Git config

Git push

Git checkout

Git branch

Git merge

# Comandos basicos de Git:

Git diff

Git rm



# Ventajas y desventajas de GIT:



- Permite el control de versiones distribuido
- Permite trabajar sin conexión a Internet
- Es gratuito y de código abierto
- Ofrece un historial completo de cambios
- Facilita la colaboración en proyectos de equipo
- Es flexible y personalizable
- Permite la creación de ramas y fusiones de código



- Requiere conocimientos en línea de comandos
- Puede ser confuso para usuarios principiantes



# GitHub

# MIND HUB.

# ¿Qué es GitHub?

- GitHub es una compañía sin fines de lucro que ofrece un servicio de hosting de repositorios almacenados en la nube. Esencialmente, hace que sea más fácil para individuos y equipos usar Git como la versión de control y colaboración.
- La interfaz de GitHub es bastante fácil de usar para el desarrollador novato que quiera aprovechar las ventajas del Git. Sin GitHub, usar un Git generalmente requiere de un poco más de conocimientos de tecnología y uso de una línea de comando.
- GitHub es tan fácil de usar, que incluso algunas personas usan GitHub para administrar otro tipo de proyectos



# ¿Qué puedes hacer con GitHub?



Controlar las  
versiones de  
tus proyectos

Compartir tu  
código con  
otras personas

Trabajar en  
equipo

Tener un  
portafolio de  
trabajo

Tener soporte  
de tus  
proyectos  
privados

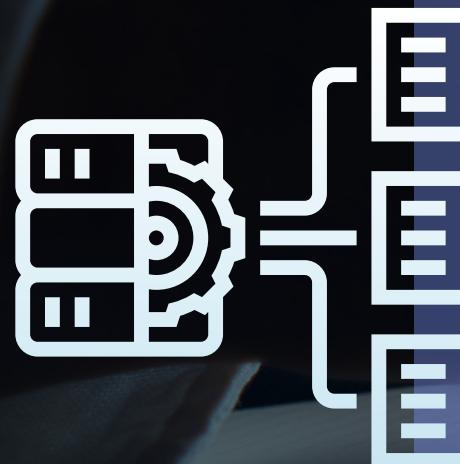
# Beneficios de utilizar GitHub



- Control de versiones
- Comunidad
- Integración con herramientas de terceros
- Almacenamiento en la nube
- Soporte para colaboración remota
- Seguridad

# Repositorio en GitHub

- En un repositorio de GitHub, se pueden encontrar todos los archivos necesarios para el proyecto, incluyendo el código fuente, documentación, archivos de configuración, recursos, imágenes y otros activos digitales. Los usuarios pueden colaborar en el mismo repositorio realizando modificaciones en los archivos y enviando solicitudes de cambio o "pull requests" para que el propietario del repositorio revise y apruebe los cambios.
- Los repositorios de GitHub también ofrecen herramientas para la gestión de problemas, la planificación de proyectos y la integración continua, lo que los convierte en una herramienta muy valiosa para los equipos de desarrollo de software y para proyectos de código abierto en general.



# Ventajas y desventajas de GitHub



- Es compatible con multiples sistemas operativos
- Proporciona alojamiento en la nube para proyectos
- Facilita la colaboracion en proyectos de equipos
- Ofrece herramientas adicionales para la gestion de proyectos
- Facilita la contribucion a proyectos de codigo abierto



- La mayoria de las funciones avanzadas son de pago
- Puede ser confuso para usuarios principiantes
- Depende de la conexion a internet

# GIT



# VS

# GIT HUB





Característica	Git	GitHub
Tipo de software	Git es un sistema de control de versiones distribuido (DVCS).	GitHub es un servicio en línea que se basa en Git.
Funcionalidad	Permite a los usuarios rastrear los cambios en el código fuente, fusionar y compartir diferentes versiones del código.	Ofrece una plataforma para alojar, compartir y colaborar en proyectos de software.
Instalación	Se puede descargar e instalar Git en cualquier sistema operativo, incluyendo Windows, Mac y Linux.	No necesita ser instalado, ya que es una plataforma en línea.
Uso	Se utiliza a través de la línea de comandos de la terminal o a través de una interfaz gráfica de usuario (GUI).	Se utiliza a través de una interfaz gráfica de usuario (GUI) o a través de la línea de comandos.
Propósito	Git es principalmente utilizado por desarrolladores de software para controlar y colaborar en el desarrollo de código fuente.	GitHub es utilizado por desarrolladores para alojar y compartir su código fuente y para colaborar en proyectos de software con otros desarrolladores.
Características adicionales	Git proporciona herramientas de línea de comando para realizar operaciones avanzadas de control de versiones, como la creación de ramas y la fusión de código.	GitHub ofrece características adicionales como seguimiento de problemas, gestión de proyectos y wikis.



# Referencias:

<https://learn.microsoft.com/es-es/devops/develop/git/what-is-git>

<https://open-bootcamp.com/aprender-programar/github>

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

MIND  
HUB.



Muchas Gracias!

MIND  
HUB.