



Repositorios:

Anderson Stiven Pulido, David Soto Rodriguez, Julian Steven Sanchez, Sebastian
Triana Rey. | Programación II | Fecha: 16/11/2021

¿QUÉ ES UN REPOSITORIO DE GIT?

Un repositorio de Git es un almacenamiento virtual de tu proyecto. Te permite guardar versiones del código a las que puedes acceder cuando lo necesites.

Para crear un nuevo repositorio, usa el comando `git init`. `git init` es un comando que se utiliza una sola vez durante la configuración inicial de un repositorio nuevo. Al ejecutar este comando, se creará un nuevo subdirectorio `.git` en tu directorio de trabajo actual. También se creará una nueva rama principal.

En este ejemplo, suponemos que ya cuentas con una carpeta de proyecto en la que quieres crear un repositorio. Primero deberás establecer el directorio de la carpeta raíz del proyecto con el comando `cd` y luego ejecutar `git init`.

```
cd /path/to/your/existing/code
git init
```

Apuntar `git init` a un directorio de proyecto existente hará que se ejecute la misma configuración de inicio mencionada arriba, pero en el ámbito de ese directorio.

```
git init <project directory>
```

CLONACIÓN DE UN REPOSITORIO EXISTENTE:

Si un repositorio ya se ha configurado en un repositorio central, el comando de clonación es la manera más común de obtener una copia de desarrollo local. Igual que `git init`, la clonación suele ser una operación única. Una vez que un desarrollador ha obtenido una copia de trabajo, todas las operaciones de control de versiones se administran por medio de su repositorio local.

```
git clone <repo url>
```

El comando `git clone` se usa para crear una copia o clonar un repositorio remoto. Se utiliza `git clone` con la URL de un repositorio. Git es compatible con varios protocolos de red y sus formatos de URL correspondientes. En este ejemplo, usaremos el protocolo Git SSH. Las URL Git SSH siguen la siguiente estructura: `git@HOSTNAME:USERNAME/REPONAME.git`.

Un ejemplo de una URL Git SSH sería el siguiente: `git@bitbucket.org:rhyolight/javascript-data-store.git`, donde los valores de la plantilla equivalen a:

- HOSTNAME: `bitbucket.org`
- USERNAME: `rhyolight`
- REPONAME: `javascript-data-store`

Al ejecutarlo, se extraerá la última versión de los archivos del repositorio remoto en la rama principal y se añadirá a una nueva carpeta. Esta carpeta nueva tendrá el mismo nombre que el repositorio (REPONAME), en este caso, `javascript-data-store`. La carpeta contendrá el historial completo del repositorio remoto y una rama principal recién creada.

GUARDAR DATOS EN EL REPOSITORIO:

Ahora que has iniciado o clonado un repositorio, puedes realizar commits en la versión del archivo. Para el siguiente ejemplo asumiremos que has configurado un proyecto en `/path/to/project`. Los pasos son los siguientes:

- Cambia el directorio a `/path/to/project`
- Crea un archivo nuevo `CommitTest.txt` con el contenido "test content for git tutorial"
- Ejecuta el comando `git add` para añadir `CommitTest.txt` al área de preparación del repositorio
- Crea un commit nuevo con un mensaje que describa qué trabajo se ha hecho en el commit

```
cd /path/to/project
echo "test content for git tutorial" >> CommitTest.txt
git add CommitTest.txt
git commit -m "added CommitTest.txt to the repo"
```

COLABORACIONES ENTRE REPOSITORIOS:

Es importante comprender que la idea de "copia de trabajo" de Git es muy distinta a la copia de trabajo que se obtiene al extraer código fuente de un repositorio SVN. A diferencia de SVN, Git no distingue entre las copias de trabajo y el repositorio central: todos son repositorios Git completos.

Por tanto, colaborar con Git es intrínsecamente distinto que con SVN. Mientras que SVN depende de la relación entre el repositorio central y la copia de trabajo, el modelo de colaboración de Git se basa en la interacción entre repositorios. En lugar de insertar una copia de trabajo en el repositorio central de SVN, se extraen o envían commits de un repositorio a otro.