

Git es un sistema de control de versiones de código abierto, diseñado para manejar grandes y pequeños proyectos con rapidez y eficiencia



## Configuración inicial

Abra su terminal de Git para comenzar con la ejecución de comandos, por ejemplo, abrirá el programa **Git bash** en Windows para ingresar a la línea de comandos de este programa.

Una vez que ingrese, use el siguiente comando para establecer el nombre de usuario de git:

```
git config—global user name “Johel Burgos”
```

Recuerde sustituir el texto entre comillas por su nombre real. Ahora indique el correo electrónico del usuario para git:

```
git config--global user email "micorreopersonal@johel.com"
```

Sustituyendo el texto entre comillas por su cuenta de correo electrónico. Esta configuración inicial debería ser suficiente para comenzar. Para comprobar otros valores de su configuración actual ejecute:

```
git config --list
```

Se mostrarán los nuevos valores configurados al final, y otros valores de configuración predeterminados:

```
...
```

```
color.diff=auto
```

```
color.status=auto
```

```
...
```

```
user.name=Johel Burgos
```

```
user.email=micorreopersonal@johel.com
```

## 1. Primeros pasos

### Inicializar repositorio git vacío

Lo primero que debemos hacer es navegar a la carpeta donde vamos a almacenar los archivos y subdirectorios para nuestro repositorio local. Podemos hacerlo con el comando `cd` de Linux y presionando la tecla **TAB** para ir autocompletando los directorios en cuestión. Para este ejemplo particular usaremos la siguiente ubicación:

```
cd Desktop/MiRepo/
```

Una vez allí, ingrese el siguiente comando:

```
git init
```

Este comando inicializa un repositorio git vacío en la carpeta en cuestión.



The screenshot shows a terminal window titled "MINGW32:/c:/Users/Johel/Desktop/MiRepo". The user is at the prompt "Johel@JOHEL ~". They enter the command "\$ cd Desktop/MiRepo/", which is autocompleted to "\$ cd Desktop/MiRepo/". The prompt changes to "Johel@JOHEL ~/Desktop/MiRepo". They then enter "\$ git init", which results in the output "Initialized empty Git repository in c:/Users/Johel/Desktop/MiRepo/.git/". The prompt changes to "Johel@JOHEL ~/Desktop/MiRepo (master)". A blue arrow points to the "(master)" part of the prompt. At the bottom, there is a blue text annotation: "Así inicializamos un repositorio git vacío".

```
MINGW32:/c:/Users/Johel/Desktop/MiRepo
Johel@JOHEL ~
$ cd Desktop/MiRepo/
Johel@JOHEL ~/Desktop/MiRepo
$ git init
Initialized empty Git repository in c:/Users/Johel/Desktop/MiRepo/.git/
Johel@JOHEL ~/Desktop/MiRepo (master)
$
```

Así inicializamos un repositorio git vacío

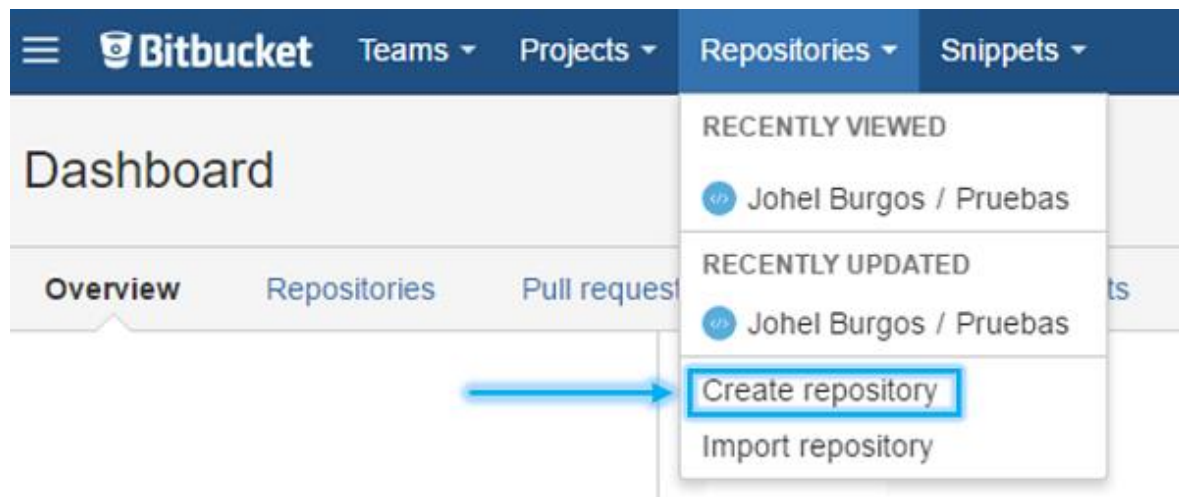
### Crear un repositorio remoto

Ahora debe seleccionar la plataforma de desarrollo colaborativo donde alojará sus proyectos de Git. La plataforma que seleccione le permitirá un control preciso y sencillo de sus proyectos y le proveerá un repositorio remoto para sus archivos. Entre las más utilizadas se encuentran [GitHub](#) y otras alternativas gratuitas como [Bitbucket](#) y [GitLab](#).

Para nuestra guía utilizaremos **Bitbucket** como ejemplo. Ingrese en el sitio web de [Bitbucket](#) y haga clic en **Get Started** para crear una cuenta siguiendo las instrucciones allí expuestas.

Una vez que haya establecido los datos de su cuenta de Bitbucket, vuelva a ingresar al sitio web y haga clic en **Log In** para iniciar sesión. Luego entrará al escritorio de administración de Bitbucket. Para crear un nuevo repositorio:

1. Haga clic en **Repositories** -> **Create repository** en el menú superior de la página de Bitbucket.



2. Se mostrará un cuadro con los campos para crear un nuevo repositorio. Seleccione el nombre deseado para el repositorio, el propietario, el nivel de acceso (mantenga seleccionada esta casilla si quiere que su repositorio sea privado) y el tipo de repositorio, en nuestro caso, **Git**.  
*Nota:* El nombre que asigne al repositorio afectará su URL de su acceso. En nuestro ejemplo la dirección del repositorio estará dada por: `https://bitbucket.org/miusuario/mirepo`

3. Para completar el proceso haga clic en el botón **Create Repository**:

The screenshot shows the 'Create a new repository' interface. At the top right is a link 'Import repository'. The form fields are: 'Owner' with a dropdown menu showing 'miusuario'; 'Repository name' with a text input containing 'MiRepo'; 'Access level' with a checked checkbox 'This is a private repository'; and 'Repository type' with radio buttons for 'Git' (selected) and 'Mercurial'. Below these is a link '> Advanced settings'. At the bottom are two buttons: 'Create repository' (highlighted with a blue arrow) and 'Cancel'.

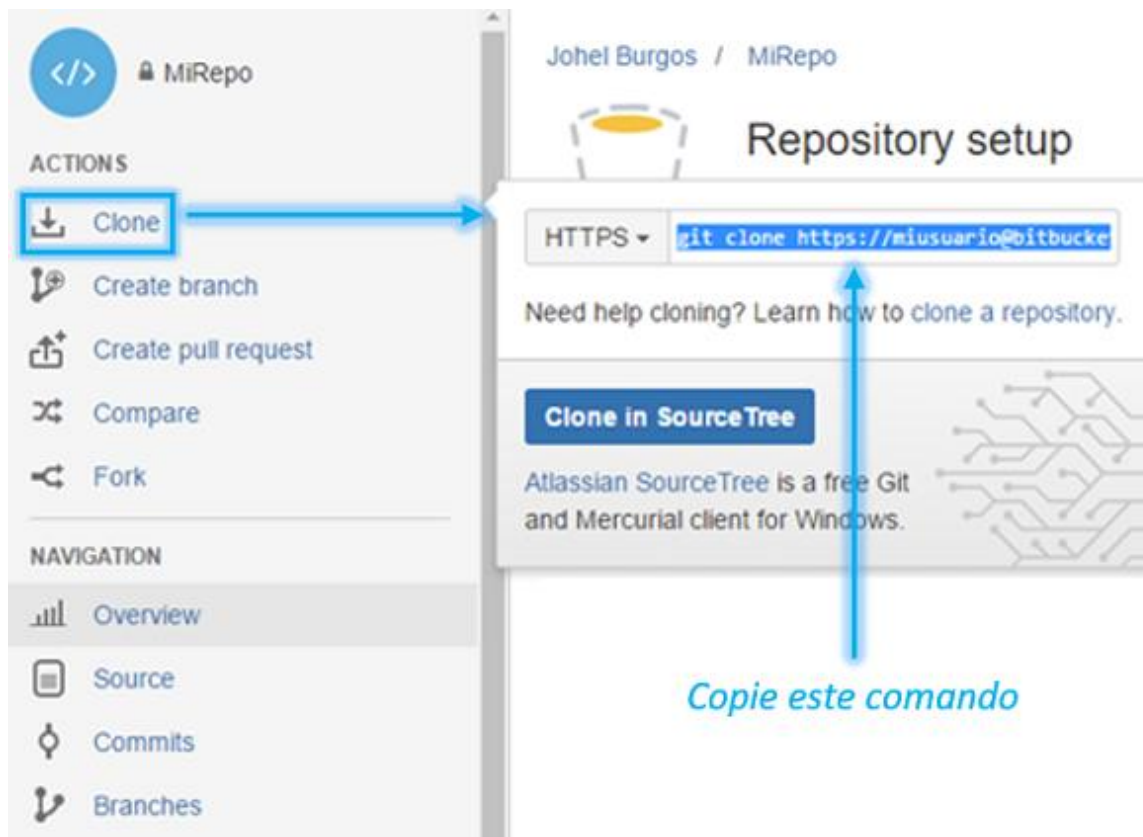
4. ¡Esto es todo, ya ha creado un nuevo repositorio remoto en Bitbucket!

### Uso básico

#### Clonar repositorio

Es momento de poblar nuestro repositorio remoto con un archivo. En el lenguaje **Git** nos referimos al repositorio remoto como el **origin** (u *origen*) y a nuestro repositorio local como **local**. Para comenzar debemos clonar el repositorio remoto en nuestro *repo* local. Para hacerlo, seleccione el repositorio creado bajo la sección **Repositories** del menú superior.

Haga clic en **Clone** bajo la sección "ACTIONS" del panel izquierdo y **copie** el comando allí mostrado:



Para nuestro ejemplo hemos copiado el comando

```
git clone https://miusuario@bitbucket.org/miusuario/mirepo.git
```

Pegue y ejecute este comando en su consola **Git**. No olvide hacerlo desde el directorio en el cual desea clonar el repositorio en nuestro ejemplo `Desktop/MiRepo`. Al ejecutar el comando, se le pedirá la contraseña asociada con su nombre de usuario de Bitbucket, ingrésela y presione *Enter*. Si la clonación es exitosa se mostrará algo similar a lo siguiente:

```
Cloning into 'mirepo'...
```

```
Password for 'https://miusuario@bitbucket.org':
```

```
warning: You appear to have cloned an empty repository
```

```
Checking connectivity... done.
```

**Crear un archivo inicial y comprobar el estatus del proyecto**

Hasta ahora tenemos un repositorio vacío local y remoto. Puede utilizar el siguiente comando para crear un nuevo archivo de prueba:

```
echo "#Título de mi primer archivo Contenido de mi primer archivo" >> prueba.md
```

Si el comando no devuelve nada, quiere decir que se ejecutó correctamente.

Uno de los comandos más útiles y frecuentes que utilizaremos en Git es `git status`. El uso simple de este comando indicará el progreso de su proyecto en comparación al repositorio en Bitbucket. En este punto, Git indicará que ya creó un archivo:

```
git status
```

Resultado del comando

```
On branch master
```

```
Initial commit
```

```
Untracked files:
```

```
(use "git add <file>..." to include in what will be committed)
```

```
prueba.md
```

```
test.txt
```

```
nothing added to commit but untracked files present (use "git add" to track)
```

Como podemos observar, el comando `git status` nos indica el estatus de nuestro proyecto y nos da consejos sobre cómo continuar.

Cuando usted administra un proyecto más grande, este comando será útil para comprender cuáles archivos se han creado, añadido al área de andamiaje, borrado o encomendado para su entrega.

## Añadir el archivo al repositorio remoto y enviarlo a Bitbucket

Indíquele a Git que haga seguimiento de su nuevo archivo `prueba.txt` usando el comando:

```
git add prueba.md
```

En este momento, usted está en el área de *staging* o andamiaje.

Si ejecuta el comando `git status` ahora, podrá ver que el nuevo archivo ha sido añadido; llega entonces el momento de hacer el **commit** de los archivos añadidos.

Para encomendar su archivo o hacer un *commit* utilice el comando `git commit` seguido del parámetro `-m`. Después de este parámetro coloque un mensaje entre comillas indicando qué está por enviar a Bitbucket. Por ejemplo:

```
git commit-m "mi primer commit"
```

El comando `git commit` toma la foto andamiada y la encomienda al historial del proyecto. Combinado con `git add` este proceso define el flujo de trabajo básico para los usuarios de Git.

Todo lo que ha realizado hasta este punto, es en su sistema local y es –por lo tanto- invisible a su repositorio de BitBucket hasta que envíe o haga **push** de esos cambios.

Vuelva a su ventana terminal y envíe los cambios encomendados a bickbucket usando el comando que sigue:

```
git push origin master
```

Este comando especifica que usted está “empujando” el contenido hacia la rama master en (la rama principal del *repo* de BitBucket) en el origin (servidor de Bitbucket). Al ejecutar este comando se solicitará la contraseña asociada con su cuenta de Bitbucket.