

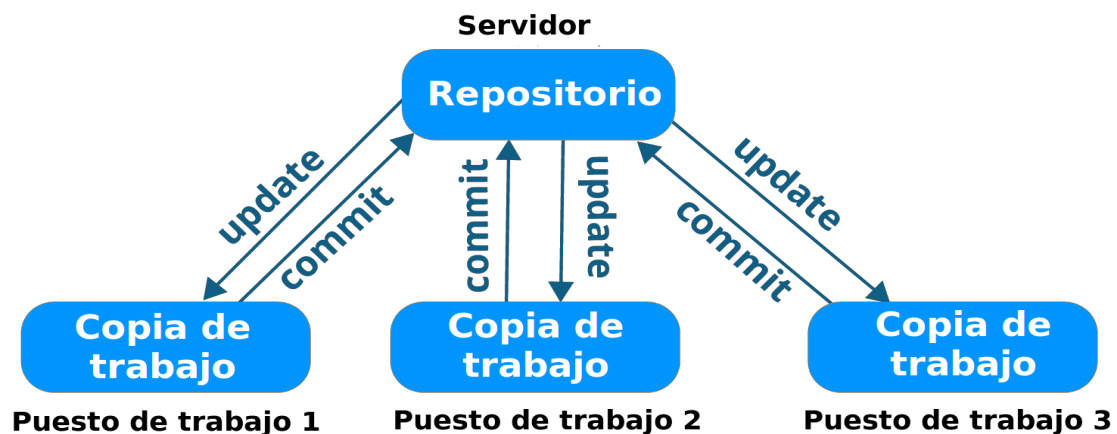
# CONTROL DE VERSIONES

El control de versiones sirve para la gestión de cambios en documentos, código fuente de un software, o cualquier otro tipo de ficheros. Hay dos tipos de sistemas de control de versiones:

- Sistemas de control de versiones centralizados.
- Sistemas de control de versiones distribuidos.

## Sistema de control de versiones centralizado (por ejemplo SVN)

Un sistema de **control de versiones centralizado (SCVC)** utiliza un servidor central para almacenar todos los archivos y permite el trabajo colaborativo de un equipo. Trabaja sobre **un repositorio único** al que los usuarios pueden acceder desde un servidor central. La siguiente imagen puede clarificar el concepto de SCVC:



El repositorio indica un servidor central que está directamente conectado al puesto de trabajo de cada programador. Todos los programadores pueden actualizar (**update**) sus puestos de trabajo con los datos presentes en el repositorio o pueden hacer cambios (**commit**) en los mismos. Cada operación se realiza directamente en el repositorio.

A pesar que parece más conveniente mantener un solo repositorio, tiene algunos inconvenientes importantes:

- No está **disponible localmente**, por lo que siempre es necesario estar conectado a una red para realizar cualquier acción.
- Como todo está centralizado, si el servidor se rompe **provocará la pérdida completa** del proyecto.

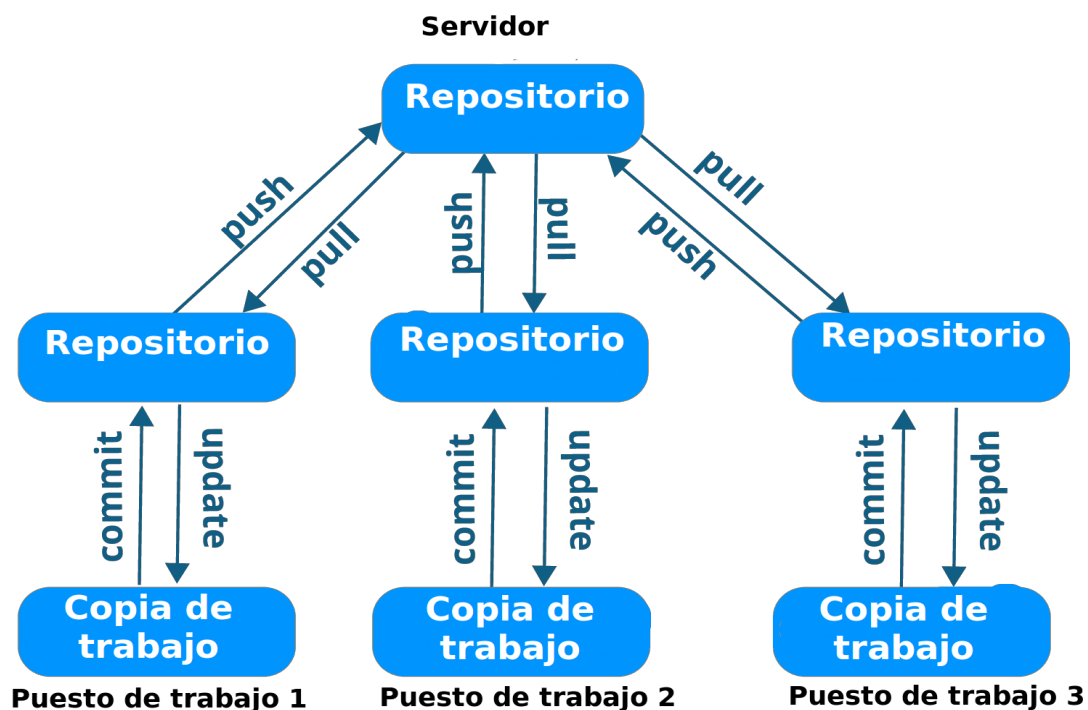
Los sistemas de control de versiones distribuidos **resuelven estos inconvenientes**.

### Sistema de control de versiones distribuido (por ejemplo Git)

Los sistemas de **control de versiones distribuidos (en adelante SCVD)** no dependen necesariamente de un servidor central para almacenar las versiones de los ficheros del proyecto.

En los SCVD cada programador tiene **una copia local o clon** del repositorio principal. Esto quiere decir que cada programador mantiene un repositorio local propio que contiene todos los archivos y metadata presente en el repositorio principal. Todos pueden operar con su repositorio local sin ninguna interferencia.

La siguiente imagen puede clarificar el concepto de SCVD:



Todos los programadores pueden actualizar sus repositorios locales con nuevos datos del servidor central con una operación llamada **'pull'** y persistir cambios en el repositorio principal con una operación llamada **'push'** desde su repositorio local. El hecho de clonar un repositorio entero en tu propio puesto de trabajo para tener un repositorio local proporciona las siguientes ventajas:

- Todas las operaciones (excepto push y pull) son **muy rápidas** porque la herramienta sólo necesita acceder al disco duro, no a un servidor remoto. Por tanto, no siempre se necesita conexión a internet.
- Los nuevos cambios se pueden guardar (commit) localmente **sin manipular** los datos del repositorio principal. Una vez se tenga listo un conjunto de cambios, se pueden persistir (push) todos a la vez en el repositorio principal.
- Dado que cada programador tiene una copia completa del repositorio del proyecto, pueden **compartir los cambios** entre sí si fuera necesario obtener un feedback antes de persistir los cambios en el repositorio principal.

- Si el servidor central sufre algún percance en algún momento, los datos perdidos pueden ser recuperados fácilmente desde cualquiera de los repositorios locales de los colaboradores.

## TRABAJO CON GIT

### Creación de cuenta y repositorio remoto en GitHub

[https://github.com/signup?](https://github.com/signup?ref_cta=Sign+up&ref_loc=header+logged+out&ref_page=%2F&source=header-home)

[ref\\_cta=Sign+up&ref\\_loc=header+logged+out&ref\\_page=%2F&source=header-home](https://github.com/signup?ref_cta=Sign+up&ref_loc=header+logged+out&ref_page=%2F&source=header-home)

Welcome to GitHub!  
Let's begin the adventure

**Enter your email**  
✓ `echacon@formacionprofesionalccc.com`



**Create a password**  
✓ ••••••••

**Enter a username**  
→

You're almost done!  
We sent a launch code to  
`echacon@formacionprofesionalccc.com`

→ Enter code

Didn't get your email? [Resend the code](#) or [update your email address](#).



# Welcome to GitHub

We are glad you're here.

**How many team members will be working with you?**

This will help us guide you to the tools that are best suited for your projects.


**Are you a student or teacher?**

## Crear nuevo repositorio

Profile => Your repositories => New

Owner \*

Repository name \*

 echacon32 ▾


 / 

primero ✓


Great repository names are short and memorable. Need inspiration? How about [glowing-doodle](#)?

Description (optional)

---

☒  Public

Anyone on the internet can see this repository. You choose who can commit.

☐  Private

You choose who can see and commit to this repository.

---

Initialize this repository with:

Skip this step if you're importing an existing repository.

☒ Add a README file

This is where you can write a long description for your project. [Learn more.](#)

Add .gitignore

Choose which files not to track from a list of templates. [Learn more.](#)





.gitignore template: None ▾

Choose a license


A license tells others what they can and can't do with your code. [Learn more.](#)


License: None ▾


echacon32 / primero Public

 Pin  Unwatch 1  Fork 0  Star 0

[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#) [Settings](#)


 main ▾


 1 branch

 0 tags


Go to file


Add file ▾

 Code ▾

 echacon32 Initial commit

1e8f96 10 seconds ago


 1 commit

 README.md

Initial commit


10 seconds ago


README.md





primero


About

 No description, website, or topics provided.

 Readme

 0 stars

 1 watching

 0 forks

Releases

No releases published

[Create a new release](#)


## Instalación de Git en el ordenador


<https://git-scm.com/book/es/v2/Inicio---Sobre-el-Control-de-Versiones-Instalaci%C3%B3n-de-Git>


## Instalación de SourceTree en el ordenador (cliente gráfico)


<https://www.sourcetreeapp.com/>


## Clonado del repositorio

 Local

 Remote

 Clone


 Add

 Create

# Clone

Cloning is even easier if you set up a remote account

Browse

Repository Type:  This is a Git repository

Browse

Local Folder:

> Advanced Options

Clone