

Herramientas de Desarrollo

Semana 02

**Instalación, configuración y comandos básicos
de solución de control de versiones**



**Universidad
Tecnológica
del Perú**

Inicio

¿Tienen alguna consulta o duda sobre la clase anterior?



Logro de la Unidad

Al finalizar la unidad, el estudiante gestiona los sistemas de control de versiones para el desarrollo de una solución de software.



Utilidad

- ¿Si necesitamos retroceder en nuestro programa, a un punto anterior, como podemos determinar hasta donde sería?

contenido

1 Introducción

2 Instalación

3 Configuración

4 Comandos Básicos

Transformación

1 Introducción

Introducción

Git es el sistema de control de versiones más utilizado en el mundo del desarrollo de software.

A continuación, se describen los pasos para su instalación y configuración básica, seguido de ejemplos de comandos comunes.

2 Instalación

Instalación de Git



En Windows:

Descarga el instalador de Git desde [Git for Windows](#).

Ejecuta el archivo descargado y sigue las instrucciones del instalador. Asegúrate de marcar la opción de añadir Git a tu PATH para usarlo desde la línea de comandos.

En macOS:

Puedes instalar Git mediante Homebrew:

```
brew install git
```

O bien, puedes descargar el instalador de [Git for macOS](#).

En Linux:

En distribuciones basadas en Debian (como Ubuntu), puedes instalar Git con:

```
sudo apt-get update
```

```
sudo apt-get install git
```



[Click here to download](#) the latest (2.41.0) 64-bit version of Git for Windows. This is the most recent maintained build. It was released 29 days ago, on 2023-06-01.

Standalone Installer

32-bit Git for Windows Setup.

64-bit Git for Windows Setup.

Portable ("thumbdrive edition")

32-bit Git for Windows Portable.

64-bit Git for Windows Portable.

Using winget tool

Install **winget** tool if you don't already have it, then type this command in command prompt or Powershell.

```
winget install --id Git.Git -e --source winget
```

*The amount cannot be reduced to 0. If you want the maximum, you can build it from the

<https://www.youtube.com/watch?v=wJnPN-x80FM>

3 Configuración

Configuración Básica de Git

Una vez instalado, es importante configurar Git con tu nombre y correo electrónico, que se asociarán con tus commits.

Configurar tu nombre y correo electrónico:

```
git config --global user.name "Tu Nombre"
```

```
git config --global user.email "tuemail@ejemplo.com"
```

Verifica la configuración:

```
git config --global --list
```



```
Archivo Editar Buscar Vista Codificación Lenguaje Configuración Herramientas Menú Ejecutar Plugins Ventana ?
1 Git utiliza esta información para identificar al autor de los cambios
2 realizados en los repositorios. Cuando se realiza un commit en Git,
3 se registra el nombre del autor y la dirección de correo electrónico
4 en los metadatos del commit. Si no se configura esta información correctamente,
5 puede ser difícil rastrear quién hizo qué cambios en un repositorio y dificultar
6 la colaboración con otros usuarios.
7
8
9
10 1. Configurar usuario: git config --global user.name "John Doe"
11 2. Configurar correo electrónico: git config --global user.email johndoe@example.com
12 3. Ver configuraciones: git config --list
13 4. Consultar un parametro especifico: git config user.name
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Instale la versión más reciente de PowerShell para obtener nuevas características y mejoras. https://aka.ms/PSWindows

PS C:\Users\Sysnet> git config --global user.name "Nasc2k12"
PS C:\Users\Sysnet> git config --global user.email nasc2k12@gmail.com
PS C:\Users\Sysnet> git config --list
```

4 Comandos Básicos

Comandos Básicos de Git



Inicializar un repositorio:

```
git init
```

Crea un nuevo repositorio vacío en el directorio actual.

Clonar un repositorio:

```
git clone https://github.com/usuario/repositorio.git
```

Crea una copia local de un repositorio remoto.

Verificar el estado de los archivos:

```
git status
```

Muestra los archivos modificados, agregados, eliminados, etc.

Agregar archivos al área de preparación:

```
git add archivo.txt
```

Agrega un archivo específico al área de preparación.

...Comandos Básicos de Git



Hacer un commit:

`git commit -m "Mensaje del commit"`

Guarda los cambios en el repositorio con un mensaje descriptivo.

Ver el historial de commits:

`git log`

Muestra un historial de los commits hechos en el repositorio.

Subir cambios al repositorio remoto:

`git push origin master`

Envía los cambios locales al repositorio remoto (en la rama principal).

...Comandos Básicos de Git

Obtener cambios del repositorio remoto:

`git pull origin master`

Trae los cambios del repositorio remoto a tu copia local.

Crear una nueva rama:

`git branch nueva-rama`

Crea una nueva rama en el repositorio.

Cambiar de rama:

`git checkout nueva-rama`

Cambia a la rama especificada

```
mkdir my-project
cd my-project
git init
```

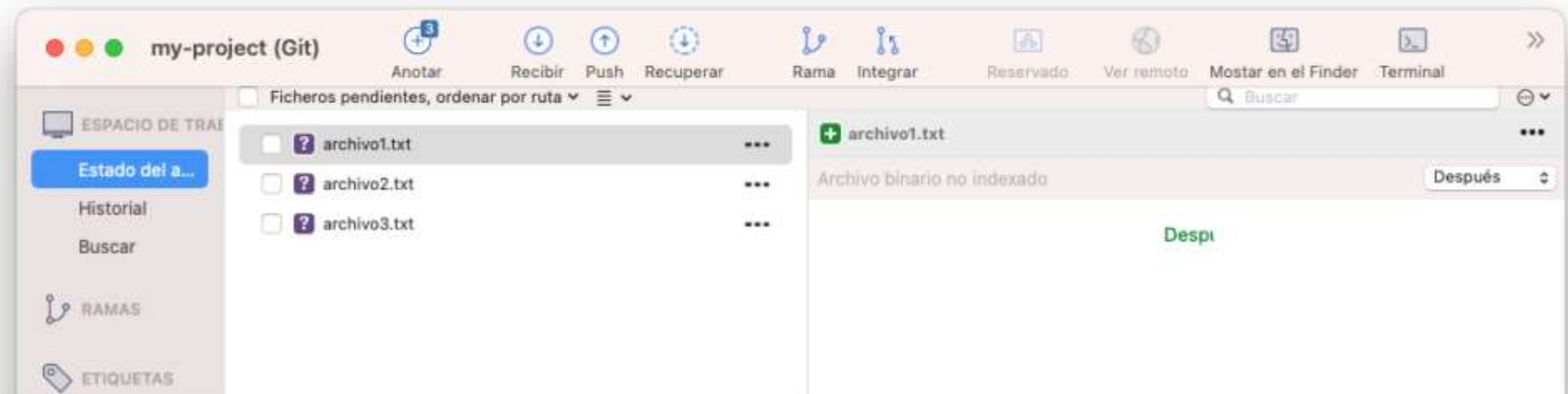
Esto creará el directorio `.git` dentro del directorio `my-project` que contendrá la información sobre el repositorio.



Si queremos crear varios archivos y marcarlos como archivos que queremos incluir al control de cambios, basta con ejecutar estos comandos:

```
touch archivo1.txt
touch archivo2.txt
touch archivo3.txt
git add .
```

Esto creará los archivos vacíos `archivo1.txt`, `archivo2.txt` y `archivo3.txt` y los incluirá a los archivos marcados para incluir al control de cambios.



<https://gandasoftwarefactory.com/git-comandos-basicos-repositorios-ramas-commits/>

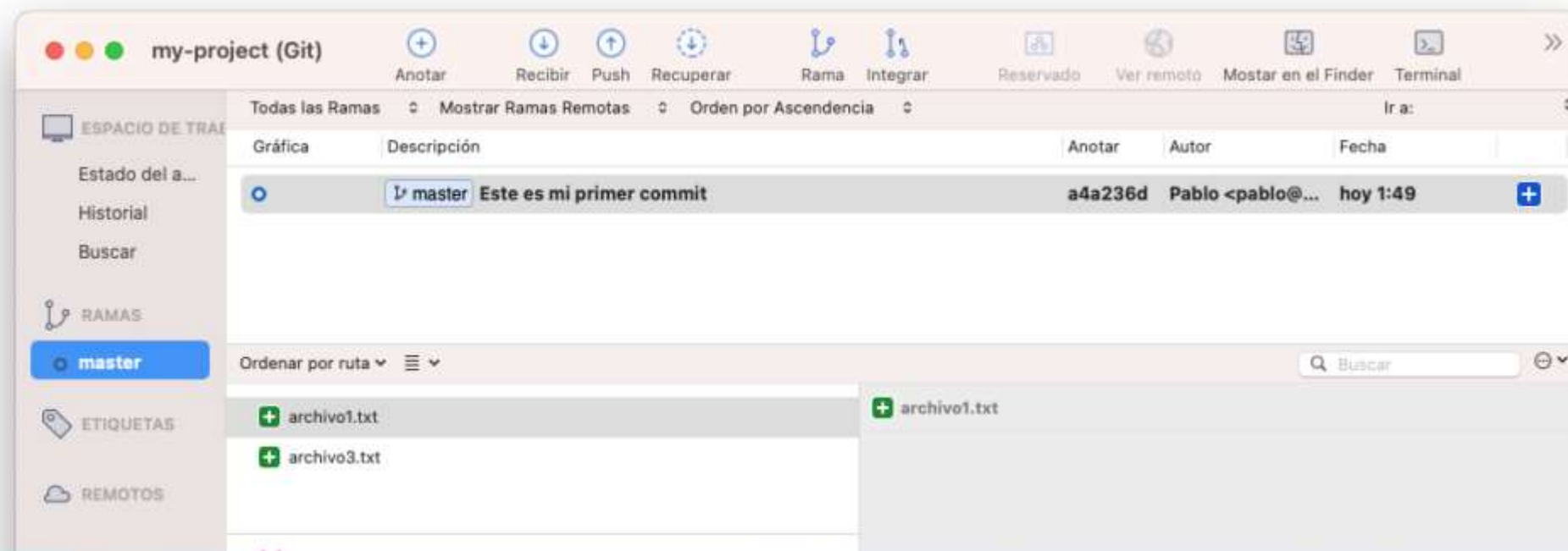
```
git commit -m <descripcion_commit>
```

Ejemplo:

Como queremos incluir los archivos anteriores (archivo1.txt y archivo3.txt) al historial de cambios, ejecutamos este comando:

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

Recuerda que el archivo2.txt lo hemos quitado y por tanto no queda incluido en el historial de cambios.



Práctica

Realiza la configuración de Git con tu nombre y correo electrónico.

Crea un repositorio local en tu máquina.

Clona un repositorio remoto.

Crea una nueva rama llamada adiciones/login.

Cierre

1. ¿Qué es Git y para qué se utiliza?
2. ¿Qué significa "hacer un commit" en Git?
3. ¿Cuál es la función de las ramas en Git?
4. ¿Qué es un repositorio en Git?



Bibliografía

Hernández Bejarno, Miguel. *Ciclo de vida de desarrollo ágil de software seguro*. Fundación Universitaria Los Libertadores. <https://tubiblioteca.utp.edu.pe/cgi-bin/koha/opac-detail.pl?biblionumber=36016>

Guillamón Morales, Alicia. (). *Manual desarrollo de elementos software para gestión de sistemas*.

Editorial CEP, S.L. <https://tubiblioteca.utp.edu.pe/cgi-bin/koha/opac-detail.pl?biblionumber=34982>

Chacon, S., & Straub, B. (2014). *Pro Git* (2nd ed.). Apress.
Enlace: <https://git-scm.com/book/es/v2>

Poulton, N. (2017). *Docker Deep Dive*. Independently published.
Enlace: <https://www.nigelpoulton.com/dvd/>



**Universidad
Tecnológica
del Perú**