

# [L][3] GIT basics

Created by Mike

Last updated about 3 hours ago



## Introducción

Git es un software de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando éstas tienen un gran número de archivos de código fuente. Su propósito es llevar registro de los cambios en archivos de computadora y coordinar el trabajo que varias personas realizan sobre archivos compartidos.

## ¿Cuáles son los principales objetivos en este proyecto?

- Entender que es un control de versiones y para qué se usa.
- Entender que es GIT y cuales son sus fundamentos.
- Familiarizarte con los repositorios

## Descripción del Proyecto

### 1. Instala GIT

El primero paso es instalar git para poder empezar a usarlo. Para ello sigue los pasos que se indican en la guía oficial:

<https://git-scm.com/book/es/v1/Empezando-Instalando-Git>

Es importante que una vez instalado, configures GIT de forma correcta, para ello sigue los pasos indicados en la guía oficial:

<https://git-scm.com/book/es/v1/Empezando-Configurando-Git-por-primera-vez>

## 2. Elabora un listado de operaciones

A continuación, necesitamos que investigues y elabores un listado de las operaciones principales que puedes llevar a cabo usando GIT, tales como:

- añadir un fichero a la zona de stage
- añadir todos los ficheros a la zona de stage con un solo comando
- eliminar un fichero de la zona de stage
- ejecutar un comando para saber como se comprueba la rama en la que te encuentras actualmente
- crear una nueva rama llamada develop
  - aplicar un cambio en esta nueva rama
- hacer un merge de develop a master ( master es la rama principal por que viene creada por defecto )
- comprobar en que estado se encuentra el repositorio

## 3. Crea un repositorio local

Como no podría ser de otro modo, el primer paso consiste en crear un repositorio de Git local ( estará alojado únicamente en tu máquina )

## 4. Crea un repositorio remoto

Si todavía no tienes cuenta en **Bitbucket** y crea un nuevo repositorio online. Es importante que desarrolles las operaciones mencionadas anteriormente y subas tu código al repositorio remoto para que veas como funciona el flujo básico de GIT.

 En este proyecto únicamente realizarás funciones básicas para poder trabajar con

el resto de proyectos y entender los fundamentos de **GIT**.

## 5. Añade cambios desde la interfaz de BitBucket

A continuación editar cualquier fichero de tu repositorio ( que previamente has creado y publicado en **Bitbucket** ) mediante la interfaz web.

Una vez creado el cambio, descárgalo en tu repositorio local para verificar que tu repositorio esta totalmente actualizado.

## Requisitos

- Debes de realizar todos los pasos haciendo uso únicamente de la línea de comandos.
- Debes de configurar tu repositorio para que ignore los siguientes ficheros y directorios
  - Directorios que tengan como nombre dist
  - Todos los ficheros con extensión .txt

## Ejercicio Práctico y presentación

Para finalizar esta píldora, será necesario que realices las siguientes tareas:

1. Crea una presentación explicando todo lo que has aprendido acerca de GIT
2. Explica cual es la diferencia entre Merge y Rebase y haz un ejemplo práctico para explicarlo
3. Explica que es GITFLOW y pon tres ejemplos que sean usados normalmente
4. Explica para que se usan los tags con un ejemplo práctico
5. Explica cual es la diferencia entre GIT fetch y Git pull

## Resources

- GIT Oficial: <https://git-scm.com/>
- Guía: <https://rogerdudler.github.io/git-guide/>
- Guía práctica oficial: <https://guides.github.com/activities/hello-world/>
- Git Cheat sheet: <https://github.github.com/training-kit/downloads/github-git-cheat-sheet.pdf>
- Libro PRO GIT: <https://git-scm.com/book/en/v2>

## ENGLISH

### Introduction

Git is a version control software designed by Linus Torvalds, thinking about the efficiency and reliability of maintaining application versions when they have a large number of source code files. Its purpose is to keep track of changes in computer files and coordinate the work that several people do on shared files.

### What are the main objectives in this project?

- Understand what version control is and what it is used for.
- Understand what GIT is and what its fundamentals are.
- Become familiar with the repositories

### Project description

#### 1. Install GIT

The first step is to install git. To do this, follow the steps indicated in the official guide:

<https://git-scm.com/book/es/v1/Empzando-Instalando-Git>

It is important that once installed, you configure GIT correctly, to do this follow the steps indicated in the official guide:

<https://git-scm.com/book/es/v1/Empzando-Configurando-Git-por-primera-vez>

#### 2. Make a list of operations

Next, we need you to research and prepare a list of the main operations that you can carry out using GIT, such as:

- add a file to the stage area
- add all files to the stage area with a single command
- delete a file from the stage area
- execute a command to know how the branch in which you are currently is checked
- create a new branch called develop or apply a change in this new branch
- make a merge of develop a master (master is the main branch because it is created by default)
- check the state of the repository

### **3. Create a local repository**

As it could not be otherwise, the first step is to create a local Git repository (it will be hosted only on your machine)

### **4. Create a remote repository**

If you do not have a Bitbucket account yet, then create a new online repository. It is important that you develop the operations mentioned above and upload your code to the remote repository, so you can see how the basic GIT flow works.

In this project you will only perform basic functions to be able to work with the rest of the projects and understand the fundamentals of GIT.

### **5. Add changes from the BitBucket interface**

Then edit any file in your repository (which you have previously created and published in Bitbucket) using the web interface.

Once the change is created, download it to your local repository to verify that your repository is fully updated.

## **Requirements**

- You must perform all the steps using only the command line.
- You must configure your repository to ignore the following files and directories:
  - Directories whose name is dist
  - All files with extension .txt

## **Practical exercise and presentation**

To finish this pill, you will need to perform the following tasks:

1. Create a presentation explaining everything you've learned about GIT
2. Explain the difference between Merge and Rebase and make a practical example to explain it.
3. Explain what GITFLOW is and give three examples that are normally used
4. Explain why tags are used with a practical example
5. Explain what is the difference between GIT fetch and Git pull

## Resources

- Official GIT: <https://git-scm.com/>
- Guide: <https://rogerdudler.github.io/git-guide/>
- Official practical guide: <https://guides.github.com/activities/hello-world/>
- Git Cheat sheet: <https://github.github.com/training-kit/downloads/github-git-cheat-sheet.pdf>
- PRO GIT Book: <https://git-scm.com/book/en/v2>

 Me gusta   Sé el primero al que le gusta esto

Sin etiquetas 