

# Sistemas de control de versiones (I)

Ricardo Pérez López

IES Doñana, curso 2020/2021



1. Preparación del entorno de desarrollo
2. Primeros pasos
3. Estado
4. La máquina del tiempo
5. Borrar y mover
6. Git y los directorios
7. Ejercicios
8. Metadatos

# 1. Preparación del entorno de desarrollo

1.1 Instalación automatizada

1.2 Terminal

1.3 Navegador

1.4 Editores de texto

1.5 DokuWiki

1.6 Ejercicios

## 1.1. Instalación automatizada

### 1.1.1 Acciones previas

### 1.1.2 Usar <https://github.com/ricpelo/conf> y seguir las instrucciones del [README.md](#)

## 1.2. Terminal

1.2.1 `zsh`

1.2.2 Oh My Zsh

1.2.3 `less`

## 1.3. Navegador

## 1.4. Editores de texto

1.4.1 Vim y less

1.4.2 Atom

1.4.3 Alternativa: PhpStorm

## 1.5. DokuWiki

1.5.1 Elaboración de documentación

1.5.2 La wiki como sistema de control de versiones

1.5.3 La wiki como herramienta colaborativa



## 1.6. Ejercicios

1.6.1 Indica en la wiki tu nombre completo junto a tu nombre de usuario en GitHub.

1.6.2 Sigue el tutorial de [vimtutor](#) y envía el archivo resultante.

1.6.3 Escoge un paquete del repositorio de paquetes de Atom e indica en la wiki su nombre, su funcionamiento básico, un enlace al paquete dentro del repositorio y por qué te ha resultado interesante.

1.6.4 Escoge una extensión de Google Chrome e indica en la wiki su nombre, su funcionamiento básico, un enlace a la extensión y por qué te ha resultado interesante.

1.6.5 Indica qué editor de textos sería más apropiado usar en cada una de las situaciones siguientes:

Indica qué editor de textos sería más apropiado usar en cada una de las situaciones siguientes:

- ▶ Programar en PHP.
- ▶ Escribir un pequeño script.
- ▶ Cambiar un archivo de configuración del sistema.
- ▶ Editar un archivo situado en otro equipo a través de la red.

## 2. Primeros pasos

2.1 `config`

2.2 `git-config.sh`

2.3 `init`

2.4 `add`

2.5 `commit`

2.6 `checkout` (descartar cambios)

2.7 `reset`

2.8 `.gitignore`

## 2.1. config

## 2.2. git-config.sh

## 2.3. init

## 2.4. add

## 2.5. commit

2.5.1 Con la opción `-m`

2.5.2 Sin la opción `-m`



## 2.6. checkout (descartar cambios)

## 2.7. reset

## 2.8. .gitignore

## 3. Estado

3.1 `status`

3.2 `log`

3.3 Alias `lg`

3.4 `show`

3.5 `diff`

3.6 Referencias

## 3.1. status

## 3.2. log

### 3.3. Alias $l_g$

## 3.4. show



## 3.5. diff

3.5.1 `git diff`

3.5.2 `git diff --staged`

3.5.3 `git diff <commit>`

3.5.4 `git diff inicial..final`

## 3.6. Referencias

3.6.1 HEAD y master

3.6.2 237ab45^

3.6.3 237ab45~1

## 4. La máquina del tiempo

4.1 `checkout` (mover el `HEAD`)

4.2 `revert`

4.3 `reset`

4.4 `tag`

4.5 `--amend`

## 4.1. checkout (mover el HEAD)

## 4.2. revert

## 4.3. reset

## 4.4. tag

## 4.5. --amend



## 5. Borrar y mover

5.1 `rm`

5.2 `mv`

## 5.1. rm

5.2. mv

## 6. Git y los directorios

## 7. Ejercicios

7.1 Sigue el tutorial de Git Immersion desde el LAB3 hasta el LAB20, comprime y envía el directorio resultante.

7.1. Sigue el tutorial de Git Immersion desde el LAB3 hasta el LAB20, comprime y envía el directorio resultante.

## 8. Metadatos

8.1 Objetivos de la unidad

8.2 Resultados de aprendizaje y criterios de evaluación asociados

## 8.1. Objetivos de la unidad

- 8.1.1 Reconocer la importancia y la necesidad de usar un sistema de control de versiones durante el desarrollo de software.
- 8.1.2 Reconocer la utilidad de un sistema de control de versiones en tareas tan diversas como documentación, copias de seguridad, colaboración, despliegue de aplicaciones, etc.
- 8.1.3 Entender la diferencia entre sistemas de control de versiones centralizados y distribuidos, y cómo estos últimos superan abiertamente a los primeros.
- 8.1.4 Reconocer a Git como un sistema de control de versiones distribuido.
- 8.1.5 Reconocer la importancia que tiene Git en el panorama actual de desarrollo de software.
- 8.1.6 Entender los conceptos de repositorio, directorio de trabajo, stage, commit, log.
- 8.1.7 Aprender el funcionamiento básico de Git en un repositorio local.
- 8.1.8 Aprender a moverse a través del tiempo por los commits de un repositorio Git.
- 8.1.9 Aprender a corregir commits creando nuevos commits.



## 8.2. Resultados de aprendizaje y criterios de evaluación asociados