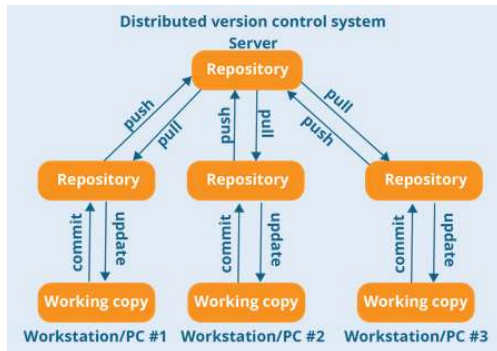


# GIT

Software de control de versiones.

Primero hacemos un clon del repositorio general para tener un repositorio propio y una copia local de esta. Si se rompe el ordenador la copia se pierde pero nuestro repositorio propio no.



Una vez hemos hecho cambios hacemos un commit para actualizar nuestro repositorio.

Ahora hacemos un push para actualizar del repositorio central los ficheros modificados en nuestro repositorio.

Si ahora otro intenta subir el mismo archivo modificado le daría error, ya que debería hacer un push para descargarse la última versión del archivo y trabajar con ella.

Si se han modificado varios archivos es preferible hacer commit por cada archivo cuando se ha acabado cada uno pero un único push.

Si trabajando con un archivo local nos equivocamos, hacemos un **revert** que nos trae la copia de ese archivo de nuestro repositorio.

Se debe hacer un **update** para comprobar si alguien ha actualizado el repositorio central.

Se trabaja por ramas, donde la principal se llama master y no se le hace commit. Para cada tarea se hace una rama,

Vamos a crear una página en GitHub.

En la página de comandos ponemos `git --version` y nos sale la versión.

Todo lo que hagamos y entregemos lo subimos a git.

En ajustes le damos a new SSH key y ponemos la public key que creamos.

The screenshot shows the GitHub account settings page for user 'sergijoan22'. On the left is a sidebar menu with options: Account settings, Profile, Account, Appearance, Account security, Billing & plans, Security log, Security & analysis, Sponsorship log, Emails, and Notifications. The main content area is titled 'SSH keys' and includes a 'New SSH key' button. Below this, it states: 'This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.' A table lists one 'Personal key' with a SHA256 fingerprint, added on 18 Sep 2021, and marked as 'Never used'. A 'Delete' button is next to it. Below the table is a link to a guide on generating SSH keys. The 'GPG keys' section below it states 'There are no GPG keys associated with your account' and includes a 'New GPG key' button and a link to learn how to generate a GPG key.

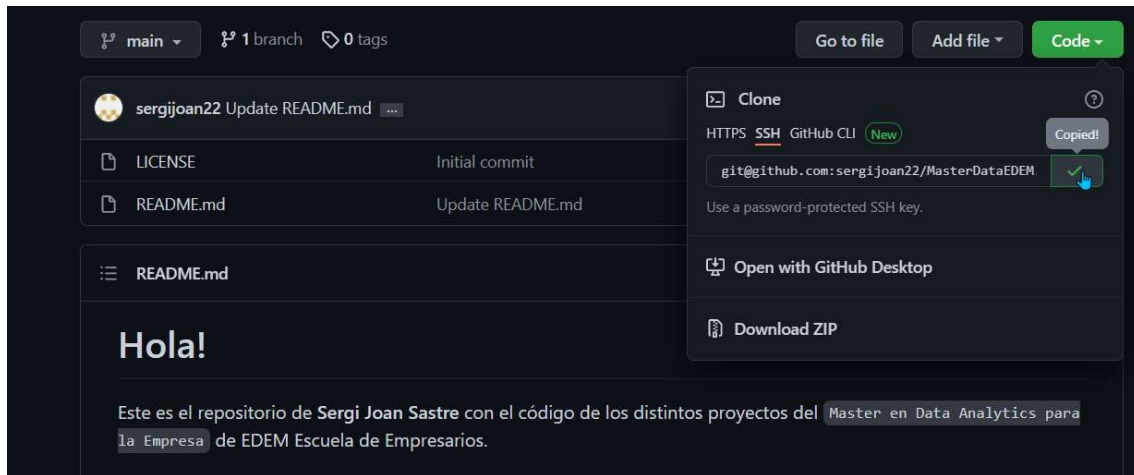
Ahora creamos un repositorio nuevo

The screenshot shows the 'Create new repository' form in GitHub. It includes fields for 'Owner' (sergijoan22) and 'Repository name' (MasterDataEDEM), which is marked with a green checkmark. A tip suggests repository names should be short and memorable. There is an optional 'Description' field. Under 'Visibility', 'Public' is selected. The 'Initialize this repository with' section has three checked options: 'Add a README file', 'Add .gitignore', and 'Choose a license' (set to 'Apache License 2.0'). A note at the bottom states the default branch will be 'main'. A green 'Create repository' button is at the bottom.

En <https://stackedit.io/> podemos crear la página principal para GitHub.

En el icono de lápiz de la página principal permite cambiar el README, enviando un commit para ello.

En insights se pueden ver los commits de todos.



En comandos nos vamos a la carpeta que queremos

```
C:\Users\sergi\Archivos\Master\Git>cd C:\Users\sergi\OneDrive\Máster\Fundamentos intro\Git
```

Y ponemos

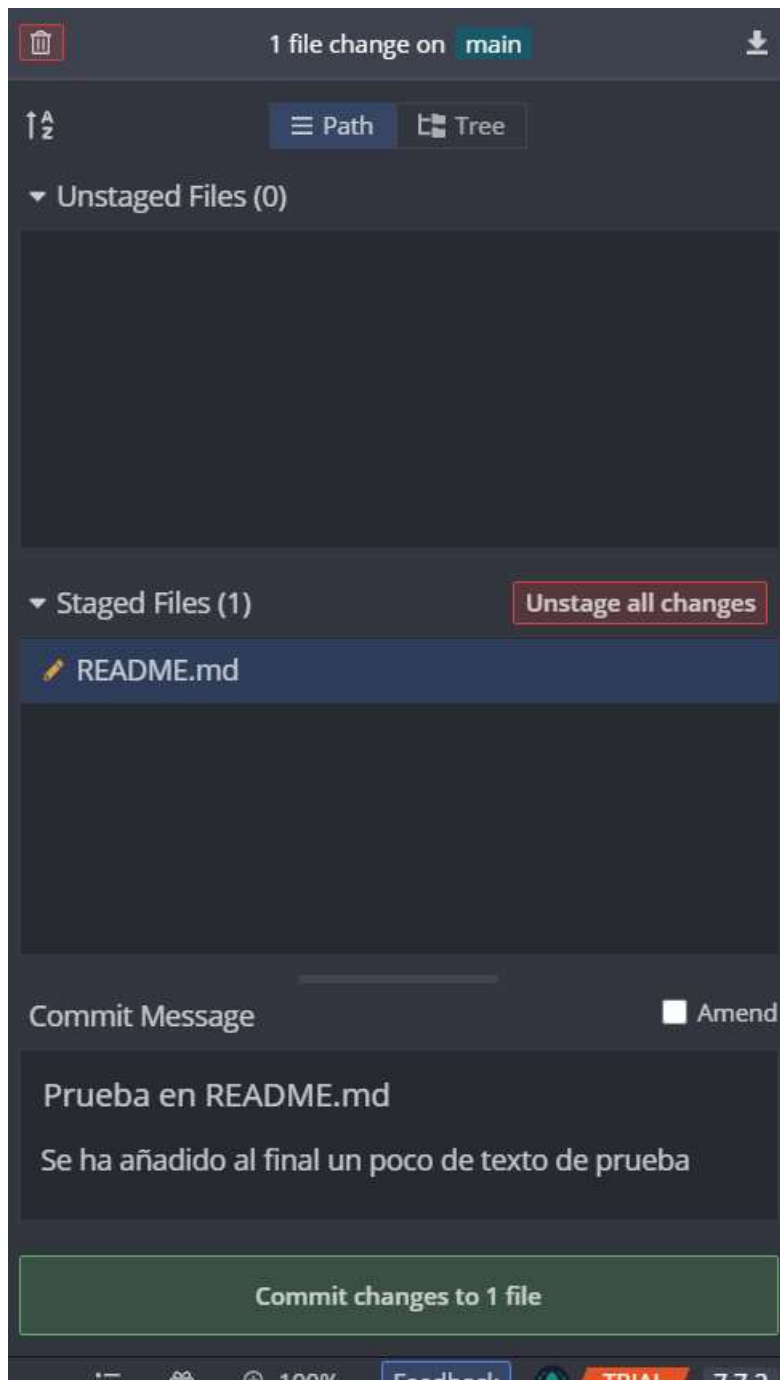
```
C:\Users\sergi\OneDrive\Máster\Fundamentos intro\Git>git clone git@github.com:sergijoan22/MasterDataEDEM.git
```

No nos dejará así que vamos a GitKraken y en Clone a Repo creamos una copia local del repositorio de GitHub. En file – preferencias – ssh hay que poner la dirección a las claves público y privada.

Abrimos la copia y se muestra la línea temporal en vertical. En file – open in file manager nos lleva al explorador local donde esta la copia. Si editamos en la línea sale una línea que pone WIP que muestra que hay cambios. Al darle en la derecha pone los archivos cambiados. Al archivo que queramos le damos a stage file. Lo que está en stage es lo que se commitea.

## EJERCICIO

Volvemos a hacer una copia



Le damos a commit y luego arriba a push



En branch creamos una rama que le llamamos rama1. Este se crea en el repositorio intermedio pero no en local.

Hacemos una nueva modificación y un nuevo commit igual. A la hora de hacer push, lo ponemos en rama1.

En GitHub saldrá un Pull request y la aceptamos con los botones verdes.

En GitKraken ahora se muestran las dos líneas temporales.



Ahora vamos a crear un nuevo fichero en la carpeta local, y saldrá un WIP en GitKraken. Hacemos stage al archivo y hacemos el commit.

## EJERCICIO

Se nos añade a un repositorio de clase, se hace una copia

Cuando creamos una rama se una línea temporal a la derecha y al hacer un pull request al main se vuelve a la línea principal