

Git y GitHub

Práctica 2

- 1.- Explica brevemente que es un sistema de control de versiones distribuido y sus diferencias con respecto a uno centralizado.
- 2.- Explica los fundamentos de git: instantáneas frente a modificaciones.
- 3.- Establecer tu nombre de usuario y dirección de correo electrónico. Esto es importante porque las confirmaciones de cambios (commits) en Git usan esta información, y es introducida de manera inmutable en los commits que envías. Utiliza tu usuario y correo de la uco, por ejemplo:

```
$ git config --global user.name "i22lojal"  
$ git config --global user.email "i22lojal@uco.es"
```
- 4.- Crea un directorio compuesto por tu nombre de usuario más “Prueba1” (ej. i22lojalPrueba1), trasládase a dicho directorio y inicializa un nuevo repositorio en él. Comprueba que de él cuelga un fichero .git. ¿Qué función tiene este fichero?
- 6.- Crea un directorio que cuelgue del anterior llamado “/home/i22lojal/i22lojalPrueba1/includes”
- 7.- Escribe un programa llamado “/home/i22lojal/i22lojalPrueba1/holamundo.c” que contenga:

```
#include "../includes/miinclude.h"  
  
int main(){  
    printf("Hola mundo\n");  
    return 0;  
}
```

y un fichero “/home/i22lojal/i22lojalPrueba1/includes/miinclude.h” que contenga:

```
#include <stdio.h>
```

- 8.- Añade estos ficheros a tu repositorio y comprueba el estado del repositorio con “git status”. Haz el primer commit con la etiqueta “Init commit” y vuelve a comprobar el estado de tu repositorio.

- 9.- Compila los el programa anterior y añade el ejecutable al repositorio.

- 10.- Abre una rama que se llama “ramaA” y modifica el fichero .c

```
#include "../includes/miinclude.h"  
  
int main(){  
    char c[100]= "Hola mundo";  
  
    printf("%s\n", strcat(c, " soy i22lojal"));  
    return 0;  
}
```

11.- En la rama master añade “#include <string.h>”

12.- Fusiona ambas ramas.

13.- Sube este proyecto a GitHub

14.- Bájate un proyecto de uno de tus compañeros, modifícalo y sube las modificaciones.