

TALLER DE SUPERVIVENCIA EN SHELL + GIT

créditos RAC



@robotech_urjc

asociacion.robotech@urjc.es



(cc) 2018 Grupo de Sistemas y Comunicaciones.

(cc) 2020 Robotech.

Algunos derechos reservados. Este trabajo se entrega bajo la licencia Creative Commons Reconocimiento –NoComercial - SinObraDerivada (by-nc-nd). Para obtener la licencia completa, véase

<http://creativecommons.org/licenses/by-sa/2.1/es>. También puede solicitarse a Creative Commons, 559 Nathan

Abbott Way, Stanford, California 94305, USA.

¿Quiénes somos?

Claudia Élez Mencía

- Estudiante de Ingeniería en Robótica Software (2º curso).

Adrián Manzanares Romera

- Estudiante de Ingeniería en Robótica Software (2º curso).

¿Qué debemos saber?



Uso básico de un comando.

```

manza@apel: ~/Desktop/taller_shell_24_25
manza@apel:~/Desktop/taller_shell_24_25$ ls
prueba1.txt
manza@apel:~/Desktop/taller_shell_24_25$ ls -last
total 8
4 drwxrwxr-x  2 manza manza 4096 Sep 20 18:41 .
0 -rw-rw-r--  1 manza manza   0 Sep 20 17:45 prueba1.txt
4 drwxr-xr-x 19 manza manza 4096 Sep 20 17:45 ..
manza@apel:~/Desktop/taller_shell_24_25$ touch prueba 2.txt
manza@apel:~/Desktop/taller_shell_24_25$ ls prueba 2.txt
2.txt prueba
manza@apel:~/Desktop/taller_shell_24_25$ ls -last
total 8
4 drwxrwxr-x  2 manza manza 4096 Sep 20 18:42 .
0 -rw-rw-r--  1 manza manza   0 Sep 20 18:42 2.txt
0 -rw-rw-r--  1 manza manza   0 Sep 20 18:42 prueba
0 -rw-rw-r--  1 manza manza   0 Sep 20 17:45 prueba1.txt
4 drwxr-xr-x 19 manza manza 4096 Sep 20 17:45 ..
manza@apel:~/Desktop/taller_shell_24_25$

```

Comando

Flag(s)

Argumento(s)

Salida

username@pcname

Ruta de trabajo

Comando: man

- Las páginas del manual se pueden consultar con el comando `man sección asunto`. Ejemplo: `man 1 ls`.
- Secciones de interés: comandos (1), llamadas al sistema(2), llamadas a biblioteca(3).
- Para buscar sobre una palabra: `apropos`. Ejemplo: `apropos gcc`.

```

manza@apel: ~/Desktop/taller_shell_24_25
LS(1)                                User Commands                                LS(1)

NAME
    ls - list directory contents

SYNOPSIS
    ls [OPTION]... [FILE]...

DESCRIPTION
    List information about the FILES (the current directory by default). Sort entries
    alphabetically if none of -cftuvSUX nor --sort is specified.

    Mandatory arguments to long options are mandatory for short options too.

    -a, --all
        do not ignore entries starting with .

    -A, --almost-all
        do not list implied . and ..

    --author
        with -l, print the author of each file

Manual page ls(1) line 1 (press h for help or q to quit)

```

Comandos básicos

- `cd`: cambia de directorio actual.
- `echo`: escribe sus argumentos por su salida.
- `touch`: cambia la fecha de modificación de un fichero. Si no existe el fichero, se crea.
- `ls`: lista el contenido de un directorio.
- `cp`: copia ficheros.
- `mv`: mueve ficheros.
- `rm`: borra ficheros.
- `mkdir`: crea directorios.
- `rmdir`: borra directorios vacíos.
- `date`: muestra la fecha.

Comandos básicos

- `who`: muestra los usuarios que están en el sistema.
- `whoami`: muestra tu nombre de usuario.
- `sort`: ordena las líneas de un fichero.
- `wc`: cuenta caracteres, palabras y líneas de ficheros.
- `grep`: buscan cadenas dentro de ficheros.
- `diff`: compara ficheros.
- `cat`: escribe en su salida el contenido de uno o varios ficheros.
- `less`: permite leer un fichero de texto en el terminal usando scroll.

Comandos básicos

- `file`: da pistas sobre el contenido de un fichero.
- `head`, `tail`: escriben el las primeras / últimas líneas del fichero en su salida.
- `tar`: crea un fichero con múltiples ficheros dentro(comprimidos o no).
- `gzip/gunzip`: comprime/descomprime un fichero.
- `top`: muestra los procesos y el estado de sistema.
- `reset`: restablece el estado del terminal.
- `exit`: el shell termina su ejecución.
- `sudo`: otorga permisos de administrador ¡PELIGROSO!

Editor en modo texto: nano

- Uso: nano [nombre del fichero]

```

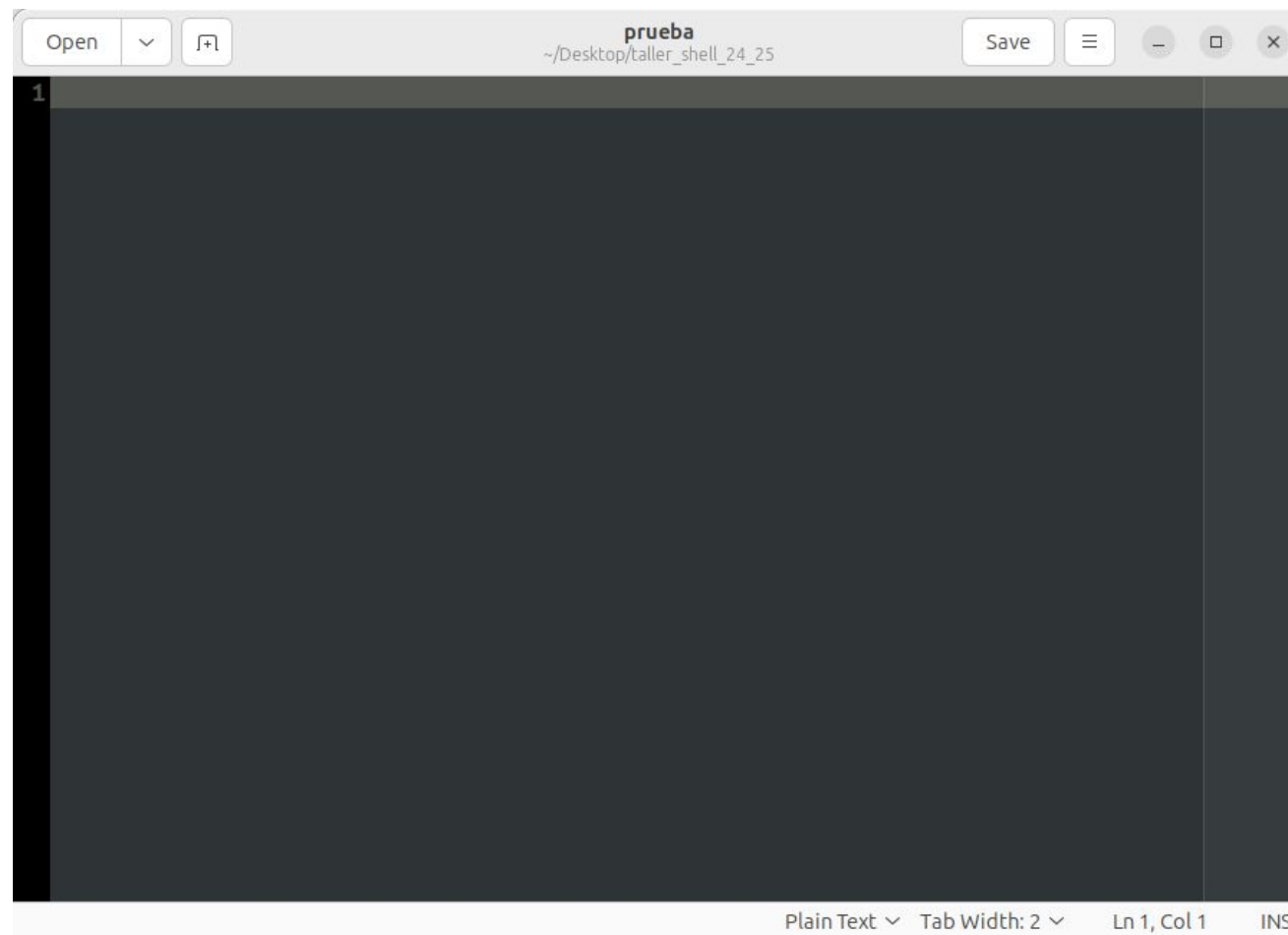
manza@apel: ~/Desktop/taller_shell_24_25
GNU nano 7.2 prueba33.txt

[ New File ]

^G Help    ^O Write Out  ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit    ^R Read File  ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line
  
```

Editor en modo texto: gedit

- **Uso:** `gedit [nombre del fichero]`



Git

- Git es un sistema de control de versiones especializado en código.



Git: Plataformas

- Git es un protocolo que necesita comunicarse con una plataforma, las principales son:



CREACIÓN CUENTA EN GITHUB

Github.com —> Sign up

Welcome to GitHub!
Let's begin the adventure

Enter your email*

✗ EMAIL@EMAIL.COM

Create a password*

✓ CONTRASEÑA1234++++

Enter a username*

✓ USERNAME12345678-DOTCO

Email preferences

☐ Receive occasional product updates and announcements.

Continue

VINCULAR - TERMINAL

Github.com → Sign up → cat .ssh/id_rsa.pub



```
manza@apel: ~  
manza@apel:~$ cat .ssh/id_rsa.pub  
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQgQDhC4oR3XX1RcnRloSREGa9X25cIXYW5A9jh7HXLnSr  
manza@apel:~$
```

INFORMACIÓN PRIVADA - NO COMPARTIR

VINCULAR - GITHUB

Click perfil → Settings → ssh and gpg keys → new ssh key

Add new SSH Key

Title

PORTATIL_PERSONAL

Key type

Authentication Key

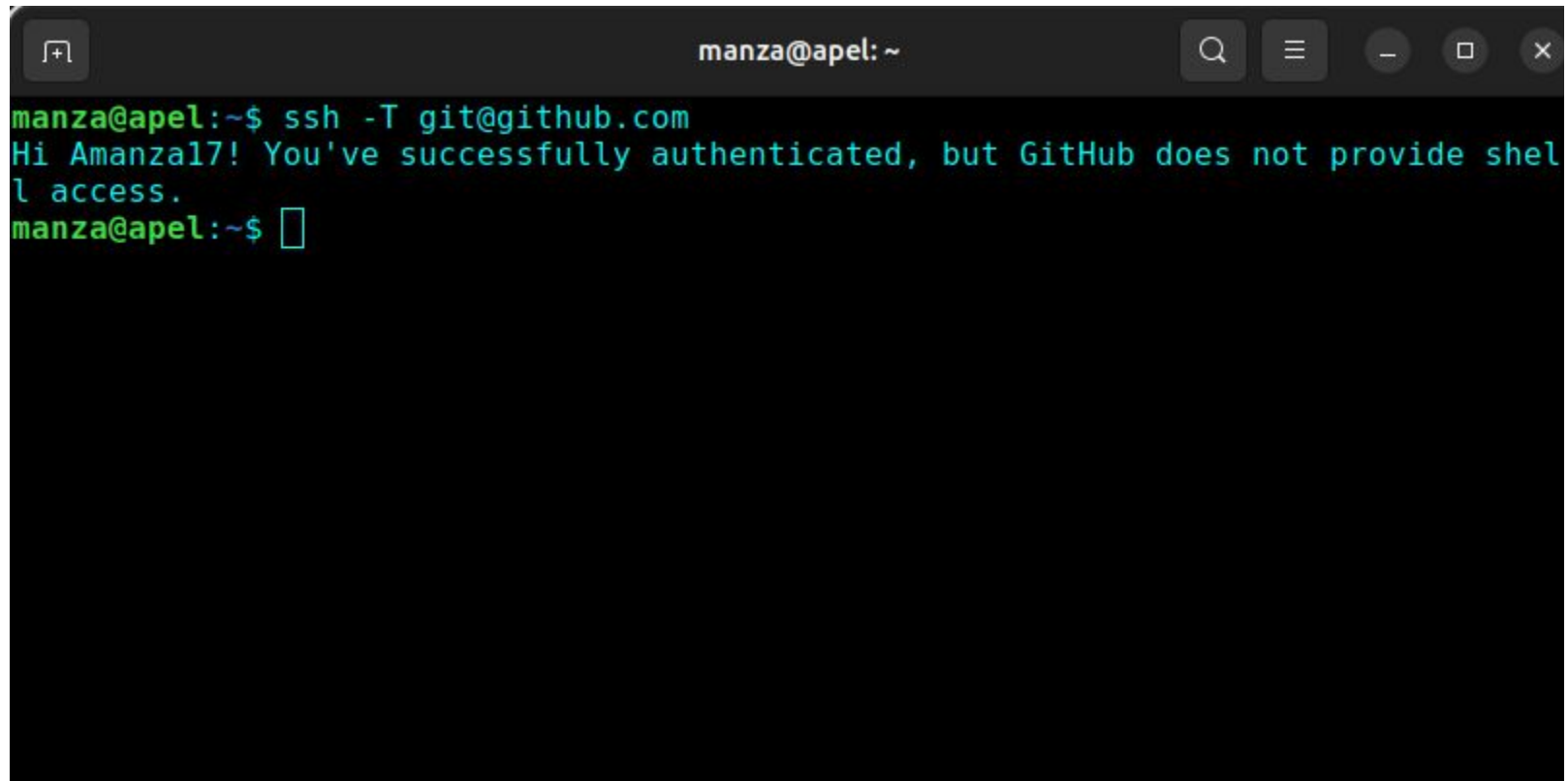
Key

ssh-rsa

Add SSH key

VINCULAR - TERMINAL

ssh -T git@github.com



```
manza@apel: ~  
manza@apel:~$ ssh -T git@github.com  
Hi Amanza17! You've successfully authenticated, but GitHub does not provide shell  
access.  
manza@apel:~$
```

Git: configuración inicial

- Es necesario realizar una configuración inicial antes de empezar a utilizar git.
- `git config --global user.name "[Nombre de usuario de git]"`
- `git config --global user.email "[Correo de Git]"`

sudo apt install git

```
manza@apel: ~
manza@apel:~$ git config --global user.name "Amanza17"
manza@apel:~$ git config --global user.email "a.manzanares.2023@alumnos.urjc.es"
manza@apel:~$
```

Git: Comandos básicos

- `git add [fichero o directorio]`: Agrega los cambios.
- `git commit -m "Version 1.0"`: Registra esos cambios como una versión.
- `git push`: Sube los commits a la plataforma elegida.
- `git pull`: Baja los commits más actualizados.
- `git clone`: Clona un repositorio remoto a un directorio local.
- `git branch [nombre de la rama]`: Crea una rama.
- `git checkout [nombre de la rama]`: Cambia la rama del repositorio local.
- `git checkout [commit hash]`: Cambia al estado de ese commit.
- `git merge [nombre de la rama]`: Fusiona la rama actual con la rama introducida.

¡Al turrón!

