

Act 1 Despliegue Juan Carlos Saldaña Herrero

En este documento para la asignatura de **despliegue** trato de proporcionar una **guía básica de comandos** de consola para la **administración de un servidor web** o servidor de aplicaciones.

Revisar conexión a Internet: Ping y Tracert, Ifconfig

Podemos usar simplemente el comando **ping** a una **dirección** como la del instituto o **tracert** (tracert para Linux/Mac), en cuyo caso trazamos la ruta que hace un paquete entrante desde un host (la web del instituto, por ejemplo), y nuestro PC. De esta forma confirmamos que la **conexión al exterior funciona correctamente**.

```
PS C:\Windows\system32> ping www.telefonicaeducaciondigital.com

Haciendo ping a www.telefonicaeducaciondigital.com [194.224.110.38] con 32 bytes de datos:
Respuesta desde 194.224.110.38: bytes=32 tiempo=17ms TTL=246
Tiempo de espera agotado para esta solicitud.
Respuesta desde 194.224.110.38: bytes=32 tiempo=14ms TTL=246
Respuesta desde 194.224.110.38: bytes=32 tiempo=130ms TTL=246

Estadísticas de ping para 194.224.110.38:
    Paquetes: enviados = 4, recibidos = 3, perdidos = 1
      (25% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
      Mínimo = 14ms, Máximo = 130ms, Media = 53ms
PS C:\Windows\system32> tracert www.telefonicaeducaciondigital.com

Traza a la dirección www.telefonicaeducaciondigital.com [194.224.110.38]
sobre un máximo de 30 saltos:

 1    33 ms    67 ms    15 ms    192.168.68.1
 2    10 ms     5 ms     5 ms    192.168.1.1
 3     6 ms    11 ms     6 ms    192.168.144.1
 4    17 ms    11 ms    14 ms    185.red-81-46-66.customer.static.ccgg.telefonica.net [81.46.66.185]
 5      *      *      *      Tiempo de espera agotado para esta solicitud.
 6      *      *      *      Tiempo de espera agotado para esta solicitud.
 7      *      *      *      Tiempo de espera agotado para esta solicitud.
 8      *      *      *      Tiempo de espera agotado para esta solicitud.
 9      *      *      *      Tiempo de espera agotado para esta solicitud.
10    14 ms   107 ms   109 ms   38.red-194-224-110.customer.static.ccgg.telefonica.net [194.224.110.38]

Traza completa.
```

Cuento con una máquina virtual **Linux** en la que podemos ejecutar comandos similares para ver la diferencia, es este caso contamos con el comando **ifconfig**, que al ser usado sin argumentos nos muestra detalles de las interfaces activas:

```
carlos@carlos-VirtualBox:~/Escritorio$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
    inet6 fe80::3d7a:4e2f:8ece:8206 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:33:11:2d txqueuelen 1000 (Ethernet)
    RX packets 78329 bytes 117392499 (117.3 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 19424 bytes 1219054 (1.2 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Además, el comando **ping** también funciona en esta consola:

```
carlos@carlos-VirtualBox:~/Escritorio$ ping www.telefonicaeducaciondigital.com
PING www.telefonicaeducaciondigital.com (194.224.110.38) 56(84) bytes of data.
64 bytes from 38.red-194-224-110.customer.static.ccgg.telefonica.net (194.224.110.38): icmp_seq=1 ttl=245 time=13.1 ms
64 bytes from 38.red-194-224-110.customer.static.ccgg.telefonica.net (194.224.110.38): icmp_seq=2 ttl=245 time=5.25 ms
64 bytes from 38.red-194-224-110.customer.static.ccgg.telefonica.net (194.224.110.38): icmp_seq=3 ttl=245 time=6.23 ms
```

Como hemos dicho, el **Linux** podemos usar **traceroute** en vez de tracert, pero necesita del siguiente comando para su previa **instalación**:

```
carlos@carlos-VirtualBox:~/Escritorio$ sudo apt install traceroute
[sudo] contraseña para carlos:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes NUEVOS:
  traceroute
```

Vemos que ahora se ejecuta correctamente:

```
carlos@carlos-VirtualBox:~/Escritorio$ traceroute www.telefonicaeducaciondigital.com
traceroute to www.telefonicaeducaciondigital.com (194.224.110.38), 30 hops max, 60 byte packets
 1 _gateway (10.0.2.2)  0.247 ms  0.246 ms  0.239 ms
```

De igual forma podemos añadir muchas terminaciones a este comando ifconfig que nos permiten establecer la máscara de red IP para la interfaz, la métrica, abrir o cerrar la interfaz con up/down. Volveré a esto en el siguiente punto

Servidor accesible: Netstat, Ufw

Tenemos muchas formas de ejecutar este comando, por poner un ejemplo, en el caso de **netstat -a**, podemos ver que tenemos el **puerto 80** abierto y escuchando, el que usaremos para html en la siguiente actividad.

Nota: por defecto se usaba el puerto 8080 que ya estaba en uso por la aplicación adobe connect, hubo que cerrarla para proceder con la instalación. Sabemos de su correcto funcionamiento gracias a este comando netstat.

```
PS C:\Windows\system32> netstat -a

Conexiones activas

Proto  Dirección local      Dirección remota      Estado
TCP    0.0.0.0:80            DESKTOP-UURFOJ9:0     LISTENING
TCP    0.0.0.0:135           DESKTOP-UURFOJ9:0     LISTENING
TCP    0.0.0.0:443           DESKTOP-UURFOJ9:0     LISTENING
TCP    0.0.0.0:445           DESKTOP-UURFOJ9:0     LISTENING
TCP    0.0.0.0:5040          DESKTOP-UURFOJ9:0     LISTENING
TCP    0.0.0.0:7680          DESKTOP-UURFOJ9:0     LISTENING
TCP    0.0.0.0:22333         DESKTOP-UURFOJ9:0     LISTENING
```

En **Linux** podemos ejecutar el **mismo comando**:

```
carlos@carlos-VirtualBox:~/Escritorio$ netstat -a
Conexiones activas de Internet (servidores y establecidos)
Proto Recib Envíad Dirección local      Dirección remota      Estado
tcp    0      0 localhost:domain      0.0.0.0:*              ESCUCHAR
tcp    0      0 localhost:ipp          0.0.0.0:*              ESCUCHAR
tcp6   0      0 ip6-localhost:ipp     [::]:*                 ESCUCHAR
```

Sin embargo, nos da un resultado muy largo, con **netstat -i** podemos ver la lista de interfaces:

```
carlos@carlos-VirtualBox:~/Escritorio$ netstat -i
Tabla de la interfaz del núcleo
Iface    MTU     RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR Flg
enp0s3   1500    78566 0      0 0      19804 0      0 0  BMRU
lo       65536   541    0      0 0      541    0      0 0  LRU
```

Que, como dijimos antes, podemos abrir y cerrar. En la secuencia de comando siguientes mostramos las **interfaces activas**, **cerramos una**, **las mostramos** y vemos que ya no está la que hemos cerrado, y la **volvemos a abrir**:

```
carlos@carlos-VirtualBox:~/Escritorio$ netstat -i
Tabla de la interfaz del núcleo
Iface      MTU      RX-OK RX-ERR RX-DRP RX-OVR      TX-OK TX-ERR TX-DRP TX-OVR Flg
enp0s3     1500     78588 0      0 0      19868 0      0 0 BMRU
lo         65536     613 0      0 0      613 0      0 0 LRU
carlos@carlos-VirtualBox:~/Escritorio$ sudo ifconfig enp0s3 down
carlos@carlos-VirtualBox:~/Escritorio$ netstat -i
Tabla de la interfaz del núcleo
Iface      MTU      RX-OK RX-ERR RX-DRP RX-OVR      TX-OK TX-ERR TX-DRP TX-OVR Flg
lo         65536     619 0      0 0      619 0      0 0 LRU
carlos@carlos-VirtualBox:~/Escritorio$ sudo ifconfig enp0s3 up
carlos@carlos-VirtualBox:~/Escritorio$ netstat -i
Tabla de la interfaz del núcleo
Iface      MTU      RX-OK RX-ERR RX-DRP RX-OVR      TX-OK TX-ERR TX-DRP TX-OVR Flg
enp0s3     1500     78598 0      0 0      19898 0      0 0 BMRU
lo         65536     635 0      0 0      635 0      0 0 LRU
carlos@carlos-VirtualBox:~/Escritorio$
```

También disponemos de una serie de comandos con **ufw** para modificar de forma sencilla el **firewall**. Con el prefijo **sudo** para entrar como administrador, estos son algunos ejemplos:

```
carlos@carlos-VirtualBox:~/Escritorio$ sudo ufw disable
El cortafuegos está detenido y deshabilitado en el arranque del sistema
carlos@carlos-VirtualBox:~/Escritorio$ sudo ufw enable
El cortafuegos está activo y habilitado en el arranque del sistema
carlos@carlos-VirtualBox:~/Escritorio$ sudo ufw status
Estado: activo
carlos@carlos-VirtualBox:~/Escritorio$ sudo ufw status verbose
Estado: activo
Acceso: on (low)
Predeterminado: deny (entrantes), allow (salientes), disabled (enrutados)
Perfiles nuevos: skip
carlos@carlos-VirtualBox:~/Escritorio$
```

¿A quién pertenece una URL? Nslookup, Dig

Gracias a **nslookup**, es posible la **búsqueda inversa**, es decir, consultar el propietario de la ip.

También podemos introducir directamente la ip:

```
PS C:\Windows\system32> nslookup 194.224.110.38
Servidor: 250.red-80-58-61.staticip.rima-tde.net
Address: 80.58.61.250

Nombre: 38.red-194-224-110.customer.static.ccgg.telefonica.net
Address: 194.224.110.38
Aliases: 38.110.224.194.in-addr.arpa

PS C:\Windows\system32> nslookup www.telefonicaeducaciondigital.com
Servidor: 250.red-80-58-61.staticip.rima-tde.net
Address: 80.58.61.250

Respuesta no autoritativa:
Nombre: www.telefonicaeducaciondigital.com
Address: 194.224.110.38
```

En Linux vamos a instalar **Dig** con el mismo propósito. Vemos que nos solicita super usuario:

```
carlos@carlos-VirtualBox:~/Escritorio$ apt-get install dnsutils
E: No se pudo abrir el fichero de bloqueo «/var/lib/dpkg/lock-frontent» - open (
13: Permiso denegado)
E: No se pudo obtener el bloqueo de la interfaz dpkg (/var/lib/dpkg/lock-fronten
d). ¿Es usted superusuario?
carlos@carlos-VirtualBox:~/Escritorio$ sudo apt-get install dnsutils
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes NUEVOS:
dnsutils
```

Comprobamos su **correcta instalación**:

```
carlos@carlos-VirtualBox:~/Escritorio$ dig -v
DiG 9.16.6-Ubuntu
carlos@carlos-VirtualBox:~/Escritorio$
```

Aplicamos de forma **similar a nslookup**:

```
carlos@carlos-VirtualBox:~/Escritorio$ dig 194.224.110.38

; <<>> DiG 9.16.6-Ubuntu <<>> 194.224.110.38
;; global options: +cmd
;; Got answer:
;; ->HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 11174
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1
```

Hacemos un **trazado similar a tracert**:

```
carlos@carlos-VirtualBox:~/Escritorio$ dig www.telefonicaeducaciondigital.com +trace

; <<>> DiG 9.16.6-Ubuntu <<>> www.telefonicaeducaciondigital.com +trace
;; global options: +cmd
.          7184      IN      NS      b.root-servers.net.
.          7184      IN      NS      a.root-servers.net.
.          7184      IN      NS      m.root-servers.net.
.          7184      IN      NS      l.root-servers.net.
.          7184      IN      NS      k.root-servers.net.
.          7184      IN      NS      j.root-servers.net.
.          7184      IN      NS      i.root-servers.net.
.          7184      IN      NS      h.root-servers.net.
.          7184      IN      NS      g.root-servers.net.
.          7184      IN      NS      f.root-servers.net.
.          7184      IN      NS      e.root-servers.net.
.          7184      IN      NS      d.root-servers.net.
.          7184      IN      NS      c.root-servers.net.
;; Received 262 bytes from 127.0.0.53#53(127.0.0.53) in 0 ms
```

Probamos acceso a servidor: Curl, wget

Curl es la abreviación de “**Client URL**”. Instalamos en Linux como de costumbre:

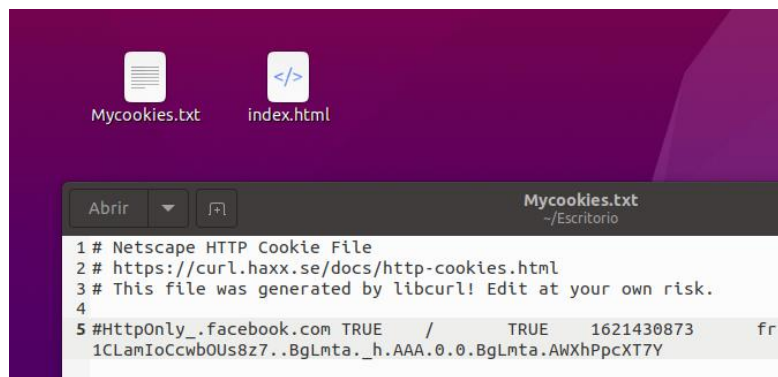
```
carlos@carlos-VirtualBox:~/Escritorio$ sudo apt install curl
[sudo] contraseña para carlos:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes NUEVOS:
curl
```

Este comando es útil para **transferir datos**, en este caso, por ejemplo, lo usamos para ver el encabezado de la página de Facebook:

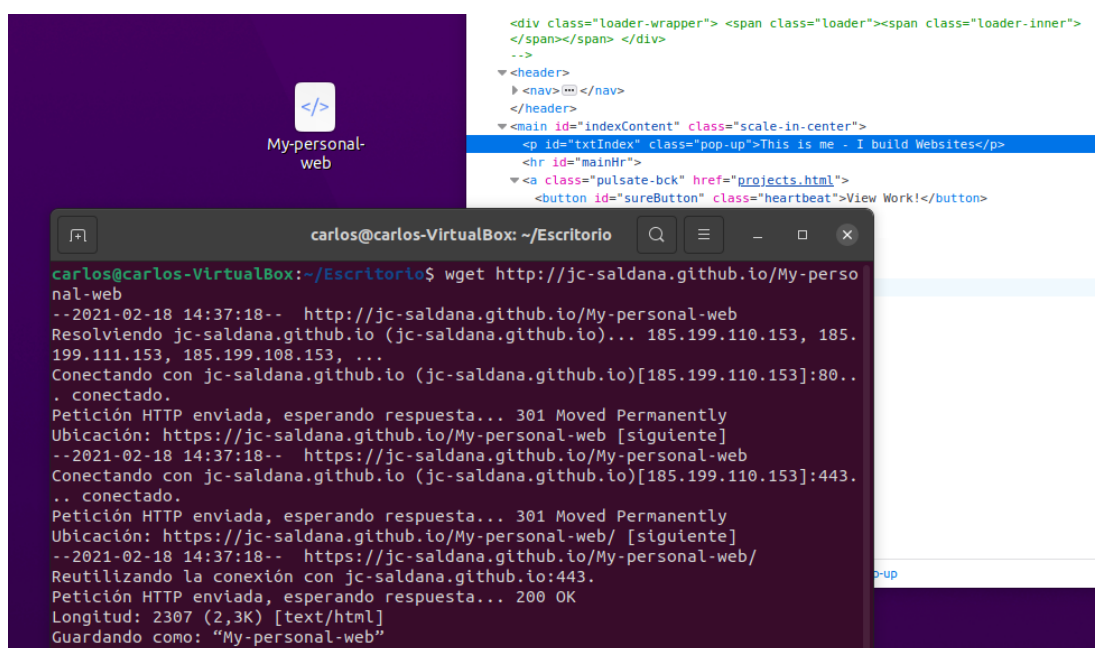
```
carlos@carlos-VirtualBox:~/Escritorio$ curl -I www.facebook.com
HTTP/1.1 301 Moved Permanently
Location: https://www.facebook.com/
Content-Type: text/html; charset="utf-8"
X-FB-Debug: 5Tno+bf03ZyLXEhpgGLYnwkqI7sdt3BfAQ6TaCCvCFhsFuf8uVz5eBRdMiJWdZq3ALLcxxW0iHEq4VK+jakgKQ==
Date: Thu, 18 Feb 2021 13:23:53 GMT
Alt-Svc: h3-29=":443"; ma=3600,h3-27=":443"; ma=3600
Connection: keep-alive
Content-Length: 0
```

También podemos usarlo para **descargar archivos desde un servidor remoto con FTP**, o incluso **ver las cookies que se descargan**. Para este último caso, vemos que nos genera un txt y un html con el contenido de la página que especifiquemos, en este caso Facebook de nuevo:

```
carlos@carlos-VirtualBox:~/Escritorio$ curl --cookie-jar Mycookies.txt https://www.facebook.com/index.html -O
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           % Done    0     0     0    0         0         0         0         0
100 243k    0 243k    0     0 962k    0 --:--:-- --:--:-- --:--:-- 958k
```



Wget es similar, pero no necesita instalación. En este caso significa “www get”. Esta vez lo he usado para **descargar el html.index completo de mi propia página web**:



Otros comandos necesarios:

He instalado **apache** como ejemplo:

```
carlos@carlos-VirtualBox:~/Escritorio$ sudo apt install apache2
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  apache2-bin apache2-data apache2-utils libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.2-0
```

A continuación, deberíamos configurar el **cortafuegos** con **ufw** como se mostró anteriormente.


Ahora, un comando importante es **systemd**, para comprobar que se ejecuta el servicio:

```
carlos@carlos-VirtualBox:~/Escritorio$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor prese
   Active: active (running) since Thu 2021-02-18 14:57:54 CET; 58s ago
     Docs: https://httpd.apache.org/docs/2.4/
    Main PID: 5586 (apache2)
      Tasks: 55 (limit: 4649)
     Memory: 5.1M
    CGroup: /system.slice/apache2.service
            └─5586 /usr/sbin/apache2 -k start
              └─5588 /usr/sbin/apache2 -k start
                └─5589 /usr/sbin/apache2 -k start
```

Podemos **obtener la ip con el dominio** con el comando **curl** que ya hemos probado. Pero otra manera más sencilla es con el comando **"hostname -i"**:

```
carlos@carlos-VirtualBox:~/Escritorio$ hostname -I
10.0.2.15
carlos@carlos-VirtualBox:~/Escritorio$
```

Ahora ya solo tenemos que introducir la **ip** obtenida en el buscador, de manera que nuestro servidor queda listo y configurado:



Apache2 Ubuntu Default Page

ubuntu

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|   |-- ports.conf
|-- mods-enabled
|   |-- *.load
|   |-- *.conf
|-- conf-enabled
|   |-- *.conf
|-- sites-enabled
|   |-- *.conf
```

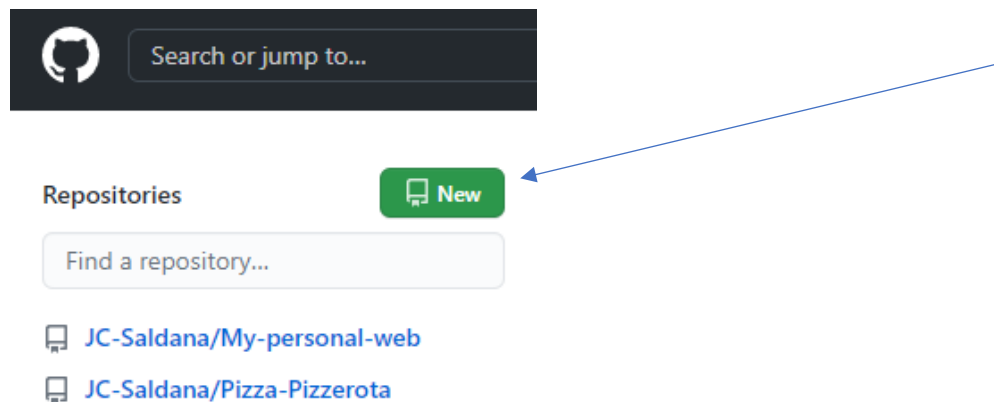
¿Servidor web o Servidor de aplicaciones?

Ya tenemos funcionando **Tomcat**, nuestro **servidor web (que no servidor de aplicaciones)**. Si quisiéramos un servidor de aplicaciones, podríamos instalar alguno como JBoss, ya que tomcat es un contenedor de servlets.

Subida a Github

Ahora voy a mostrar cómo **subir esta propia guía a un repositorio** ubicado en Github, a mi repositorio personal en este caso.

Para esto, y tras crear una cuenta, simplemente pulsamos **New** en el apartado **repositories**:



Elegimos el nombre y si lo queremos privado:

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner * / Repository name *

JC-Saldana / DespliegueAct_1_y_5 ✓

Great repository names are DespliegueAct_1_y_5 is available. [Inspiration?](#) How about [congenial-winner?](#)

Description (optional)

Este repositorio incluye la actividad 1 para usarse próximamente en la 5

☒ **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**
You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

☐ **Add a README file**
This is where you can write a long description for your project. [Learn more.](#)

☐ **Add .gitignore**
Choose which files not to track from a list of templates. [Learn more.](#)

☐ **Choose a license**
A license tells others what they can and can't do with your code. [Learn more.](#)

Create repository

Se instala **git** en nuestro equipo como con las utilidades anteriores:

```
carlos@carlos-VirtualBox:~/Escritorio$ sudo apt-get install git
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  git-man liberror-perl
Paquetes sugeridos:
  git-daemon-run | git-daemon-sysvinit git-doc git-el git-email git-gui gitk
  gitweb git-cvs git-mediawiki git-svn
```

A continuación:

- entramos en la carpeta despliegue con el comando **cd**
- **iniciamos** repositorio con **init**
- **clonamos** el repositorio que creamos anteriormente
- **entramos** en la carpeta clonada
- **creamos un txt** a modo de placeholder
- **añadimos** todos los cambios con **git --all**
- usamos **status** para ver qué hay a la espera de commit
- hacemos **commit** a lo que ya hemos añadido

```
carlos@carlos-VirtualBox: ~/Escritorio/despliegue/Despliegue_Act_1_y_5
carlos@carlos-VirtualBox:~/Escritorio$ cd despliegue
carlos@carlos-VirtualBox:~/Escritorio/despliegue$ git init
Iniciado repositorio Git vacío en /home/carlos/Escritorio/despliegue/.git/
carlos@carlos-VirtualBox:~/Escritorio/despliegue$ git clone https://github.com/JC-Saldana/Despliegue_Act_1_y_5.git
Clonando en 'Despliegue_Act_1_y_5'...
warning: Parece haber clonado un repositorio sin contenido.
carlos@carlos-VirtualBox:~/Escritorio/despliegue$ cd Despliegue_Act_1_y_5
carlos@carlos-VirtualBox:~/Escritorio/despliegue/Despliegue_Act_1_y_5$ touch placeholder.txt
carlos@carlos-VirtualBox:~/Escritorio/despliegue/Despliegue_Act_1_y_5$ git add --all
carlos@carlos-VirtualBox:~/Escritorio/despliegue/Despliegue_Act_1_y_5$ git status
En la rama master

No hay commits todavía



Cambios a ser confirmados:
  (usa "git rm --cached <archivo>..." para sacar del área de stage)
nuevo archivo: placeholder.txt

carlos@carlos-VirtualBox:~/Escritorio/despliegue/Despliegue_Act_1_y_5$ git commit -m "first commit despliegue"
[master (commit-raíz) 9f93239] first commit despliegue
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 placeholder.txt
```

Y, por último, el **push final** para enviar los datos al repositorio

```
carlos@carlos-VirtualBox:~/Escritorio/despliegue/Despliegue_Act_1_y_5$ git push
Username for 'https://github.com': JC-Saldana
Password for 'https://JC-Saldana@github.com':
Enumerando objetos: 3, listo.
Contando objetos: 100% (3/3), listo.
Escribiendo objetos: 100% (3/3), 223 bytes | 223.00 KiB/s, listo.
Total 3 (delta 0), reusado 0 (delta 0), pack-reusado 0
To https://github.com/JC-Saldana/Despliegue_Act_1_y_5.git
 * [new branch]      master -> master
carlos@carlos-VirtualBox:~/Escritorio/despliegue/Despliegue_Act_1_y_5$
```


Vemos que queda añadido en nuestra cuenta, y por tanto ya tenemos el **ejercicio completo**:

	JC-Saldana first commit despliegue ...	7 minutes ago 🕒 1
	placeholder.txt first commit despliegue	7 minutes ago