Fase 1: Instalación y configuración del servidor web
✅ Instalar Apache o Nginx y configurar el sitio por defecto en /var/www/html.
✅ Crear una página de prueba en HTML/PHP que muestre información del servidor (phpinfo();).
✅ Configurar un vhost para alojar múltiples proyectos en el mismo servidor.

Instalación y configuración del servidor web:

sudo apt update
sudo apt install apache2

Verificamos:

sudo systemctl status apache2

En mi caso, ya lo tenía instalado de los retos anteriores, adjunto captura de la verificación:

```
raulrp@srv-base-RaulRecuero:~/Desktop$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: >
     Active: active (running) since Wed 2025-06-25 14:02:57 CEST; 4min 54s ago
       Docs: https://httpd.apache.org/docs/2.4/
    Process: 1227 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SU>
   Main PID: 1256 (apache2)
      Tasks: 55 (limit: 4598)
     Memory: 7.5M (peak: 7.8M)
        CPU: 59ms
     CGroup: /system.slice/apache2.service
             ├─1256 /usr/sbin/apache2 -k start
             ├─1257 /usr/sbin/apache2 -k start
             └─1258 /usr/sbin/apache2 -k start

jun 25 14:02:57 srv-base-RaulRecuero systemd[1]: Starting apache2.service - The>
jun 25 14:02:57 srv-base-RaulRecuero apachectl[1251]: AH00558: apache2: Could n>
jun 25 14:02:57 srv-base-RaulRecuero systemd[1]: Started apache2.service - The >
```

Página de prueba:

Creamos el archivo info.php:

cd /var/www/html
sudo nano info.php

echo “<?php phpinfo(); ?>” | sudo tee /var/www/html/info.php

```
File   Edit   View   Terminal   Tabs   Help
raulrp@srv-base-RaulRecuero:~/Desktop$ cd /var/www/html
raulrp@srv-base-RaulRecuero:/var/www/html$ sudo nano info.php
```

```
Terminal - raulrp@srv-base-RaulRecuero: /var/www/html
File  Edit  View  Terminal  Tabs  Help
  GNU nano 7.2                    info.php *
echo "<?php phpinfo();?>" | sudo tee /var/www/html/info.php
```

Configuración de VirtualHosts:

Creamos un nuevo archivo en: /etc/apache2/sites-available/proyecto1.conf

cd/ etc/apache2/sites-available/
sudo nano proyecto1.conf

```
<VirtualHost *:80>
    ServerAdmin webmaster@proyecto1.com
    ServerName proyecto1.com
    ServerAlias www.proyecto1.com
    DocumentRoot /var/www/proyecto1
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

```
File  Edit  View  Terminal  Tabs  Help
raulrp@srv-base-RaulRecuero:~/Desktop$ cd ..
raulrp@srv-base-RaulRecuero:~$ cd /etc/apache2/sites-available/
raulrp@srv-base-RaulRecuero:/etc/apache2/sites-available$ sudo nano proyecto1.co
nf
```

```
  GNU nano 7.2                    proyecto1.conf
<VirtualHost *80>
   ServerAdmin webmaster@proyecto1.com
   ServerName proyecto1.com
   ServerAlias www.proyecto1.com
   DocumentRoot /var/www/proyecto1
   ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Instalación y configuración del servidor FTP:

sudo apt install vsftpd

Habilitar y arrancar servicio:

sudo systemctl enable vsftpd
sudo systemctl start vsftpd

```
raulrp@srv-base-RaulRecuero:/etc/apache2/sites-available$ sudo systemctl enable
vsftpd
Synchronizing state of vsftpd.service with SysV service script with /usr/lib/sys
temd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable vsftpd
raulrp@srv-base-RaulRecuero:/etc/apache2/sites-available$ sudo systemctl start v
sftpd
raulrp@srv-base-RaulRecuero:/etc/apache2/sites-available$
```

Creación de usuarios restringidos:

sudo adduser usuario1
sudo usermod -s /usr/sbin/nologin usuario1
sudo mkdir /home/usuario1/ftp
sudo chown nobody:nogroup /home/usuario1/ftp
sudo chmod a-w /home/usuario1/ftp
sudo mkdir /home/usuario1/ftp/files
sudo chown usuario1:usuario1 /home/usuario1/ftp/files

```
Terminal - raulrp@srv-base-RaulRecuero: /etc/apache2/sites-available
File  Edit  View  Terminal  Tabs  Help
raulrp@srv-base-RaulRecuero:/etc/apache2/sites-available$ sudo adduser usuario1
fatal: The user `usuario1' already exists.
raulrp@srv-base-RaulRecuero:/etc/apache2/sites-available$ sudo usermod -s /usr/s
bin/nologin usuario1
raulrp@srv-base-RaulRecuero:/etc/apache2/sites-available$ sudo mkdir /home/usuar
io1/ftp
raulrp@srv-base-RaulRecuero:/etc/apache2/sites-available$ sudo chown nobody:nogr
oup /home/usuario1/ftp
raulrp@srv-base-RaulRecuero:/etc/apache2/sites-available$ sudo chmod a-w /home/u
suario1/ftp
raulrp@srv-base-RaulRecuero:/etc/apache2/sites-available$ sudo mkdir /home/usuar
io1/ftp/files
raulrp@srv-base-RaulRecuero:/etc/apache2/sites-available$ sudo chown usuario1:us
uario1 /home/usuario1/ftp/files
raulrp@srv-base-RaulRecuero:/etc/apache2/sites-available$
```

Repetimos el proceso con otros 2 usuarios.

Configuración vsftpd:

Editamos el archivo:

sudo nano /etc/vsftpd.conf

local_enable=YES
write_enable=YES
chroot_local_user=YES
allow_writeable_chroot=YES
user_sub_token=$USER
local_root=/home/$USER/ftp

```
raulrp@srv-base-RaulRecuero:/$ sudo nano /etc/vsftpd.conf
```

```
  GNU nano 7.2                    /etc/vsftpd.conf *
#
# This string is the name of the PAM service vsftpd will use.
pam_service_name=vsftpd
#
# This option specifies the location of the RSA certificate to use for SSL
# encrypted connections.
rsa_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem
rsa_private_key_file=/etc/ssl/private/ssl-cert-snakeoil.key
ssl_enable=NO

#
# Uncomment this to indicate that vsftpd use a utf8 filesystem.
#utf8_filesystem=YES
local_enable=YES
write_enable=YES
chroot_local_user=YES
allow_writeable_chroot=YES
user_sub_token=$USER
local_root=/home/$USER/ftp

^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
```

Fase 3: Seguridad y monitoreo de accesos
✅ Configurar firewall (ufw o iptables) para permitir solo conexiones desde la red interna.
✅ Implementar logs para detectar accesos sospechosos y revisar /var/log/apache2/access.log.
✅ Bloquear intentos de acceso no autorizados con reglas de firewall y fail2ban.

Configurar Firewall (ufw):

Para sólo permitir tráfico local:

sudo ufw allow from 192.168.0.0/24 to any port 80
sudo ufw allow from 192.168.0.0/24 to any port 21
sudo ufw enable

```
raulrp@srv-base-RaulRecuero:/$ sudo ufw allow from 192.168.0.0/24 to any port 80
[sudo] password for raulrp:
Rule added
raulrp@srv-base-RaulRecuero:/$ sudo ufw allow from 192.168.0.0/24 to any port 21
Rule added
raulrp@srv-base-RaulRecuero:/$ sudo ufw enable
Firewall is active and enabled on system startup
raulrp@srv-base-RaulRecuero:/$
```

Logs de acceso:

tail -f /var/log/apache2/access.log

```
raulrp@srv-base-RaulRecuero:/$ tail -f /var/log/apache2/access.log
127.0.0.1 - - [25/Jun/2025:14:17:15 +0200] "GET /info.php HTTP/1.1" 404 488 "-"
"Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:140.0) Gecko/20100101 Firefox/140.0"
127.0.0.1 - - [25/Jun/2025:14:17:16 +0200] "GET /favicon.ico HTTP/1.1" 404 487 "
http://127.0.0.1/info.php" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:140.0) Ge
cko/20100101 Firefox/140.0"
127.0.0.1 - - [25/Jun/2025:14:17:38 +0200] "GET /8/info.php HTTP/1.1" 404 488 "-
" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:140.0) Gecko/20100101 Firefox/140.
0"
192.168.1.75 - - [25/Jun/2025:14:17:57 +0200] "GET /24/info.php HTTP/1.1" 404 49
1 "-" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:140.0) Gecko/20100101 Firefox/
140.0"
192.168.1.75 - - [25/Jun/2025:14:18:25 +0200] "GET /info.php HTTP/1.1" 404 491 "
-" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:140.0) Gecko/20100101 Firefox/140
.0"
192.168.1.135 - - [25/Jun/2025:14:19:00 +0200] "GET /INFO.PHP HTTP/1.1" 404 491
"-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like G
ecko) Chrome/134.0.0.0 Safari/537.36 OPR/119.0.0.0"
192.168.1.135 - - [25/Jun/2025:14:19:00 +0200] "GET /favicon.ico HTTP/1.1" 404 4
90 "http://192.168.1.75/INFO.PHP" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) App
leWebKit/537.36 (KHTML, like Gecko) Chrome/134.0.0.0 Safari/537.36 OPR/119.0.0.0
"
192.168.1.135 - - [25/Jun/2025:14:19:12 +0200] "GET /info.php HTTP/1.1" 404 491
"-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like G
```

Fail2Ban:

sudo apt install fail2ban

Crear jail personalizada:

sudo nano /etc/fail2ban/jail.local

[apache-auth]
enabled = true
port = http,https
filter = apache-auth
logpath = /var/log/apache2/error.log
maxretry = 3

```
raulrp@srv-base-RaulRecuero:~/Desktop$ sudo nano /etc/fail2ban/jail.local
```

```
GNU nano 7.2                    /etc/fail2ban/jail
[apache-auth]
enabled=true
port=http,https
filter=apache-auth
logpath= /var/log/apache2/error.log
maxretry=3
```

Justificación: Optimización del Trabajo en Codearts Solutions
La implementación de un servidor web y un servidor FTP en una infraestructura local bajo Ubuntu Server permite a Codearts Solutions mejorar significativamente su eficiencia operativa, la colaboración entre equipos y la seguridad de sus desarrollos. A continuación se detalla cómo cada componente aporta valor a la organización:

Centralización y disponibilidad de proyectos
Al alojar múltiples proyectos mediante Virtual Hosts (vhosts) en el mismo servidor web (Apache o Nginx), se logra una plataforma centralizada donde los equipos de desarrollo, diseño y QA pueden acceder a los entornos de prueba de forma inmediata, sin depender de servicios externos o configuraciones individuales en cada máquina.
Beneficios:
Reducción de tiempos de despliegue y pruebas.
Acceso uniforme desde cualquier equipo interno.
Facilidad para mostrar avances a clientes en red privada o VPN.

Flujo de trabajo eficiente con el servidor FTP
El servidor vsftpd configurado permite a los desarrolladores y diseñadores cargar y modificar archivos directamente en los directorios asignados de cada proyecto. La creación de usuarios FTP con acceso restringido mejora el orden y la trazabilidad del trabajo.
Beneficios:
Transferencias de archivos rápidas y seguras.
Separación de entornos para evitar sobrescrituras accidentales.
Soporte para herramientas estándar como FileZilla o WinSCP.

Seguridad robusta para entornos internos
Gracias al uso del firewall (UFW) y herramientas de protección como Fail2Ban, se garantiza que solo dispositivos de la red interna puedan acceder al servidor. Además, se detectan intentos de acceso no autorizados, protegiendo la confidencialidad de los proyectos.
Beneficios:
Minimización de riesgos de intrusión o pérdida de datos.
Auditoría continua a través de logs de Apache.
Cumplimiento con buenas prácticas de ciberseguridad.

Escalabilidad y control
Este entorno base es escalable para alojar nuevos proyectos o integrar herramientas como Git, CI/CD (Jenkins, GitLab CI) o paneles de control internos sin depender de terceros. También se reduce la exposición a costos adicionales en servicios en la nube para pruebas o entornos temporales.

Beneficios:
Reducción de costos operativos.
Mayor control sobre el entorno de desarrollo.
Flexibilidad para experimentar con tecnologías sin restricciones externas.

Fase 1: Análisis en tiempo real del sistema

✅ Ejecutar y analizar las siguientes herramientas:

•top → ver procesos activos, carga y uso de CPU/RAM
•htop (si no está, instalarlo)
•uptime y free -m

✅ Identificar:

•Proceso con mayor consumo de CPU
•Proceso con mayor uso de memoria
•Tiempo que lleva encendido el sistema y carga promedio

Empezamos ejecutando top:

```
top - 15:27:25 up  1:24,  1 user,  load average: 0,15, 0,07, 0,02
Tasks: 190 total,   1 running, 189 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0,2 us,  0,7 sy,  0,0 ni, 99,2 id,  0,0 wa,  0,0 hi,  0,0 si,  0,0 st
MiB Mem :   3916,1 total,   1431,9 free,    987,2 used,   1757,6 buff/cache
MiB Swap:   3916,0 total,   3916,0 free,      0,0 used.   2928,9 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
   1340 root      20   0  359100  99804  54544 S   0,7   2,5   0:13.36 Xorg
   1833 raulrp    20   0  425676  44932  29572 S   0,3   1,1   0:01.15 xfdeskt+
   4276 root      20   0       0      0      0 I   0,3   0,0   0:00.07 kworker+
   4397 raulrp    20   0  608264  41260  33024 S   0,3   1,0   0:00.73 xfce4-t+
      1 root      20   0   23072  14152   9416 S   0,0   0,4   0:03.38 systemd
      2 root      20   0       0      0      0 S   0,0   0,0   0:00.00 kthreadd
      3 root      20   0       0      0      0 S   0,0   0,0   0:00.00 pool_wo+
      4 root       0 -20       0      0      0 I   0,0   0,0   0:00.00 kworker+
      5 root       0 -20       0      0      0 I   0,0   0,0   0:00.00 kworker+
      6 root       0 -20       0      0      0 I   0,0   0,0   0:00.00 kworker+
```

Instalamos htop:

sudo apt install htop -y htop

```
raulrp@srv-base-RaulRecuero:~/Desktop$ sudo apt install htop -y htop
[sudo] password for raulrp:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
htop is already the newest version (3.3.0-4build1).
htop set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 10 not upgraded.
raulrp@srv-base-RaulRecuero:~/Desktop$
```

uptime:

```
raulrp@srv-base-RaulRecuero:~/Desktop$ uptime
 15:30:02 up  1:27,  1 user,  load average: 0,16, 0,18, 0,08
```

free -m:

```
raulrp@srv-base-RaulRecuero:~/Desktop$ free -m
               total        used        free      shared  buff/cache   available
Mem:            3916         973        1445           2        1757        2942
Swap:           3915           0        3915
```

Identificar:

Mayor uso CPU: En la columna %CPU en top o htop.
Mayor uso de memoria: En la columna %MEM o RES.
Tiempo encendido: Primera parte del output de uptime.
Carga promedio: los 3 últimos números de uptime.

```
  0[||                          1.2%] Tasks: 109, 336 thr, 89 kthr; 2 runnin
  1[                            0.0%] Load average: 0.22 0.22 0.09
Mem[||||||||||||||          714M/3.82G] Uptime: 00:02:56
Swp[                          0K/3.82G]

 Main   I/O
   PID USER      PRI  NI  VIRT   RES   SHR S  CPU%▽MEM%   TIME+  Command
  2202 raulrp     20   0  8600  4864  3712 R   3.4  0.1  0:00.66 htop
```

```
raulrp@srv-base-RaulRecuero:~/Desktop$ uptime
 16:14:13 up 3 min,  1 user,  load average: 0,10, 0,19, 0,09
raulrp@srv-base-RaulRecuero:~/Desktop$
```

Fase 2: Gestión activa de procesos y prioridades

✅ Finalizar un proceso inactivo o que no sea esencial (kill, killall o pkill).
✅ Cambiar la prioridad de un proceso en ejecución con renice.
✅ Lanzar un proceso en segundo plano (&) y enviarlo al primer plano con fg.
✅ Usar nice para iniciar un proceso con prioridad baja (por ejemplo, una copia pesada con cp).

Finalizar proceso no esencial:

Por ejemplo: pkill nano

Cambiar prioridad con renice:

sudo renice -5 -p 1234

```
raulrp@srv-base-RaulRecuero:~/Desktop$ pkill nano
raulrp@srv-base-RaulRecuero:~/Desktop$
```

```
raulrp@srv-base-RaulRecuero:~/Desktop$ sudo renice -5 -p 1234
1234 (process ID) old priority 0, new priority -5
raulrp@srv-base-RaulRecuero:~/Desktop$
```

Ejecutar proceso en segundo plano y traerlo al frente:

sleep 1000 & fg

```
raulrp@srv-base-RaulRecuero:~/Desktop$ sleep 1000 & fg
[1] 2701
sleep 1000
```

Iniciar proceso con prioridad baja:

nice -n 19 cp archivo_grande.iso /tmp/

Fase 3: Monitorización y registro del uso de recursos

✅ Usar el comando vmstat y guardar su salida en un archivo /srv/logs/vmstat.log.
✅ Configurar una tarea en crontab que guarde el uso de recursos (top -b -n 1) cada 5 minutos en /srv/logs/top.log.
✅ Explorar iotop (si el sistema lo permite) para monitorizar I/O de disco.

Guardar salida de vmstat:

Primero creamos el directorio:

sudo mkdir -p /srv/logs

vmstat > /srv/logs/vmstat.log

```
File  Edit  View  Terminal  Tabs  Help
raulrp@srv-base-RaulRecuero:~/Desktop$ sudo vmstat > /srv/logs/vmstat.log
```

Crontab para top:

crontab -e

*/5 * * * * top -b -n 1 » /srv/logs/top.log

Verificamos:

crontab -l

```
*/5 * * * top -b -n 1 >> /srv/logs/top.log
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Exe
```

Usar iotop:

Lo instalamos:

sudo apt install iotop -y

Ejecutamos:

sudo iotop:

```
Terminal - raulrp@srv-base-RaulRecuero: ~/Desktop
File  Edit  View  Terminal  Tabs  Help
Total DISK READ:         0.00 B/s | Total DISK WRITE:         0.00 B/s
Current DISK READ:       0.00 B/s | Current DISK WRITE:       0.00 B/s
    TID  PRIO  USER     DISK READ  DISK WRITE>    COMMAND
      1 be/4 root        0.00 B/s    0.00 B/s init
      2 be/4 root        0.00 B/s    0.00 B/s [kthreadd]
      3 be/4 root        0.00 B/s    0.00 B/s [pool_workqueue_release]
      4 be/0 root        0.00 B/s    0.00 B/s [kworker/R-rcu_g]
      5 be/0 root        0.00 B/s    0.00 B/s [kworker/R-rcu_p]
      6 be/0 root        0.00 B/s    0.00 B/s [kworker/R-slub_]
      7 be/0 root        0.00 B/s    0.00 B/s [kworker/R-netns]
     10 be/0 root        0.00 B/s    0.00 B/s [kworker/0:0H-kblockd]
     12 be/0 root        0.00 B/s    0.00 B/s [kworker/R-mm_pe]
     13 be/4 root        0.00 B/s    0.00 B/s [rcu_tasks_kthread]
     14 be/4 root        0.00 B/s    0.00 B/s [rcu_tasks_rude_kthread]
     15 be/4 root        0.00 B/s    0.00 B/s [rcu_tasks_trace_kthread]
     16 be/4 root        0.00 B/s    0.00 B/s [ksoftirqd/0]
     17 be/4 root        0.00 B/s    0.00 B/s [rcu_preempt]
     18 rt/4 root        0.00 B/s    0.00 B/s [migration/0]
     19 rt/4 root        0.00 B/s    0.00 B/s [idle_inject/0]
     20 be/4 root        0.00 B/s    0.00 B/s [cpuhp/0]
     21 be/4 root        0.00 B/s    0.00 B/s [cpuhp/1]
  keys:  any: refresh  q: quit  i: ionice  o: active  p: procs  a: accum
  sort:  r: asc  left: DISK READ  right: COMMAND  home: TID  end: COMMAND
CONFIG_TASK_DELAY_ACCT and kernel.task_delayacct sysctl not enabled in kernel, c
```

Fase 4: Simulación de sobrecarga controlada

- ✅ Instalar el paquete stress o stress-ng.
- ✅ Ejecutar una prueba con carga simulada de CPU, memoria o disco durante 1 minuto.
- ✅ Observar el comportamiento del sistema con htop y anotar el resultado.

Instalamos stress-ng:

sudo apt install stress-ng -y

Ejecutar carga simulada (CPU,RAM,HDD):

CPU:

stress-ng –cpu 2 –timeout 60s

RAM:

stress-ng –vm 2 –vm-bytes 512M –timeout 60s

HDD o Disco:

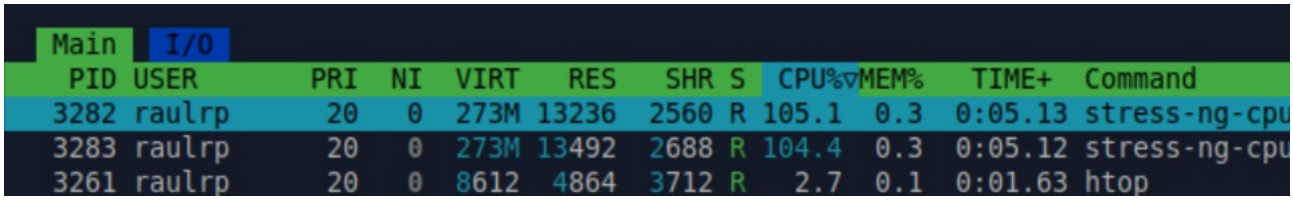
stress-ng –hdd 2 –timeout 60s

```
raulrp@srv-base-RaulRecuero:~/Desktop$ sudo apt install stress-ng -y
[sudo] password for raulrp:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libipsec-mb1 libjudydebian1 libsctp1
Suggested packages:
```

```
Terminal - raulrp@srv-base-RaulRecuero: ~/Desktop
File  Edit  View  Terminal  Tabs  Help
raulrp@srv-base-RaulRecuero:~/Desktop$ stress-ng --cpu 2 --timeout 60s
stress-ng: info:  [3238] setting to a 1 min, 0 secs run per stressor
stress-ng: info:  [3238] dispatching hogs: 2 cpu
stress-ng: info:  [3238] skipped: 0
stress-ng: info:  [3238] passed: 2: cpu (2)
stress-ng: info:  [3238] failed: 0
stress-ng: info:  [3238] metrics untrustworthy: 0
stress-ng: info:  [3238] successful run completed in 1 min, 0.26 secs
raulrp@srv-base-RaulRecuero:~/Desktop$ stress-ng --vm 2 --vm-bytes 512M --timeou
t 60s
stress-ng: info:  [3241] setting to a 1 min, 0 secs run per stressor
stress-ng: info:  [3241] dispatching hogs: 2 vm
stress-ng: info:  [3241] skipped: 0
stress-ng: info:  [3241] passed: 2: vm (2)
stress-ng: info:  [3241] failed: 0
stress-ng: info:  [3241] metrics untrustworthy: 0
stress-ng: info:  [3241] successful run completed in 1 min, 0.30 secs
raulrp@srv-base-RaulRecuero:~/Desktop$ stress-ng --hdd 2 --timeout 60s
stress-ng: info:  [3246] setting to a 1 min, 0 secs run per stressor
stress-ng: info:  [3246] dispatching hogs: 2 hdd
stress-ng: info:  [3246] skipped: 0
stress-ng: info:  [3246] passed: 2: hdd (2)
stress-ng: info:  [3246] failed: 0
stress-ng: info:  [3246] metrics untrustworthy: 0
```

Y observamos con htop:

htop



Main I/O

PID	USER	PRI	NI	VIRT	RES	SHR	S	CPU%▽	MEM%	TIME+	Command
3282	raulrp	20	0	273M	13236	2560	R	105.1	0.3	0:05.13	stress-ng-cpu
3283	raulrp	20	0	273M	13492	2688	R	104.4	0.3	0:05.12	stress-ng-cpu
3261	raulrp	20	0	8612	4864	3712	R	2.7	0.1	0:01.63	htop