

SERVICIOS TELEMATICOS

PROYECTO HAProxy - PRUEBAS DE CARGA

Santiago Fajardo Banguero - 2200401
Oscar David Serna Zarate - 2151629



TABLA CONTENIDO

FASE 1: HAPROXY

- ¿Qué es Haproxy?
- Características
- Instalar
- Archivo y explicación.
- Implementacion

FASE 2: ARTILLERY

- ¿Qué es Artillery?
- Caracteristicas
- Pruebas comunes

FASE 3: PRUEBAS

- Pruebas con Artillery y Haproxy



HAProxy es un software de balanceo de carga y proxy de nivel 7 de código abierto y alta disponibilidad. Se utiliza comúnmente en entornos de servidores web para distribuir la carga de tráfico entre múltiples servidores backend, lo que permite mejorar el rendimiento y la capacidad de respuesta de los servicios en línea.



- PERMITE SSL CONFIGURADO EN HAProxy Y/O EN EL BACKEND
- SOPORTA IPV6
- ADMITE ACL
- SOPORTA VIRTUALHOSTS
- PUEDE CONFIGURAR HEALTHCHECKS O COMPROBACION DE ESTADO DE LOS BACKENDS
- PERMITE HTTP KEEP-ALIVE

FUENTE: APRENDEIT



TIPOS DE BALANCEO:

ROUND ROBIN



LEASTCONN



SOURCE



FUENTE: APRENDEIT



VAGRANTFILE

```
# -- mode: ruby --
# vi: set ft=ruby :
Vagrant.configure("2") do |config|
  config.vm.define :servidor1 do |servidor1|
    servidor1.vm.box = "generic/centos8"
    servidor1.vm.network :private_network, ip: "192.168.50.2"
    servidor1.vm.hostname = "servidor1"
  end
  config.vm.define :servidor2 do |servidor2|
    servidor2.vm.box = "generic/centos8"
    servidor2.vm.network :private_network, ip: "192.168.50.3"
    servidor2.vm.hostname = "servidor2"
  end
  config.vm.define :haproxy do |haproxy|
    haproxy.vm.box = "generic/centos8"
    haproxy.vm.network :private_network, ip: "192.168.50.4"
    haproxy.vm.hostname = "haproxy"
  end
end
```

FUENTE: ELABORACION PROPIA



INSTALANDO HAProxy:

YUM INSTALL HAProxy

```
CentOS Linux 8 - BaseOS
CentOS Linux 8 - Extras
Extra Packages for Enterprise Linux 8 - x86_64
Extra Packages for Enterprise Linux 8 - x86_64
Dependencies resolved.

=====
 Package
=====

Installing:
haproxy

Transaction Summary
=====

Install 1 Package

Total download size: 1.4 M
Installed size: 4.2 M
```

CD /ETC/HAPROXY/

```
[root@centos8 haproxy]# ls
haproxy.cfg
```

VIM HAPROXY.CFG

FUENTE: ELABORACION PROPIA

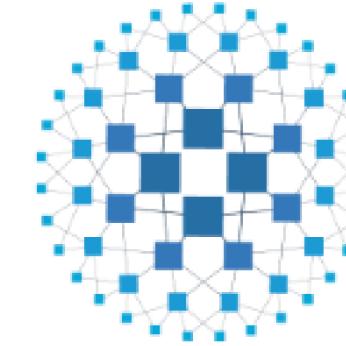


FUENTE: ELABORACION PROPIA



```
#configuracion global  
global—> CONEXION GLOBAL / SERVIDOR  
  
log 127.0.0.1 local2—> IP / SISTEMA REGISTRO  
  
chroot /var/lib/haproxy—> DONDE SE EJECUTA HAProxy  
pidfile /var/run/haproxy.pid—> PID  
maxconn 4000—> CONEXIONES SIMULTÁNEAS  
user haproxy  
group haproxy—> SEGUNDO PLANO EJECUCION  
  
# turn on stats unix socket  
stats socket /var/lib/haproxy/stats—> ESTADISTICAS SOCKET  
  
# utilize system-wide crypto-policies  
ssl-default-bind-ciphers PROFILE=SYSTEM  
ssl-default-server-ciphers PROFILE=SYSTEM
```

```
#-----  
# common defaults that all the  
# use if not designated in their block  
#-----  
  
defaults  
    mode http  
    log global  
    option httplog  
    option dontlognull  
    option http-server-close  
    option forwardfor  
    option redispatch  
    retries 3  
    timeout http-request 10s  
    timeout queue 1m  
    timeout connect 10s  
    timeout client 1m  
    timeout server 1m  
    timeout http-keep-alive 10s  
    timeout check 10s  
    maxconn 3000
```



HAPROXY

http CONEXION GLOBAL / SERVIDOR

global REGISTROS

httplog REGISTROS DETALLADOS

dontlognull SIN SOLICITUDES VACIOS

option http-server-close RESPUESTA/CIERRA

option forwardfor except 127.0.0.0/8

option redispatch

retries 3

timeout http-request 10s

timeout queue 1m

timeout connect 10s

timeout client 1m

timeout server 1m

timeout http-keep-alive 10s

timeout check 10s

maxconn 3000

→ CONEXION CONCURRENTE

FUENTE: ELABORACION PROPIA



FUENTE: ELABORACION PROPIA

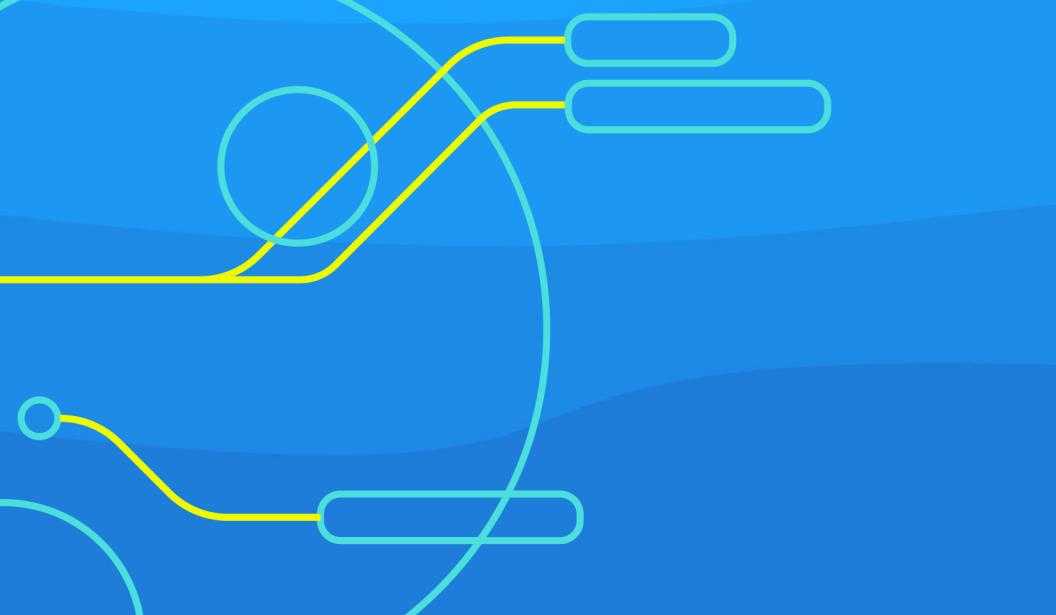
```
frontend stats
  bind 192.168.50.5:5000
  mode http
  log global

  maxconn 10  CONEXIONES CONCURRENTES

  clitimeout    100s
  srvtimeout    100s
  contimeout    100s
  timeout queue  100s

  stats enable  HABILITACION DE ESTADISTICAS
  stats hide-version
  stats refresh 30s
  stats show-node
  stats auth admin:password
  stats uri /haproxy?stats
```

IP/PUERTO
MODO HTTP
GLOBAL EVENTOS

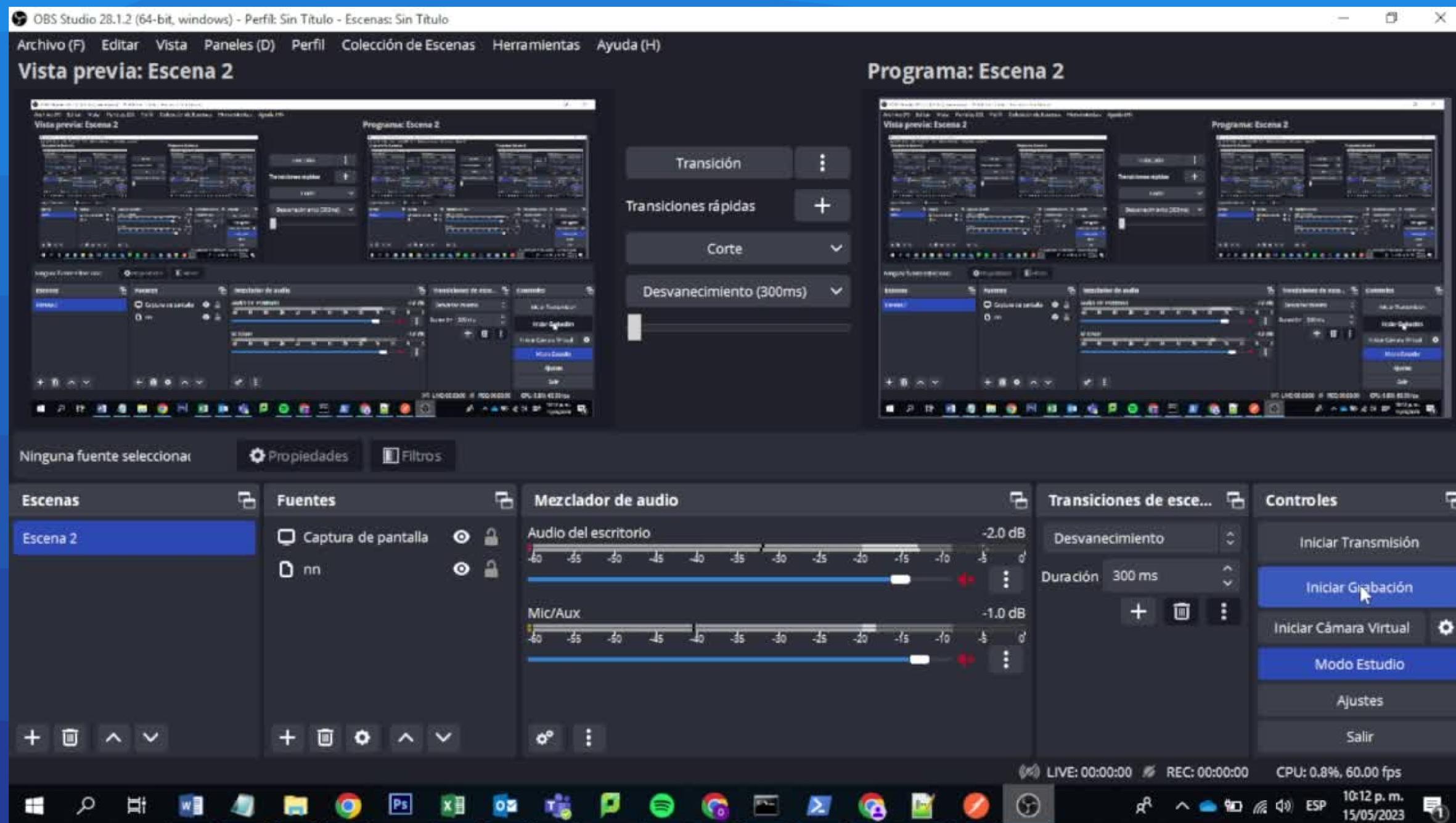


FUENTE: ELABORACION PROPIA

```
#-----  
# main frontend which proxys to the backends  
#-----  
frontend main  
    bind *:80  
    # acl url_static      path_beg      -i /static /images /javascript /sty  
    default_backend      app  
        LISTEN  
        BACKEND A ELEGIR  
#-----  
# static backend for serving up images, stylesheets and such  
#-----  
backend static  
    balance roundrobin  
    server  static 127.0.0.1:4331 check  
#-----  
# round robin balancing between the various backends  
#-----  
backend app  
    balance roundrobin  
    server app1 192.168.50.3:80 check  
    server app2 192.168.50.4:80 check  
~  
~  
~
```

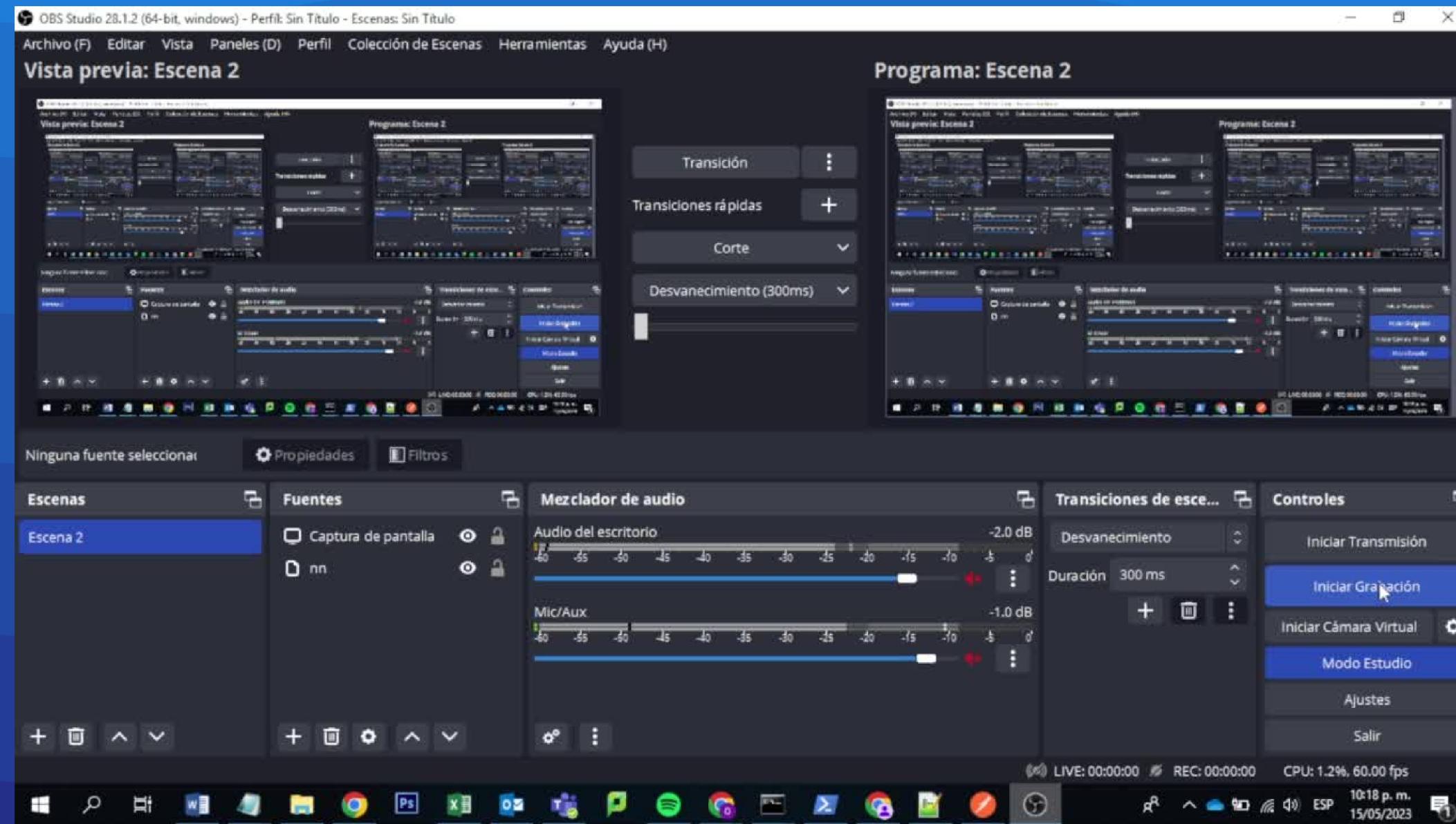
BACKEND A UTILIZAR

VIDEOS BALANCEADOR - RR

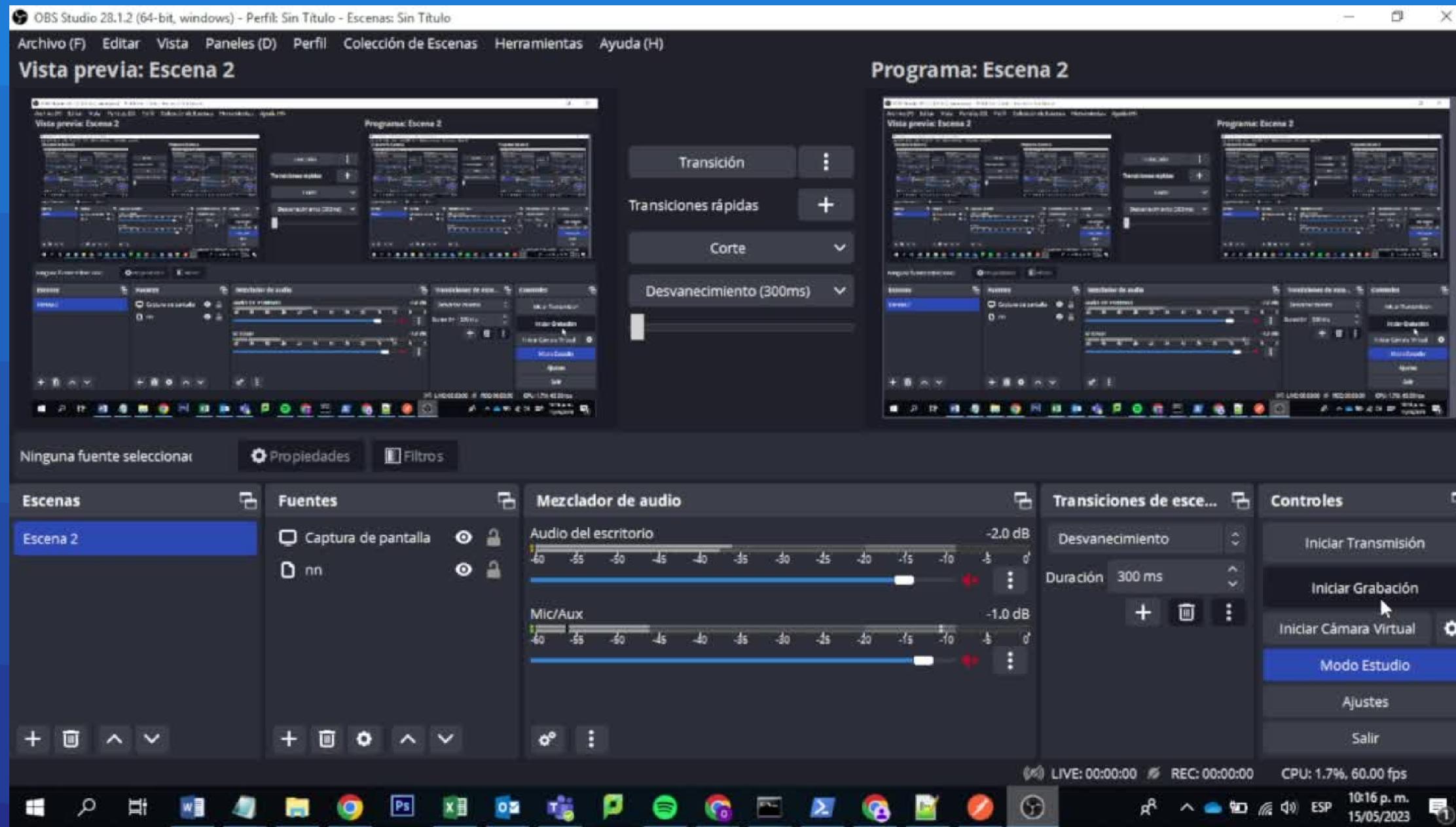


FUENTE: ELABORACION
PROPIA

VIDEOS BALANCEADOR- SOURCE



VIDEOS BALANCEADOR- LEASTCONN



FUENTE: ELABORACION
PROPIA



ARTILLERY ES UNA HERRAMIENTA DE CÓDIGO ABIERTO UTILIZADA PARA REALIZAR PRUEBAS DE CARGA Y PRUEBAS DE ESTRÉS EN APLICACIONES Y SERVICIOS WEB. SU OBJETIVO PRINCIPAL ES SIMULAR CARGAS DE USUARIOS CONCURRENTES Y MEDIR EL RENDIMIENTO Y LA CAPACIDAD DE RESPUESTA DE UN SISTEMA BAJO CONDICIONES DE ALTA CARGA.

FUENTE: ARTILLERY



Artillery

PRUEBAS MAS COMUNES:

PRUEBAS DE CARGA:



PRUEBAS DE ESTRÉS:



FUENTE: ARTILLERY

MEDICIÓN DE RENDIMIENTO:



```

http.codes.200: ..... 3400
http.request_rate: ..... 46/sec
http.requests: ..... 3400
http.response_time:
    min: ..... 2
    max: ..... 55
    median: ..... 5
    p95: ..... 15
    p99: ..... 24.8
http.responses: ..... 3400
vusers.completed: ..... 3400
vusers.created: ..... 3400
vusers.created_by_name.0: ..... 3400
vusers.failed: ..... 0
vusers.session_length:
    min: ..... 4.4

```

HAProxy

Statistics Report for pid 17929 on haproxy

> General process information

pid = 17929 (process #1, nbproc = 1, nbthread = 1)
 uptime = 0d 0h23m28s
 system limits: memmax = unlimited; ulimit-n = 6038
 maxsoek = 8038; maxconn = 4000; maxpipes = 0
 current conn = 2; current pipes = 0/0; conn rate = 2/sec
 Running tasks: 1/12; idle = 100 %
 Note: "NOLB"/"DRAIN" = UP with load-balancing disabled.

Display option:																			External resources:													
																			Scope : <input type="text"/> Primary site Hide DOWN servers Disable refresh Refresh now CSV export Updates (v1.8) Online manual													
stats																																
Queue	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Denied	Errors	Conn	Req	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtile		
Frontend	1	2	-	1	2	10	7			1 979	1 472	0	0	2								OPEN										
main																																
Queue	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Denied	Errors	Conn	Req	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtile		
Frontend	1	2	-	1	2	3 000	4			15 084	4 805	0	0	2									OPEN									
static																																
Queue	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Denied	Errors	Conn	Req	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtile		
static	0	0	-	0	0	0	0	0	-	0	0	7	0	0	0	0	0	0	0	0	23m28s DOWN	L4CON in 0ms	1	Y	-	1	1	23m28s	-			
Backend	0	0	0	0	0	0	0	0	1	0	0	?	0	0	0	0	0	0	0	0	23m28s DOWN		0	0	0	0	1	23m28s				
app																																
Queue	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Denied	Errors	Conn	Req	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtile		
app1	0	0	-	0	0	0	0	0	-	0	0	?	0	0	0	0	0	0	0	0	23m26s UP	L4OK in 0ms	1	Y	-	0	0	0s	-			
app2	0	0	-	0	3	0	1	300	28	28	23m12s	15 084	4 181	0	0	0	0	0	0	0	0	23m28s UP	L4OK in 0ms	1	Y	-	0	0	0s	-		
Backend	0	0	0	0	3	0	1	300	28	28	23m12s	15 084	4 181	0	0	0	0	0	0	0	0	23m26s L	?	2	Y	-	0	0	0s	-		

Summary report @ 05:30:30(-0500)

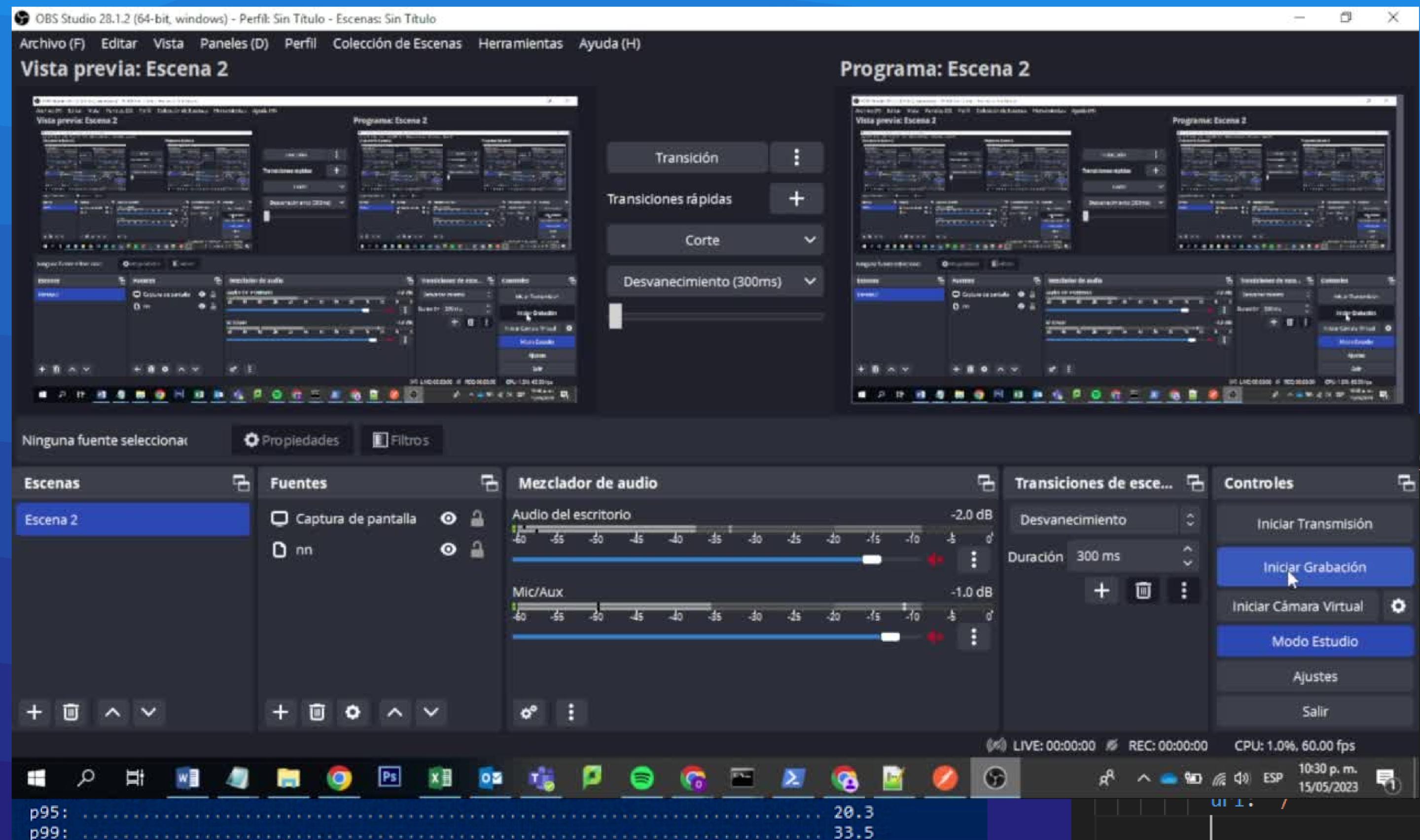
```

errors.ETIMEDOUT: ..... 2176
tp.codes.404: ..... 7949
tp.dns: ..... 0
min: ..... 535
max: ..... 56.3
median: ..... 3
p95: ..... 153
p99: ..... 254/sec
tp.request_rate: ..... 10125
tp.requests: ..... 12
tp.response_time:
    min: ..... 8332
    max: ..... 550.1
    median: ..... 3197.8
    p95: ..... 4965.3
    p99: ..... 7949
tp.responses: ..... 54
tp.tcp: ..... 8259
min: ..... 539.2
max: ..... 1755
median: ..... 2836.2
p95: ..... 113
p99: ..... 9447
tp.tls: ..... 1200.1
min: ..... 5598.4
max: ..... 7407.5
median: ..... 9801.2
p95: ..... 241
p99: ..... 10029
sers.completed: ..... 2893.5
sers.created: ..... 8868.4
sers.created_by_name.0: ..... 9801.2
sers.failed: ..... 2176
sers.session_length:
    min: ..... 246.4
    max: ..... 10252.6
    median: ..... 3072.4
    p95: ..... 8868.4
    p99: ..... 9801.2

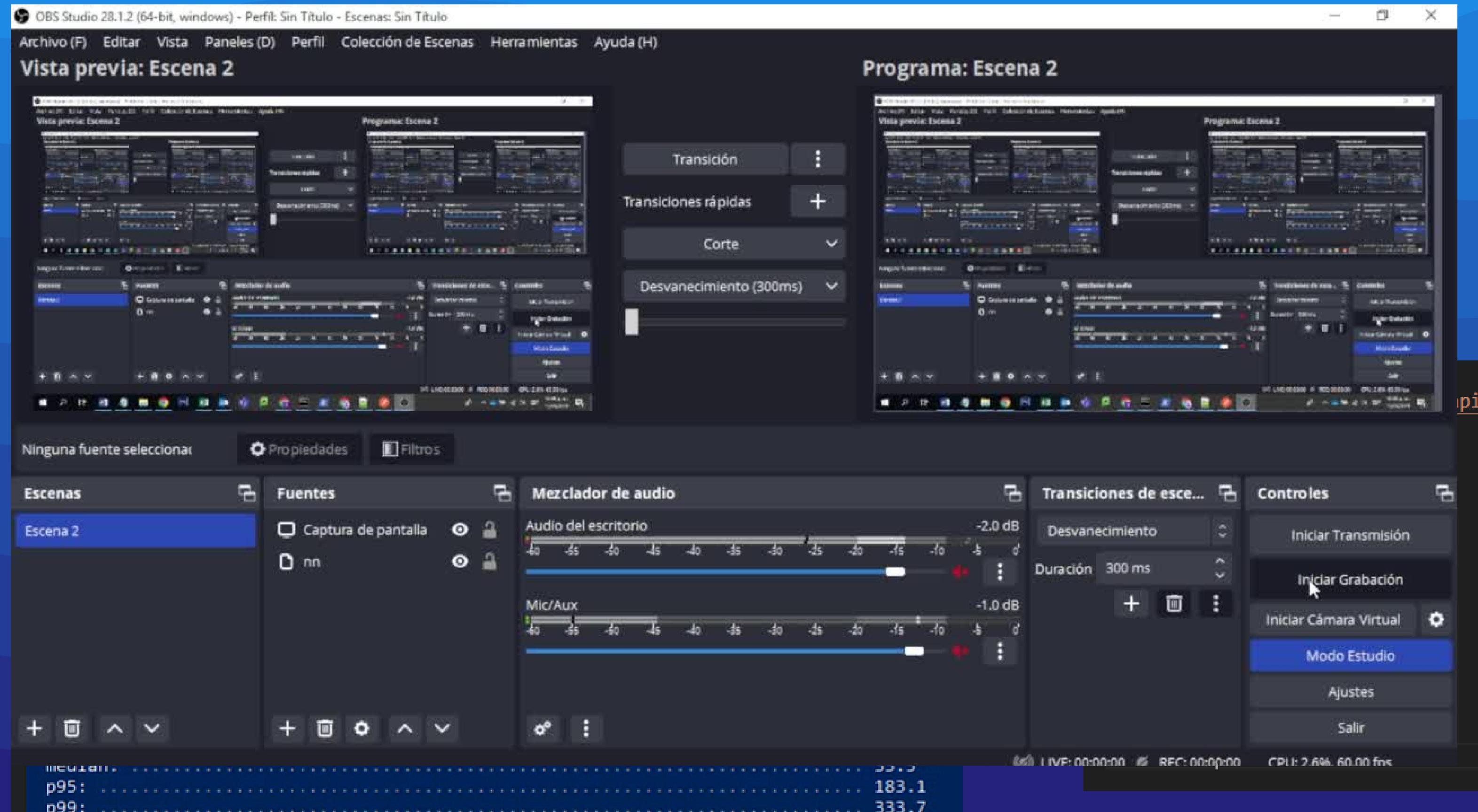
```

FUENTE: ELABORACION PROPIA

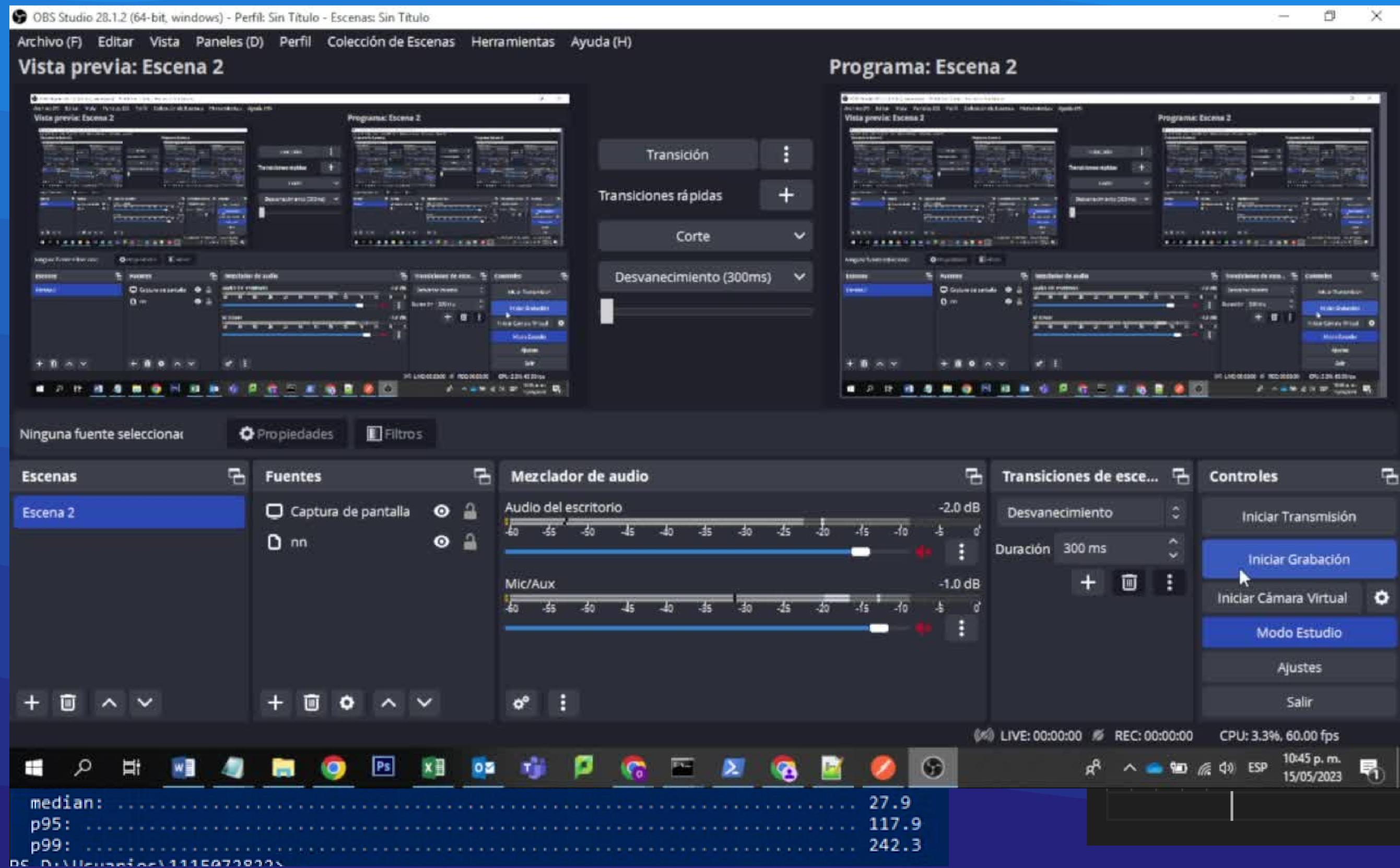
PRUEBAS DE CARGA ARTILLERY- RR



PRUEBAS DE CARGA ARTILLERY - LEASTCONN



PRUEBAS DE CARGA ARTILLERY- SOURCE

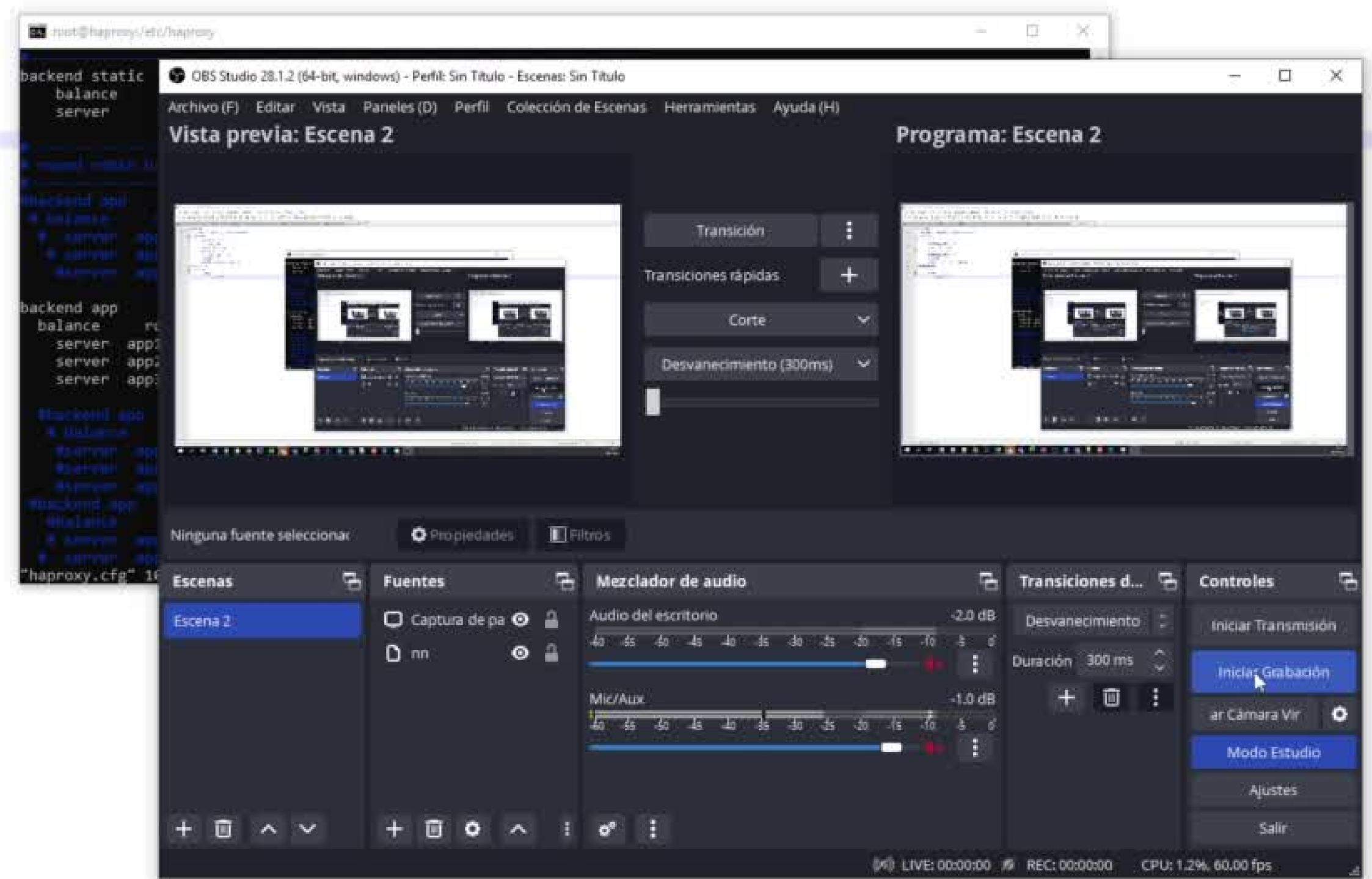


ole.com/api"

)-up

BASES DE DATOS CARGA

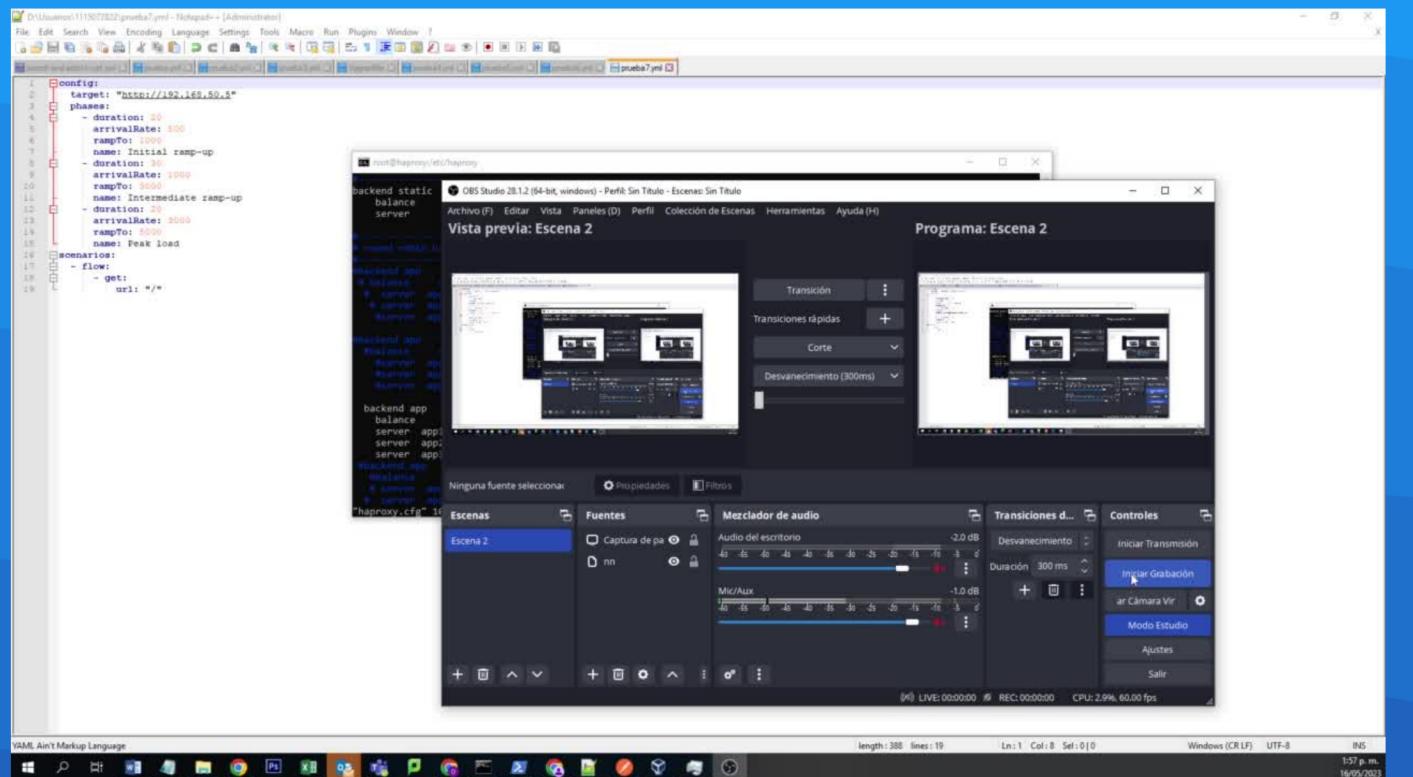
```
 1 <!--fig:
 2   <!--args: http://12.16.1.1
 3   <!--has:
 4     - duration: 10
 5       arrivalRate: 20
 6       name: Warm up
 7     - duration: 10
 8       arrivalRate: 20
 9       rampTo: 400
10       name: Gradual ramp-up
11 
12 <!--scenarios:
13   - flow:
14     - get:
15       url: /*
```



PRUEBAS DE ESTRÉS SERVIDORES

```
Summary report @ 14:06:20(-0500)

errors.EADDRINUSE: 82459
errors.ECONNRESET: 43
errors.ETIMEDOUT: 3332
http.codes.200: 68972
http.request_rate: 330/sec
http.requests: 154787
http.response_time:
  min: 0
  max: 9626
  median: 1326.4
  p95: 4965.3
  p99: 6439.7
http.responses: 68972
vusers.completed: 68953
vusers.created: 154787
vusers.created_by_name.0: 154787
vusers.failed: 85834
vusers.session_length:
  min: 17.9
  max: 12578.8
  median: 2018.7
  p95: 8868.4
  p99: 10407.3
```



```
config:
  http:
    extendedMetrics: true
  target: "https://192.168.50.5"
  phases:
    - duration: 20
      arrivalRate: 500
      rampTo: 1000
      name: Initial ramp-up
    - duration: 30
      arrivalRate: 1000
      rampTo: 3000
      name: Intermediate ramp-up
    - duration: 20
      arrivalRate: 3000
      rampTo: 5000
      name: Peak load
  scenarios:
    - flow:
        - get:
            url: "/"
```





REFERENCIAS



1. A. Peña, "Enrutamiento y balanceo básico HTTP con HAProxy", Aprende IT, [En línea]. Disponible en: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut a enim nec nisl ullamcorper eleifend.
Praesent risus leo, fringilla et nulla at, egestas euismod orci.
Suspendisse porttitor diam eu condimentum aliquam. Fusce interdum
cursus nisl ut rutrum.. [Accedido: Día mes año].
2. Autor desconocido, "Título del artículo", Título de la publicación, vol. X, no. X, pp. XX-XX, mes año. DOI: XXXX
3. Nota: No se proporcionó suficiente información para generar una referencia precisa para el segundo enlace. Por favor, proporcione más detalles si es necesario.
4. Autor desconocido, "ThingWorx High Availability: HAProxy Example", Worx Help, [En línea]. Disponible en: [URL]. [Accedido: Día mes año].



GRACIAS!