

## Desafío 1

```
docker@root:~$ ./server
----- DESAFIO -----
Bienvenidos al TP3 y felicitaciones, ya resolvieron el primer acertijo.

En este TP deberán finalizar el juego que ya comenzaron resolviendo los desafíos de cada nivel.
Además tendrán que investigar otras preguntas para responder durante la defensa.
El desafío final consiste en crear un programa que se comporte igual que yo, es decir, que provea los mismos desafíos y
que sea necesario hacer lo mismo para resolverlos. No basta con esperar la respuesta.
Además, deberán implementar otro programa para comunicarse conmigo.

Deberán estar atentos a los easter eggs.

Para verificar que sus respuestas tienen el formato correcto respondan a este desafío con la palabra 'entendido\n'

----- PREGUNTA PARA INVESTIGAR -----
¿Cómo descubrieron el protocolo, la dirección y el puerto para conectarse?
```

Nos conectamos con “netcat 0.0.0.0 8080”, dado que vimos el uso de ese cliente en clase. Para obtener la dirección y el puerto, corrimos el archivo “server” provisto con “strace”, donde se ve la dirección, puerto y protocolo (SOCK\_STREAM = TCP) usados.

```

mpipelect(0x7f041c98900, 4096, PROT_READ) = 0
mmap(0x7f041c9a1000, 22954) = 0
socket(PF_INET, SOCK_STREAM, IPPROTO_IP) = 3
setsockopt(3, SOL_SOCKET, SO_REUSEPORT, [1], 1)
bind(3, {sa_family=AF_INET, sin_port=htons(8080), sin_addr=inet_addr("0.0.0.0")}, 16) = 0
listen(3, 5)
accept(3,

```

## Desafío 2

```
----- DESAFIO -----
The Wire S1E5
5295 888 6288

----- PREGUNTA PARA INVESTIGAR -----
¿Qué diferencias hay entre TCP y UDP y en qué casos conviene usar cada uno?
```

La principal diferencia entre TCP y UDP pasa fundamentalmente por el sistema de verificación de la transmisión de la información entre el dispositivo emisor y el dispositivo receptor.

El protocolo TCP es un protocolo de transporte orientado a conexión, mientras que el protocolo UDP no lo es. De esta manera, el protocolo TCP verifica la correcta transmisión de los datos entre los dispositivos emisores y los receptores, mientras que el protocolo UDP no lo hace.

Conviene usar UDP cuando se requiere gran velocidad de transmisión y no es un problema que se pierda algo de información (por ejemplo en streaming de video), mientras que es preferible usar TCP cuando la integridad de la información es una prioridad.

Vimos la serie y el método para desencriptar el código que presentan. El código desencriptado según ese método es: 0810 222 4822, que al googlearlo nos apareció el teléfono del ITBA. Entonces, pusimos como respuesta: "itba".

### Desafío 3

```
----- DESAFIO -----
https://ibb.co/tc0Hb6w

----- PREGUNTA PARA INVESTIGAR -----
¿El puerto que usaron para conectarse al server es el mismo que usan para mandar las respuestas? ¿Por qué?
```

Para enviar las respuestas, el cliente escribe a un fd.

El cliente de netcat usa su propio puerto, el cual depende de la PC y se puede ver corriendo el siguiente comando: "lsof -nP -iTCP -sTCP:ESTABLISHED"

Al ver que el desafío era una foto en blanco, sospechábamos que había algo dentro en la imagen. Primero intentamos ver si había algo dentro del código del PNG, pero no había nada raro. Luego editamos la imagen con paint, y al pintarla de negro vimos la clave: M4GFKZ289aku.

### Desafío 4

```
----- DESAFIO -----
EBADF...

write: Bad file descriptor

----- PREGUNTA PARA INVESTIGAR -----
¿Qué útil abstracción es utilizada para comunicarse con sockets? ¿se puede utilizar read(2) y write(2) para operar?
```

Se utilizan los file descriptors como abstracción, sobre los cuales se lee o escribe con read(2) y write(2).

Al leer el desafío, nos dice que el file descriptor es incorrecto, por lo que corrimos el servidor con strace y vimos que estaba intentado realizar un write en el file descriptor 13, se nos ocurrió entonces crear el file descriptor 13 y hacer que sea un archivo txt para que en vez de intentar imprimir en un file descriptor inexistente, escriba sobre el .txt.

Investigando cómo realizar esto, encontramos esta página "<https://copyconstruct.medium.com/bash-redirection-fun-with-descriptors-e799ec5a3c16>" en donde muestra como redireccionar un file descriptor a un archivo, por lo que corriendo el servidor con "./server 13>a.txt" pudimos hacer que el servidor cuando intente escribir en el file descriptor 13, escriba en el archivo a.txt que ya habíamos creado. De ahí obtuvimos el siguiente mensaje:

```
.....La respuesta es fk3wflCm3QvS
.....La respuesta es fk3wflCm3QvS
|
```

## Desafío 5

```
----- DESAFIO -----
respuesta = strings:277

----- PREGUNTA PARA INVESTIGAR -----
¿Cómo garantiza TCP que los paquetes llegan en orden y no se pierden?
```

Método: mediante checksum

Checksum es un código de detección de errores que tiene como propósito principal detectar cambios accidentales en una secuencia de datos para proteger la integridad de estos datos, verificando que no haya discrepancias entre los valores en la comprobación inicial y final.

La primera idea que nos surgió fue que tenía que ver algo con la página del “man strings”, intentamos con la letra número 277 y con la palabra 277, ya que no tenía 277 líneas.

Al ver que el camino no era por acá, se nos ocurrió correr el comando strings con el server, y ahí vimos que tenía 277 líneas (pipeando a wc) y la línea 277 era “too\_easy”, la cual resultó ser la clave.

## Desafío 6

```
----- DESAFIO -----
.data .bss .comment ? .shstrtab .symtab .strtab

----- PREGUNTA PARA INVESTIGAR -----
Un servidor suele crear un nuevo proceso o thread para atender las conexiones entrantes. ¿Qué conviene más?
```

Los servidores que utilizan procesos, pueden sufrir de problemas de context switching y los costos de los mecanismos de IPC y los que utilizan threads pueden tener problemas de comunicación al compartir todos los recursos. En cuanto a eficiencia utilizar threads es mejor pero puede tener problemas de comunicación por lo que una combinación de ambos suele ser la mejor opción.

Al ver que eran secciones de assembler, probamos las distintas secciones que existen pero ninguna era.

Reusando el comando strings del ej anterior, en “strings server” vimos una sección dudosa llamada “.RUN\_ME”, por lo que probamos si era la respuesta y lo era.

## Desafío 7

```
----- DESAFIO -----
Filter error

LvL++C&v%a %re@?/9F#k}AsG.3rCtHpGn<\ME=j/b,Z@u&iuV@e7IHJw_MeNtu;I6st;1ap%iYW(O_{*}l+ Ce`y Zs03+V3% LXDk=UE$p26x5n9X2cU<T
5Fj0voief%7RWP>YE_W{JE7`z&~|aq8m5'~BP>y&^"_pA~^*\FFAwLT25dM6bY?(Xv2MU%AL4Cv7Kkx|^_Lhy6GjBONdD.T=o3:)y

----- PREGUNTA PARA INVESTIGAR -----
¿Cómo se puede implementar un servidor que atienda muchas conexiones sin usar procesos ni threads?
```

Mediante el uso de sockets, donde los sockets en el servidor esperan por peticiones de un cliente. El servidor establece una dirección que el cliente puede utilizar para conectarse al servidor. Cuando la dirección es establecida, el servidor espera al cliente para que haga una petición por un servicio, para que luego el servidor le mande la respuesta al cliente.

Primero, al correr muchas veces vimos que el filtro iba cambiando pero vimos algunas coincidencias entre distintos output, consideramos analizar las coincidencias entre muchos output y pudimos filtrar la parte de “La respuesta es” pero el resto era muy complejo, por lo que investigando más y siguiendo la pista de “Filter error” se nos ocurrió que podía estar sucediendo algún error en el código, y para verlo redirigimos stderr a un archivo. Resultó ser que nos quedo basura en el archivo pero la respuesta en el servidor, la cual era “K5n2UFfpFMUN”.

## Desafío 8

```
----- DESAFIO -----
¿?

----- PREGUNTA PARA INVESTIGAR -----
¿Qué aplicaciones se pueden utilizar para ver el tráfico por la red?
```

Algunas de las aplicaciones que se pueden utilizar para ver el tráfico por la red son:

- Wireshark
- Capsa
- ManageEngine OpManager
- IPTráf
- NLoad
- Iftop

Este desafío lo resolvimos de casualidad, ya que yendo a copiar el texto del desafío para implementarlo luego en nuestro servidor, también se copia cual es la respuesta. Al pegar el contenido de la consola en un documento se pega:

```
----- DESAFIO -----
¿?
```

La respuesta es BUmyYq5XxXGt

```
----- PREGUNTA PARA INVESTIGAR -----
¿Qué aplicaciones se pueden utilizar para ver el tráfico por la red?
```

## Desafío 9

```
----- DESAFIO -----
Latexme

Si

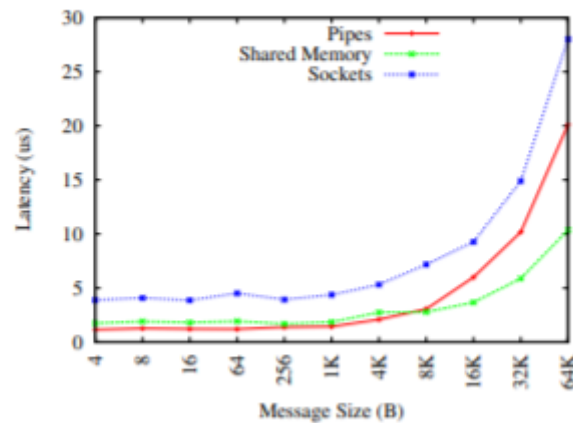
$$\mathrm{d}y = u^v \cdot (v' \cdot \ln(u) + v \cdot \frac{u'}{u})$$

entonces
y =

----- PREGUNTA PARA INVESTIGAR -----
sockets es un mecanismo de IPC. ¿Qué es más eficiente entre sockets y pipes?
```

Como mecanismo de IPC local, el mecanismo de pipes es más eficiente en cuanto a tiempo. En el siguiente gráfico, obtenido de un paper realizado en la universidad de Wisconsin-Madison en donde comparan pipes, sockets y shared memory, se puede ver como los pipes tienen menor latencia que los sockets al transferir datos.

**Figure 1: Latency vs Message Size**



[http://pages.cs.wisc.edu/~adityav/Evaluation\\_of\\_Inter\\_Process\\_Communication\\_Mechanisms.pdf](http://pages.cs.wisc.edu/~adityav/Evaluation_of_Inter_Process_Communication_Mechanisms.pdf)

En esta otra fuente: [https://wiki.openlighting.org/index.php/Pipe\\_vs\\_Unix\\_Socket\\_Performance](https://wiki.openlighting.org/index.php/Pipe_vs_Unix_Socket_Performance), se puede ver como en los distintos sistemas, el uso de pipes local es bastante mas eficiente en cuanto a latencia.

La ventaja que tiene sockets es que pueden enviar datos entre distintos hosts mediante diferentes protocolos como IPv4 e IPv6.

Este desafío nos muestra una función y la pista “Latexme”, intuimos que era una fórmula en formato LATEX, por lo que al traducirla en un editor de ecuaciones LATEX nos da lo siguiente:

$$dy = u^v \cdot (v' \cdot \ln(u) + v \cdot \frac{u'}{u})$$

Poniendo la expresión en una calculadora de integrales, la respuesta nos quedó  $u^v$ .

## Desafío 10

```

----- DESAFIO -----
quine

gcc: error: quine.c: No such file or directory
gcc: fatal error: no input files
compilation terminated.

ENTER para reintentar.

----- PREGUNTA PARA INVESTIGAR -----
¿Cuáles son las características del protocolo SCTP?

```

SCTP es un protocolo de comunicación como alternativa de TCP y UDP que provee confiabilidad, control de flujo y secuenciación como TCP pero es un protocolo orientado al mensaje, siendo más similar al envío de datagramas de UDP.

SCTP provee:

- Soporte de multihoming
- Soporte de multistreaming
- Orientado a la conexión
- Delimitadores de mensajes

Lo primero que intentamos fue hacer un archivo “quine.c” vacío. Al hacerlo, nos tiro:

```

----- DESAFIO -----
quine

/usr/lib/gcc/x86_64-linux-gnu/4.9/../../../../x86_64-linux-gnu/crt1.o: In function `_start':
/build/glibc-6V9RKT/glibc-2.19/csu/../sysdeps/x86_64/start.S:118: undefined reference to `main'
collect2: error: ld returned 1 exit status

ENTER para reintentar.

----- PREGUNTA PARA INVESTIGAR -----
¿Cuáles son las características del protocolo SCTP?

```

Luego intentamos ponerle un main al archivo, y nos tiró lo siguiente:

```

----- DESAFIO -----
quine

¡Genial!, ya lograron meter un programa en quine.c, veamos si hace lo que corresponde.
0a1,5
> int main(void) {
>
>     return 0;
>
> }
\ No newline at end of file

diff encontró diferencias.
ENTER para reintentar.

----- PREGUNTA PARA INVESTIGAR -----
¿Cuáles son las características del protocolo SCTP?

```

Tratamos de modificar el código para ver si dejaba de tirar “diff encontró diferencias” pero no lo logramos.

Luego investigamos en internet que es un quine program, y entendimos que teníamos que tener un programa que imprima su código fuente para que diff no encuentra diferencia entre el output y el código del programa.

```

C quine.c  X
C quine.c > main()
1  #include <stdio.h>
2  int main(){
3  char*s="#include <stdio.h>%cint main(){%cchar*s=%c%c;%cprintf(s,10,10,34,s,34,10);return 0;}";
4  printf(s,10,10,34,s,34,10);return 0;}

----- DESAFIO -----
quine

¡Genial!, ya lograron meter un programa en quine.c, veamos si hace lo que corresponde.
La respuesta es chin_chu_lan_cha

----- PREGUNTA PARA INVESTIGAR -----
¿Cuáles son las características del protocolo SCTP?

```

La respuesta entonces queda: chin\_chu\_lan\_cha.

## Desafío 11

```

----- DESAFIO -----
b gdbme y encontrará el valor mágico

ENTER para reintentar.

----- PREGUNTA PARA INVESTIGAR -----
¿Qué es un RFC?

|

```

Los RFC son una serie de publicaciones que describen aspectos del funcionamiento de internet y otras redes de computadoras, como protocolos y procedimientos, y también comentarios e ideas sobre estos.

La pista del desafío menciona de hacer un breakpoint en gdbme, entonces probamos realizar eso pero al llegar al desafío del filter nos cerraba el gdb ya que realizaba un kill. Entonces luego del desafío del filter iniciamos gdb de nuevo mediante el pid con la consola, creabamos el breakpoint en gdbme y continuabamos hasta el breakpoint.

Al llegar al breakpoint y ver el código assembler, gdb muestra:

```

B+> 0x4019f0 <gdbme>      sub     rsp,0x88
0x4019f7 <gdbme+7>        call    0x400cc0 <getpid@plt>
0x4019fc <gdbme+12>       cmp     eax,0x12345678
0x401a01 <gdbme+17>       mov     edx,0x80
0x401a06 <gdbme+22>       je      0x401a30 <gdbme+64>
0x401a08 <gdbme+24>       mov     rsi,QWORD PTR [rip+0x2018e9]    # 0x6032f8
0x401a0f <gdbme+31>       mov     rdi,rsp
0x401a12 <gdbme+34>       call    0x401380
0x401a17 <gdbme+39>       mov     rdi,rax
0x401a1a <gdbme+42>       call    0x400c90 <puts@plt>
0x401a1f <gdbme+47>       add     rsp,0x88
0x401a26 <gdbme+54>       ret
0x401a27 <gdbme+55>       nop     WORD PTR [rax+rax*1+0x0]
0x401a30 <gdbme+64>       mov     rsi,QWORD PTR [rip+0x201a31]    # 0x603468
0x401a37 <gdbme+71>       mov     rdi,rsp
0x401a3a <gdbme+74>       call    0x401380
0x401a3f <gdbme+79>       mov     rdi,rax
0x401a42 <gdbme+82>       call    0x400c90 <puts@plt>
0x401a47 <gdbme+87>       add     rsp,0x88

native process 3585 In: gdbme                                L??  PC: 0x4019f0
(gdb) c
Continuing.
[Detaching after fork from child process 3648]

Breakpoint 1, 0x0000000004019f0 in gdbme ()
(gdb) _

```

En el código assembler se puede ver cómo compara el pid con el valor 0x12345678 y si son iguales, salta a otra dirección. Si continuamos sin que haga ese salto, el desafío se va a



reiniciar, por lo que probamos antes de realizar el cmp cambiar el registro de rax utilizando set \$rax = 0x12345678 y esto nos realizó el salto. Continuando por ese código luego nos imprime en la consola la respuesta: gdb\_rules.

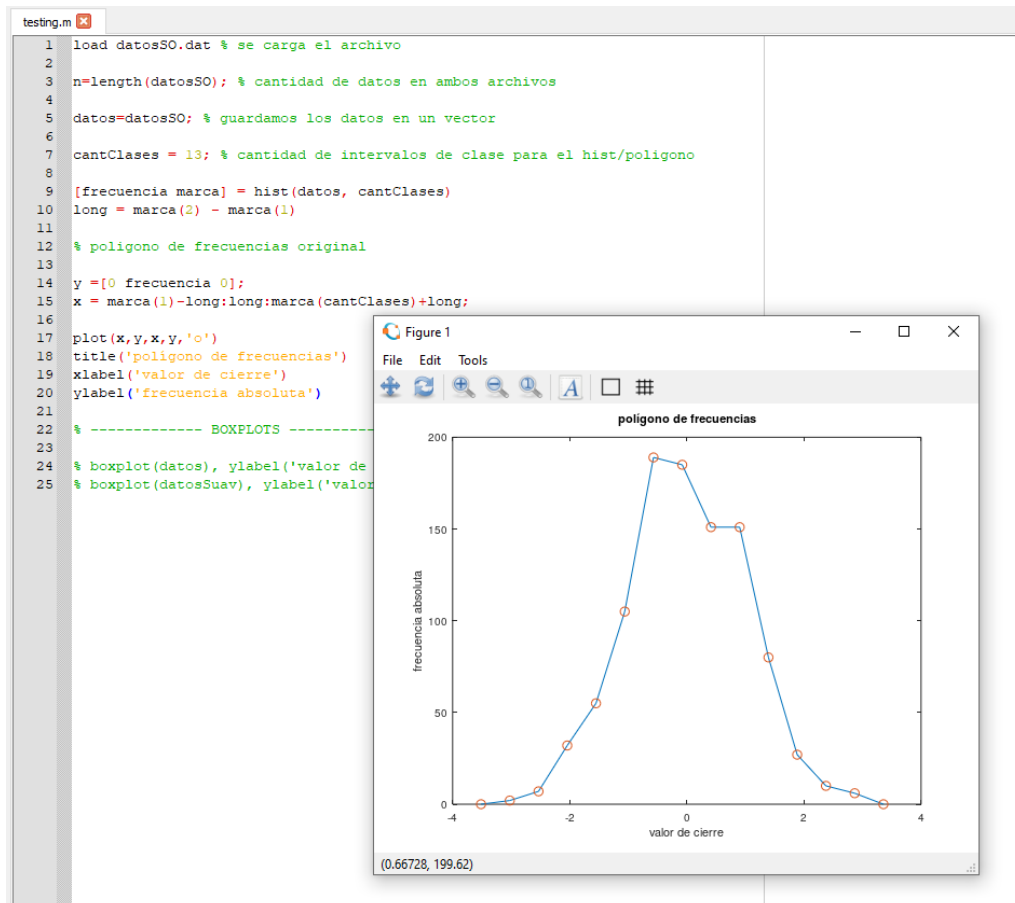
```
----- DESAFIO -----
b gdbme y encontrará el valor mágico

La respuesta es gdb_rules

----- PREGUNTA PARA INVESTIGAR -----
¿Qué es un RFC?
```

Desafío 12

```
----- DESAFIO -----
# Me conoces
1.242395 -0.122449 0.062892 -0.807854 -2.536570 -1.011193 -0.191870 -0.293738 -0.664849 -0.607788 0.513607 -1.433873 0.044753 -0.608025 -0.384248 0.970097 -0.934329 -0.065811 0.504251 0.682928 -0.045563 -1.16712
8 -0.838282 1.607298 -2.321728 -0.598451 1.282446 -1.331168 1.692139 0.645302 -1.213307 0.088720 -2.567064 1.419290 0.325051 0.476726 1.516506 -0.419626 -0.742378 -2.048674 -1.130853 0.444956 -0.023340
10.138086 -0.020330 -0.059380 0.043040 -0.301844 0.272136 1.711268 0.292832 -2.573883 -0.711058 -1.892780 -0.226787 -0.221518 -0.617796 0.486175 -0.476575 0.994756 0.356226 -0.335735 2.156734 0.127269 0.830209 -1
1.888871 1.946694 2.057098 0.259961 0.274710 -0.584044 -0.379890 0.339270 2.153765 -0.783378 -0.103429 -0.526680 -1.381999 -0.457222 -0.218543 0.211819 -0.456907 0.521881 -1.256327 0.883678 -1.045398 -0.645693 0
370960 1.107339 -0.120266 0.776290 0.432871 -0.285452 0.126310 -2.572547 -0.890409 -1.465928 0.327084 -1.020503 -0.986681 1.423109 -1.229492 0.851308 1.052854 -0.602573 0.338671 1.154953 1.752304 0.099083 1.3844
63 -1.322761 -1.612419 1.208603 -0.844607 0.593962 0.029543 2.069169 -1.206602 -1.756329 -0.207922 -0.704679 2.015994 0.437842 -0.547046 0.331622 0.644486 -0.699279 -0.831113 0.245569 -0.159257 1.151929 -1.62545
-0.009945 1.334322 0.385684 -1.192218 -1.178632 1.097611 -0.216560 -0.363707 1.674506 0.184444 -0.947293 -1.318616 -0.562105 0.608051 1.595397 0.104870 -1.731343 1.205858 0.907152 -0.444337 0.488514 0.274041 -
1.699755 -0.134819 -0.097534 1.858987 -1.624089 1.041265 -0.180321 0.161744 -1.155349 -1.345761 0.167573 0.665164 0.247153 0.461176 -0.586297 -0.364347 1.464495 -2.200924 0.065254 0.850819 0.705076 -0.979421 1.5
51637 -0.408540 -0.066245 0.304244 -2.390238 -0.362018 -0.094437 0.528243 0.306435 0.969152 -0.709492 1.075602 0.361003 -0.722250 -0.049294 0.612202 0.290919 -0.572250 0.578023 -0.544122 -0.004146 -1.308832 1.12
9195 -1.030634 1.656583 -0.569833 0.181954 -0.509362 -0.542573 -1.273655 -0.040645 0.389419 -0.297228 -0.628425 -0.892734 -0.131529 0.103181 -1.356021 0.206261 -0.376836 0.127500 -0.470157 2.947534 -0.504386 -
162865 -1.042964 1.252051 -0.556812 -0.943294 0.365314 0.676367 0.734129 0.587844 -0.222245 -0.768671 -0.869010 -0.385590 -1.843873 -1.271350 1.056371 -1.822971 -0.534846 -0.945782 2.322124 1.283559 -0.289842 -
0.842710 -0.222980 1.337007 1.517510 -0.787827 1.557239 -0.166846 -0.548364 0.638911 1.129492 -0.674426 0.103524 -0.509562 2.017438 1.117939 0.808537 1.107126 0.119615 1.431088 -0.952577 0.567750 -0.919323 -0.303
158 0.634227 -0.384835 -0.179163 -1.056947 -0.385225 1.622147 -1.364421 0.155507 -0.927332 -0.665152 1.220873 0.820804 0.830809 0.796333 0.859860 1.114440 -0.639374 1.947261 0.507587 0.605776 -0.118573
0.118078 -0.496178 -0.902888 -0.616547 0.623912 -0.694585 2.367187 -0.519333 0.441934 -0.580825 0.665607 1.064683 -0.636604 0.348005 -0.294429 -0.900725 -0.207252 -0.728027 1.711332 -0.763045 0.080778
-0.724385 -1.977842 -0.960535 1.583604 -1.203176 -0.050588 0.835986 -0.524829 -0.353803 2.444680 -0.051626 -0.751600 1.506095 1.316804 1.143928 0.660722 0.855806 0.767153 -0.374726 0.191338 1.761920 0.550302 -
1.99412 -0.312769 1.177108 -0.420460 0.770308 1.771456 -2.683387 -0.468475 -0.729752 0.991551 0.553414 0.494638 0.323428 0.376586 0.286401 -1.040981 0.313134 0.070193 -1.443353 -0.033815 -0.238140 -0.868556 -1.4
86959 0.107680 -1.498356 -0.238254 -0.094021 -0.585842 -0.968010 -0.697052 0.789390 -1.935080 1.924867 0.319772 1.254372 0.185680 -0.500060 1.456102 -0.571980 -0.351183 -0.004382 -1.174697 -0.317394 1.401751 1.6
90956 1.212766 -2.964979 0.635674 0.924466 1.483139 -0.649752 0.508383 0.215807 0.213393 0.757415 -0.703804 -0.894637 2.451564 -0.753241 -1.066692 1.011040 -0.730171 1.200323 -0.106803 0.602149 -0.402450 -1.783
628 1.381260 0.131213 0.834524 -0.452756 -0.470394 -1.504182 -0.649680 0.437686 0.278255 -0.534690 -0.360671 0.519086 1.274817 0.172193 0.017861 -0.470894 -1.581238 -0.439316 0.978639 1.445871 -0.853266 -0.72956
-0.724385 -1.977842 -0.960535 1.583604 -1.203176 -0.050588 0.835986 -0.524829 -0.353803 2.444680 -0.051626 -0.751600 1.506095 1.316804 1.143928 0.660722 0.855806 0.767153 -0.374726 0.191338 1.761920 0.550302 -
1.99412 -0.312769 1.177108 -0.420460 0.770308 1.771456 -2.683387 -0.468475 -0.729752 0.991551 0.553414 0.494638 0.323428 0.376586 0.286401 -1.040981 0.313134 0.070193 -1.443353 -0.033815 -0.238140 -0.868556 -1.4
86959 0.107680 -1.498356 -0.238254 -0.094021 -0.585842 -0.968010 -0.697052 0.789390 -1.935080 1.924867 0.319772 1.254372 0.185680 -0.500060 1.456102 -0.571980 -0.351183 -0.004382 -1.174697 -0.317394 1.401751 1.6
90956 1.212766 -2.964979 0.635674 0.924466 1.483139 -0.649752 0.508383 0.215807 0.213393 0.757415 -0.703804 -0.894637 2.451564 -0.753241 -1.066692 1.011040 -0.730171 1.200323 -0.106803 0.602149 -0.402450 -1.783
628 1.381260 0.131213 0.834524 -0.452756 -0.470394 -1.504182 -0.649680 0.437686 0.278255 -0.534690 -0.360671 0.519086 1.274817 0.172193 0.017861 -0.470894 -1.581238 -0.439316 0.978639 1.445871 -0.853266 -0.72956
-0.724385 -1.977842 -0.960535 1.583604 -1.203176 -0.050588 0.835986 -0.524829 -0.353803 2.444680 -0.051626 -0.751600 1.506095 1.316804 1.143928 0.660722 0.855806 0.767153 -0.374726 0.191338 1.761920 0.550302 -
1.99412 -0.312769 1.177108 -0.420460 0.770308 1.771456 -2.683387 -0.468475 -0.729752 0.991551 0.553414 0.494638 0.323428 0.376586 0.286401 -1.040981 0.313134 0.070193 -1.443353 -0.033815 -0.238140 -0.868556 -1.4
86959 0.107680 -1.498356 -0.238254 -0.094021 -0.585842 -0.968010 -0.697052 0.789390 -1.935080 1.924867 0.319772 1.254372 0.185680 -0.500060 1.456102 -0.571980 -0.351183 -0.004382 -1.174697 -0.317394 1.401751 1.6
90956 1.212766 -2.964979 0.635674 0.924466 1.483139 -0.649752 0.508383 0.215807 0.213393 0.757415 -0.703804 -0.894637 2.451564 -0.753241 -1.066692 1.011040 -0.730171 1.200323 -0.106803 0.602149 -0.402450 -1.783
628 1.381260 0.131213 0.834524 -0.452756 -0.470394 -1.504182 -0.649680 0.437686 0.278255 -0.534690 -0.360671 0.519086 1.274817 0.172193 0.017861 -0.470894 -1.581238 -0.439316 0.978639 1.445871 -0.853266 -0.72956
-0.724385 -1.977842 -0.960535 1.583604 -1.203176 -0.050588 0.835986 -0.524829 -0.353803 2.444680 -0.051626 -0.751600 1.506095 1.316804 1.143928 0.660722 0.855806 0.767153 -0.374726 0.191338 1.761920 0.550302 -
1.99412 -0.312769 1.177108 -0.420460 0.770308 1.771456 -2.683387 -0.468475 -0.729752 0.991551 0.553414 0.494638 0.323428 0.376586 0.286401 -1.040981 0.313134 0.070193 -1.443353 -0.033815 -0.238140 -0.868556 -1.4
86959 0.107680 -1.498356 -0.238254 -0.094021 -0.585842 -0.968010 -0.697052 0.789390 -1.935080 1.924867 0.319772 1.254372 0.185680 -0.500060 1.456102 -0.571980 -0.351183 -0.004382 -1.174697 -0.317394 1.401751 1.6
90956 1.212766 -2.964979 0.635674 0.924466 1.483139 -0.649752 0.508383 0.215807 0.213393 0.757415 -0.703804 -0.894637 2.451564 -0.753241 -1.066692 1.011040 -0.730171 1.200323 -0.106803 0.602149 -0.402450 -1.783
628 1.381260 0.131213 0.834524 -0.452756 -0.470394 -1.504182 -0.649680 0.437686 0.278255 -0.534690 -0.360671 0.519086 1.274817 0.172193 0.017861 -0.470894 -1.581238 -0.439316 0.978639 1.445871 -0.853266 -0.72956
-0.724385 -1.977842 -0.960535 1.583604 -1.203176 -0.050588 0.835986 -0.524829 -0.353803 2.444680 -0.051626 -0.751600 1.506095 1.316804 1.143928 0.660722 0.855806 0.767153 -0.374726 0.191338 1.761920 0.550302 -
1.99412 -0.312769 1.177108 -0.420460 0.770308 1.771456 -2.683387 -0.468475 -0.729752 0.991551 0.553414 0.494638 0.323428 0.376586 0.286401 -1.040981 0.313134 0.070193 -1.443353 -0.033815 -0.238140 -0.868556 -1.4
86959 0.107680 -1.498356 -0.238254 -0.094021 -0.585842 -0.968010 -0.697052 0.789390 -1.935080 1.924867 0.319772 1.254372 0.185680 -0.500060 1.456102 -0.571980 -0.351183 -0.004382 -1.174697 -0.317394 1.401751 1.6
90956 1.212766 -2.964979 0.635674 0.924466 1.483139 -0.649752 0.508383 0.215807 0.213393 0.757415 -0.703804 -0.894637 2.451564 -0.753241 -1.066692 1.011040 -0.730171 1.200323 -0.106803 0.602149 -0.402450 -1.783
628 1.381260 0.131213 0.834524 -0.452756 -0.470394 -1.504182 -0.649680 0.437686 0.278255 -0.534690 -0.360671 0.519086 1.274817 0.172193 0.017861 -0.470894 -1.581238 -0.439316 0.978639 1.445871 -0.853266 -0.72956
-0.724385 -1.977842 -0.960535 1.583604 -1.203176 -0.050588 0.835986 -0.524829 -0.353803 2.444680 -0.051626 -0.751600 1.506095 1.316804 1.143928 0.660722 0.855806 0.767153 -0.374726 0.191338 1.761920 0.550302 -
1.99412 -0.312769 1.177108 -0.420460 0.770308 1.771456 -2.683387 -0.468475 -0.729752 0.991551 0.553414 0.494638 0.323428 0.376586 0.286401 -1.040981 0.313134 0.070193 -1.443353 -0.033815 -0.238140 -0.868556 -1.4
86959 0.107680 -1.498356 -0.238254 -0.094021 -0.585842 -0.968010 -0.697052 0.789390 -1.935080 1.924867 0.319772 1.254372 0.185680 -0.500060 1.456102 -0.571980 -0.351183 -0.004382 -1.174697 -0.317394 1.401751 1.6
90956 1.212766 -2.964979 0.635674 0.924466 1.483139 -0.649752 0.508383 0.215807 0.213393 0.757415 -0.703804 -0.894637 2.451564 -0.753241 -1.066692 1.011040 -0.730171 1.200323 -0.106803 0.602149 -0.402450 -1.783
628 1.381260 0.131213 0.834524 -0.452756 -0.470394 -1.504182 -0.649680 0.437686 0.278255 -0.534690 -0.360671 0.519086 1.274817 0.172193 0.017861 -0.470894 -1.581238 -0.439316 0.978639 1.445871 -0.853266 -0.72956
-0.724385 -1.977842 -0.960535 1.583604 -1.203176 -0.050588 0.835986 -0.524829 -0.353803 2.444680 -0.051626 -0.751600 1.506095 1.316804 1.143928 0.660722 0.855806 0.767153 -0.374726 0.191338 1.761920 0.550302 -
1.99412 -0.312769 1.177108 -0.420460 0.770308 1.771456 -2.683387 -0.468475 -0.729752 0.991551 0.553414 0.494638 0.323428 0.376586 0.286401 -1.040981 0.313134 0.070193 -1.443353 -0.033815 -0.238140 -0.868556 -1.4
86959 0.107680 -1.498356 -0.238254 -0.094021 -0.585842 -0.968010 -0.697052 0.789390 -1.935080 1.924867 0.319772 1.254372 0.185680 -0.500060 1.456102 -0.571980 -0.351183 -0.004382 -1.174697 -0.317394 1.401751 1.6
90956 1.212766 -2.964979 0.635674 0.924466 1.483139 -0.649752 0.508383 0.215807 0.213393 0.757415 -0.703804 -0.894637 2.451564 -0.753241 -1.066692 1.011040 -0.730171 1.200323 -0.106803 0.602149 -0.402450 -1.783
628 1.381260 0.131213 0.834524 -0.452756 -0.470394 -1.504182 -0.649680 0.437686 0.278255 -0.534690 -0.360671 0.519086 1.274817 0.172193 0.017861 -0.470894 -1.581238 -0.439316 0.978639 1.445871 -0.853266 -0.72956
-0.724385 -1.977842 -0.960535 1.583604 -1.203176 -0.050588 0.835986 -0.524829 -0.353803 2.444680 -0.051626 -0.751600 1.506095 1.316804 1.143928 0.660722 0.855806 0.767153 -0.374726 0.191338 1.761920 0.550302 -
1.99412 -0.312769 1.177108 -0.420460 0.770308 1.771456 -2.683387 -0.468475 -0.729752 0.991551 0.553414 0.494638 0.323428 0.376586 0.286401 -1.040981 0.313134 0.070193 -1.443353 -0.033815 -0.238140 -0.868556 -1.4
86959 0.107680 -1.498356 -0.238254 -0.094021 -0.585842 -0.968010 -0.697052 0.789390 -1.935080 1.924867 0.319772 1.254372 0.185680 -0.500060 1.456102 -0.571980 -0.351183 -0.004382 -1.174697 -0.317394 1.401751 1.6
90956 1.212766 -2.964979 0.635674 0.924466 1.483139 -0.649752 0.508383 0.215807 0.213393 0.757415 -0.703804 -0.894637 2.451564 -0.753241 -1.066692 1.011040 -0.730171 1.200323 -0.106803 0.602149 -0.402450 -1.783
628 1.381260 0.131213 0.834524 -0.452756 -0.470394 -1.504182 -0.649680 0.437686 0.278255 -0.534690 -0.360671 0.519086 1.274817 0.172193 0.017861 -0.470894 -1.581238 -0.439316 0.978639 1.445871 -0.853266 -0.72956
-0.724385 -1.977842 -0.960535 1.583604 -1.203176 -0.050588 0.835986 -0.524829 -0.353803 2.444680 -0.051626 -0.751600 1.506095 1.316804 1.143928 0.660722 0.855806 0.767153 -0.374726 0.191338 1.761920 0.550302 -
1.99412 -0.312769 1.177108 -0.420460 0.770308 1.771456 -2.683387 -0.468475 -0.729752 0.991551 0.553414 0.494638 0.323428 0.376586 0.286401 -1.040981 0.313134 0.070193 -1.443353 -0.033815 -0.238140 -0.868556 -1.4
86959 0.107680 -1.498356 -0.238254 -0.094021 -0.585842 -0.968010 -0.697052 0.789390 -1.935080 1.924867 0.319772 1.254372 0.185680 -0.500060 1.456102 -0.571980 -0.351183 -0.004382 -1.174697 -0.317394 1.401751 1.6
90956 1.212766 -2.964979 0.635674 0.924466 1.483139 -0.649752 0.508383 0.215807 0.213393 0.757415 -0.703804 -0.894637 2.451564 -0.753241 -1.066692 1.011040 -0.730171 1.200323 -0.106803 0.602149 -0.402450 -1.783
628 1.381260 0.131213 0.834524 -0.452756 -0.470394 -1.504182 -0.649680 0.437686 0.278255 -0.534690 -0.360671 0.519086 1.274817 0.172193 0.017861 -0.470894 -1.581238 -0.439316 0.978639 1.445871 -0.853266 -0.72956
-0.724385 -1.977842 -0.960535 1.583604 -1.203176 -0.050588 0.835986 -0.524829 -0.353803 2.444680 -0.051626 -0.751600 1.506095 1.316804 1.143928 0.660722 0.855806 0.767153 -0.374726 0.191338 1.761920 0.550302 -
1.99412 -0.312769 1.177108 -0.420460 0.770308 1.771456 -2.683387 -0.468475 -0.729752 0.991551 0.553414 0.494638 0.323428 0.376586 0.286401 -1.040981 0.3
```



A grandes rasgos, nos pareció una campana de Gauss que, como vimos en la materia, es el gráfico de una distribución normal. Probando diversas combinaciones, llegamos a la respuesta correcta: “normal”.