



## Trabajo Práctico Nro.3

### 72.11 - Sistemas Operativos

### Sockets, Challenges e Investigación

2022 1C

Grupo 2

#### Integrantes:

- 60299 - Juan Pablo Arias
- 60570 - Andrés Podgorny
- 61278 - Sol Victoria Anselmo

# Índice

<b>Challenge 1</b>	<b>2</b>
<b>Challenge 2</b>	<b>2</b>
<b>Challenge 3</b>	<b>3</b>
<b>Challenge 4</b>	<b>3</b>
<b>Challenge 5</b>	<b>4</b>
<b>Challenge 6</b>	<b>4</b>
<b>Challenge 7</b>	<b>5</b>
<b>Challenge 8</b>	<b>5</b>
<b>Challenge 9</b>	<b>5</b>
<b>Challenge 10</b>	<b>6</b>
<b>Challenge 11</b>	<b>6</b>
<b>Challenge 12</b>	<b>8</b>
<b>Easter Eggs</b>	<b>9</b>

## Challenge 1

Para el primer challenge, nos conectamos con el servidor (el archivo binario `server` dado por la cátedra) con el comando `netcat`. Para esto, tal como fue mostrado en clase, corrimos el servidor con el comando `strace`: `strace ./server` para hallar el puerto y el IP address donde el server se encontraba. La respuesta a este challenge fue ‘entendido’.

```
----- 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".

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

Respuesta de investigación: Para resolver este primer challenge, lo que hicimos fue hacer un google search para averiguar cómo hacer la conexión entre el server y el cliente, descubriendo así el comando NETCAT, el cual requiere un puerto y una IP address para funcionar. Vimos en clase que al hacer el comando strace sobre el servidor podíamos obtener estos datos.

## Challenge 2

Para el segundo challenge buscamos el episodio asignado en la consigna encontrando el siguiente video en youtube: <https://www.youtube.com/watch?v=DQBIq45c1T4> el cual explica la decodificación que debíamos usar. Una vez descifrado el número, lo ingresamos como respuesta pero no nos dejó avanzar; luego descubrimos que era el número de teléfono del ITBA. La respuesta a este challenge fue 'itba'.

```
----- 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?

[]
```

Respuesta de investigación: En cuanto a las diferencias que podemos encontrar entre el TCP y el UDP, podemos destacar principalmente lo siguiente: El TCP está diseñado para cumplir fundamentalmente dos funciones. Una de ellas es transmitir información entre dos dispositivos, mientras que la otra función es verificar que se haya transmitido correctamente dicha información. El TCP es una gran función se la aplica junto al protocolo IP, ya que los datagramas del mismo no tienen la función de manejar las conexiones, entonces el TCP le aporta esto.

Por otro lado tenemos el UDP, que a pesar de ser similar al TCP podemos ver que su principal diferencia es que no es un protocolo de transporte orientado a conexión ya que no verifica que hayan sido transferida la información entre dos dispositivos. Sin embargo, también a diferencia del TCP cuenta con una gran ventaja en cuanto a la velocidad de transmisión de información, ya que es mucho más rápido. Es por esto que es muy útil utilizar el UDP en casos de transmisión de voz o vídeo.

## Challenge 3

Para el tercer challenge ingresamos el link de la consigna en el navegador y descargamos la imagen. Al ver que era una imagen simplemente en blanco, la abrimos con el Paint y la pintamos; descubriendo así la respuesta al desafío. La respuesta a este challenge fue 'M4GFKZ289aku'.

```
----- 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? ¿P
or qué?

[ ]
```

```
andre@LAPTOP-71SFWJ3T:/mnt/d/ITBA/2022/1erQ/50/TP3$ nc 0.0.0.0 8080
entendido
0810 222 4822
itba
[ ]
```

M4GFKZ289aku

Respuesta de investigación: Sí, ya que el cliente fue configurado de forma tal que el puerto de su servidor address 8080, entonces dicho servidor estará siempre conectado a ese IP address.

## Challenge 4

Para el cuarto challenge volvimos a correr el comando strace sobre el servidor para encontrar donde se encontraba el write con el file descriptor incorrecto y redireccionamos su output a un archivo de texto ('challenge4.txt'). La respuesta a este challenge fue 'fk3wfLCm3QvS'.

```
----- 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?

[ ]
```

```
andre@LAPTOP-71SFWJ3T:/mnt/d/ITBA/2022/1erQ/50/TP3$ nc 0.0.0.0 8080
entendido
0810 222 4822
itba
M4GFKZ289aku
[ ]
```

```

) = 10
write(13, ".....", 61) = -1 EBADF (Bad file descriptor)
dup(2) = 5
fcntl(5, F_GETFL) = 0x8002 (flags O_RDWR|O_LARGEFILE)
fstat(5, {st_mode=S_IFCHR|0620, st_rdev=makedev(136, 3), ...}) = 0
write(5, "write: Bad file descriptor\n", 27)write: Bad file descriptor
) = 27
close(5) = 0
write(1, "\n---- PREGUNTA PARA INVESTIGAR ....", 37
---- PREGUNTA PARA INVESTIGAR ----) = 37
write(1, "\n", 1
) = 1
write(1, "\302\277Qu\303\251 \303\272til abstracci\303\263n es uti"...
, 122)Qué útil abstracción es utilizada para comunicarse con sockets? ¿se puede utilizar read(2) y write(2) para operar?
) = 122
read(4,

```

Respuesta de investigación: Lo ideal en caso de no estar al tanto de si la información está siendo transmitida mediante un sistema Unix, usar send() y recv() es la mejor opción.

## Challenge 5

Para el quinto challenge corrimos el comando strings sobre el servidor para poder encontrar la respuesta al desafío en la línea 277 y redireccionamos su output a un archivo de texto ('challenge5.txt'). La respuesta a este challenge fue 'too\_easy'.

```

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

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

andre@LAPTOP-7ISFNU3T:/mnt/d/ITBA/2022/1erQ/SO/TP3$ nc 0.0.0.0 8080
entendido
itba
M4GFKZ289aku
fk3wflCm3QvS

```

Respuesta de investigación: El TCP se asegura de que la información sea transmitida correctamente de la siguiente manera, primero se ocupa de dividir los flujos de bytes en segmentos ordenados a los cuales se les asigna un número de secuencia respectivamente antes de que sean enviados a mediante el protocolo IP. El dispositivo que reciba la información será quién se ocupará de chequear el número de secuencia para que si llegara a pasar que falte algún segmento el TCP vuelve a pedir que le envíen la información hasta que la misma pueda llegar correctamente.

## Challenge 6

Para el sexto challenge hicimos un disassembly sobre el servidor para poder encontrar la respuesta al desafío y redireccionamos su output a un archivo de texto ('challenge6.txt'). La respuesta a este challenge fue '.RUN\_ME'.

```

----- 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?

andre@LAPTOP-7ISFNU3T:/mnt/d/ITBA/2022/1erQ/SO/TP3$ nc 0.0.0.0 8080
entendido
itba
M4GFKZ289aku
fk3wflCm3QvS
too_easy

```

Respuesta de investigación: Lo que sería más conveniente en cuanto a costos principalmente es usar hilos ya que estos podrían compartir los mismos espacios de memoria entonces se podría crear un hilo para cada cliente del servidor.

## Challenge 7

Para el séptimo challenge filtramos el error del servidor usando el filedescriptor de stderr (./server 2) para poder encontrar la respuesta al desafío y redireccionamos su output a un archivo de texto ('challenge7.txt'). La respuesta a este challenge fue 'K5n2UFfpFMUN'.

<pre> ----- DESAFIO ----- Filter error  DKw)PBwUK#EL:fmXa.'?AaF [A%I)r1le\aesps6S:@ns5x.)d"u7Vei=0"l-`s[Q]hH4[iK[]*m)T&amp;;s9T.K w3Zt7S=Rj .?OQaLY;x Me!6x"w)2s UE3KN5o5jo}Xn0e&amp;&lt;&lt;cSwN2&gt;U&amp;FpFMUNgd\$\$\$gx  ----- PREGUNTA PARA INVESTIGAR ----- ¿Cómo se puede implementar un servidor que atienda muchas conexiones sin usar proce sos ni threads?  </pre>	<pre> andre@LAPTOP-7ISFNU3T:/mnt/d/ITBA/2022/1erQ/50/TP3\$ nc 0.0.0.0 8080 entendido itba M4GFKZ289aku fk3wFLcm3QvS too_easy .RUN_ME </pre>
<pre> ----- DESAFIO ----- Filter error  La respuesta es K5n2UFfpFMUN  ----- PREGUNTA PARA INVESTIGAR ----- ¿Cómo se puede implementar un servidor que atienda muchas conexiones sin usar proce sos ni threads?  </pre>	<pre> andre@LAPTOP-7ISFNU3T:/mnt/d/ITBA/2022/1erQ/50/TP3\$ nc 0.0.0.0 8080 entendido itba M4GFKZ289aku fk3wFLcm3QvS too_easy .RUN_ME </pre>

Respuesta de investigación: Esto podría llevarse a cabo mediante la utilización del comando select, que al recibir varias peticiones resuelve atendiéndolas de forma secuencial.

## Challenge 8

Para el octavo challenge nos aparecio la respuesta ya dada en la terminal con un fondo negro (la terminal donde corrimos el archivo server). La respuesta a este challenge fue 'BUmyYq5XxXGt'.

<pre> ----- DESAFIO ----- ¿?  La respuesta es BUmyYq5XxXGt  ----- PREGUNTA PARA INVESTIGAR ----- ¿Qué aplicaciones se pueden utilizar para ver el tráfico por la red?  </pre>	<pre> andre@LAPTOP-7ISFNU3T:/mnt/d/ITBA/2022/1erQ/50/TP3\$ nc 0.0.0.0 8080 entendido itba M4GFKZ289aku fk3wFLcm3QvS too_easy .RUN_ME K5n2UFfpFMUN </pre>
---	--

Respuesta de investigación: Para el tráfico de red se pueden usar principalmente las siguientes aplicaciones: Ping Viewer, Ping GUI, NMAP y eToolz.

## Challenge 9

Para el noveno challenge hicimos un copy-paste de la fórmula que estaba en formato latex en un solver matemático (integrando obtenemos la respuesta). La respuesta a este challenge fue 'u^v'.

<pre> ----- 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?  </pre>	<pre> andre@LAPTOP-7ISFNU3T:/mnt/d/ITBA/2022/1erQ/50/TP3\$ nc 0.0.0.0 8080 entendido itba M4GFKZ289aku fk3wFLcm3QvS too_easy .RUN_ME K5n2UFfpFMUN BUmyYq5XxXGt </pre>
--	---

Respuesta de investigación: En caso de que se realice todo el trabajo dentro de un host, lo ideal en cuanto a eficiencia sería utilizar pipes, esto se debe a que los mismos son mucho más rápidos que los sockets. Por otro lado, si se tratara de trabajar mediante más de un host, lo más eficiente sería utilizar sockets ya que los pipes solo pueden trabajar dentro de un mismo host.

## Challenge 10

Para el décimo challenge buscamos 'quine' en google y probamos con las opciones del siguiente link: [https://es.wikipedia.org/wiki/Quine\\_\(programa\)](https://es.wikipedia.org/wiki/Quine_(programa)) para crear un archivo .c que nos dé la respuesta del challenge. La respuesta a este challenge fue 'chin\_chu\_lan\_cha'.

<pre>----- 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?  []</pre>	<pre>andre@LAPTOP-7ISFNU3T:/mnt/d/ITBA/2022/1erQ/SO/TP3\$ nc 0.0.0.0 8080 entendido itba M4GFKZ289aku fk3wflCm3QvS too_easy .RUN_ME K5n2UfFpFMUN BUmyYq5XxGt u^v</pre>
<pre>----- 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?  []</pre>	<pre>andre@LAPTOP-7ISFNU3T:/mnt/d/ITBA/2022/1erQ/SO/TP3\$ nc 0.0.0.0 8080 entendido itba M4GFKZ289aku fk3wflCm3QvS too_easy .RUN_ME K5n2UfFpFMUN BUmyYq5XxGt u^v</pre>

Respuesta de investigación: Este protocolo presenta las siguientes características: Transmite información secuenciada de usuario dentro de múltiples corrientes de datos y posee la cualidad de ordenar dicha información. Puede confirmar la transmisión de los datos de usuario evitando fallos y duplicados. Es capaz de fragmentar datos para lograr adaptarse al tamaño máximo de paquete de la ruta de red. Puede agrupar los mensajes de usuario de forma de dejarlos como un solo paquete. Por su host que admite más de una dirección de red, logra ser tolerante a fallos.

## Challenge 11

Para el onceavo challenge debugamos el servidor con el gdb haciendo un breakpoint en 'gdbme'. La respuesta a este challenge fue 'gdb\_rules'.

<pre>----- DESAFIO ----- b gdbme y encontrá el valor mágico  ENTER para reintentar.  ----- PREGUNTA PARA INVESTIGAR ----- ¿Qué es un RFC?  []</pre>	<pre>andre@LAPTOP-7ISFNU3T:/mnt/d/ITBA/2022/1erQ/SO/TP3\$ nc 0.0.0.0 8080 entendido itba M4GFKZ289aku fk3wflCm3QvS too_easy .RUN_ME K5n2UfFpFMUN BUmyYq5XxGt u^v chin_chu_lan_cha</pre>
---	---

```

Registers
rax 0xfffffffffffffe00    rbx 0x0000000000000003    rcx 0x00007f22ba9eb2c0
rdx 0x00007ffdd317780c    rsi 0x00007ffdd3177820    rdi 0x0000000000000003
rbp 0x0000000000001f90    rsp 0x00007ffdd31777e8    r8 0x0000000000000004
r9 0x00007f22bafbd010     r10 0x00007ffdd31775b0    r11 0x0000000000000246
r12 0x00000000000401096   r13 0x00007ffdd3177d50    r14 0x0000000000000000
r15 0x0000000000000000    rip 0x000000000004019f0    eflags [ PF ZF IF ]
cs 0x00000033             ss 0x0000002b             es 0x00000000
fs 0x00000000             gs 0x00000000             fs_base void
gs_base void              k_gs_base void          cr0 void
cr2 void                  cr3 void              cr4 void
cr8 void                  efer void

Assembly
!0x000000000004019f0 ? sub $0x88,%rsp
0x000000000004019f7 ? callq 0x400cc0 <getpid@plt>
0x000000000004019fc ? cmp $0x12345678,%eax
0x00000000000401a01 ? mov $0x80,%edx
0x00000000000401a06 ? je 0x401a30 <gdbme+64>
0x00000000000401a08 ? mov 0x2018e9(%rip),%rsi # 0x6032f8
0x00000000000401a0f ? mov %rsp,%rdi
0x00000000000401a12 ? callq 0x401380
0x00000000000401a17 ? mov %rax,%rdi
0x00000000000401a1a ? callq 0x400c90 <puts@plt>

Memory
Stack
[0] from 0x000000000004019f0 in gdbme
[1] from 0x0000000000040124b
[2] from 0x00000000000400f02
[3] from 0x00007f22ba923b45 in __libc_start_main+245 at libc-start.c:287
[4] from 0x000000000004010bf
>>>

```

```

Output/messages
_IO_puts (str=0x7fffe16801e0 "La respuesta es gdb_rules") at ioputs.c:34
34 ioputs.c: No such file or directory.

Assembly
~
~
~
~
~
0x00007f7dbff87990 _IO_puts+0 push %r12
0x00007f7dbff87992 _IO_puts+2 mov %rdi,%r12
0x00007f7dbff87995 _IO_puts+5 push %rbp
0x00007f7dbff87996 _IO_puts+6 push %rbx
0x00007f7dbff87997 _IO_puts+7 callq 0x7f7dbff9dc10 <strlen>

Breakpoints
[1] break at 0x00007f7dbff51330 in ../sysdeps/unix/syscall-template.S:81 for kill hit 4 times
[2] break at 0x000000000004019f0 for gdbme hit 1 time

Expressions
History
Memory
Registers
rax 0x00007fffe16801e0    rbx 0x000000000000000a    rcx 0x0000000000000021
rdx 0x0000000000000019    rsi 0x00000000000003470    rdi 0x00007fffe16801e0
rbp 0x000000000000000a    rsp 0x00007fffe16801d8    r8 0x0000000000000019
r9 0x00007f7dbff6799a     r10 0x00007f7dc02bfbe0     r11 0x0000000000000246
r12 0x0000000001a72010    r13 0x00007fffe1680770    r14 0x0000000000000000
r15 0x0000000000000000    rip 0x00007f7dbff87990    eflags [ PF ZF IF ]
cs 0x00000033             ss 0x0000002b             ds 0x00000000
es 0x00000000             fs 0x00000000             gs 0x00000000

Source
Cannot display "ioputs.c"
Stack
[0] from 0x00007f7dbff87990 in _IO_puts+0 at ioputs.c:34
[1] from 0x00000000000401a47 in gdbme
[2] from 0x00000000000400f7c
[3] from 0x00007f7dbff3db45 in __libc_start_main+245 at libc-start.c:287
[4] from 0x000000000004010bf

Threads
[1] id 27 name server from 0x00007f7dbff87990 in _IO_puts+0 at ioputs.c:34

Variables
arg str = 0x7fffe16801e0 "La respuesta es gdb_rules": 76 'L'
loc result = -1, len = <optimized out>

>>>

```

Respuesta de investigación: Un RFC es un documento que puede ser escrito por cualquier persona y que contiene una propuesta para una nueva tecnología y/o mejoras de tecnologías ya existentes. En otras palabras, las RFC conforman la documentación de protocolos y tecnologías de Internet, siendo incluso muchas de ellas estándares.



## Challenge 12

Para el doceavo challenge exportamos los datos a un archivo de texto para verlos mejor y luego los importamos en un excel (un dato por fila). Graficamos los datos y dado el comportamiento y el comentario de “Me conoces”, probamos las aproximaciones que ya conocíamos ofrecidas por el excel para ver cual se acoplaba mejor al gráfico (esta fue la campana de gauss/distribución normal). La respuesta a este challenge fue ‘normal’.

```

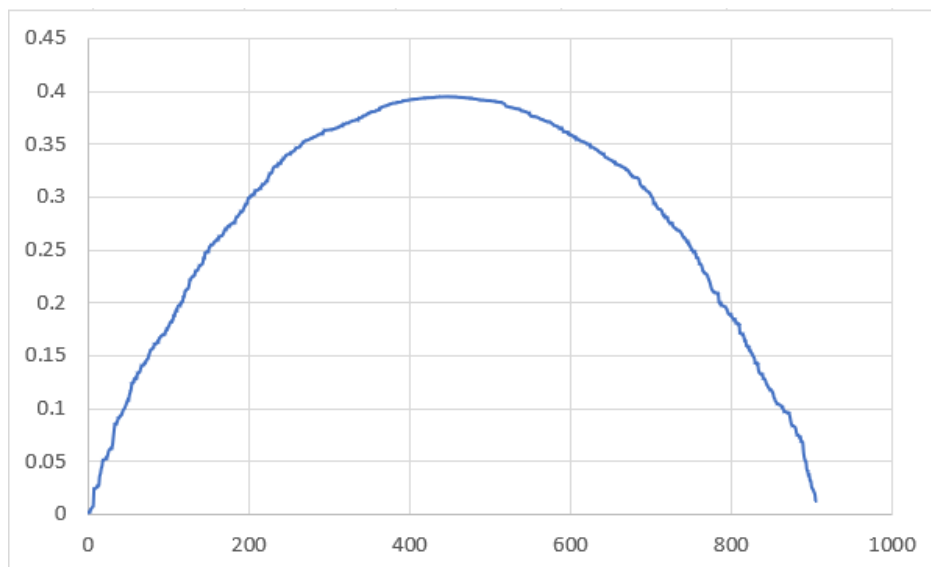
----- DESAFIO -----
Me conoces

-0.591670 -1.351000 1.181277 2.500761 1.340832 -1.463068 -0.688152 0.085827 -0.9453
93 -0.134135 -1.068381 -0.485562 0.516723 -0.402102 -1.352543 1.370974 -2.971780 0.
944259 -0.431376 0.481990 -0.408717 -1.377486 0.232603 0.617453 0.996094 0.132185 -
1.288242 -0.711511 1.306466 -0.867995 -0.852301 -0.706436 -0.323953 0.315617 -0.303
837 0.045449 0.322714 -0.554920 0.025953 -0.779202 2.205013 -0.908333 -0.439051 2.2
62459 -0.513951 -0.337605 -0.227956 2.053091 -0.920238 -0.434204 -0.364617 -0.27954
4 -0.461220 -0.096379 0.332544 0.670097 1.787359 0.894673 -0.622536 -0.230545 -0.60
6111 -1.305827 -2.339737 -0.435327 -0.471934 0.649058 -0.214330 0.239287 1.384717 0
.403343 0.152552 -0.400000 -1.340719 -1.486300 0.777428 -0.944606 -0.640728 -0.1181
07 -2.355075 1.227686 -1.313699 0.738653 -0.216995 -1.263557 -0.040104 -0.212188 0.

1.516626 -0.951547 -0.969167 -0.136804 1.612103 1.620306 0.479725 0.851516 -0.70781
6 2.061360 -0.693132 -0.209437 -0.362841 1.262635 -0.779517 -0.043669 1.361775 1.70
2587 1.195560 -0.453500 -2.358074 -0.551955 0.103440 1.202615 -0.395582 -1.085980 2.
208295 1.572387 -0.739792 0.318671 -0.072055 -0.578998 1.143309 1.058223 -0.439820
0.917655 -2.043265 -0.830963 1.012895 -0.934126 -0.404993 -0.757323 -0.826484 0.24
1314 -1.253907 1.651149 1.706184 -1.025877 0.176034 -1.026649 0.926891 -0.732763 1.
443121 0.070564 -1.445366 -0.534078 -1.509187 1.394076 0.284700 1.696844 -0.513398
-0.061710 -1.510707 1.892750 -1.258555 -0.084112 0.822952 -2.099712 1.368616 -0.852
959 -0.407909 2.397881 -1.685235 1.902475 -0.154670 0.117340 -0.155347 0.092495 0.3
58497 0.043587 -0.885135 0.131687 0.921567 0.602474 -0.364121 1.572311 0.318056 -0.
289507 0.952665 0.614171 0.370994 1.226440 -1.530892 0.657974 -1.070245 1.751582 0.
140213 -0.406571 -0.411806 -1.186825 -3.058459 -1.176317 -0.101898 -0.274889 -1.418
476 -1.944658 0.791457 0.503060 0.027044 0.779697 -1.208188 0.715093 -0.066480 -1.9
67874 0.912998 0.271368 -1.672303 0.147520 0.358182 0.142519 -0.775396 -0.858409 0.
837846 1.693827 0.514198 1.405633 0.423096 1.650169 0.477720 0.859702 1.065894 -0.7
52267 -0.809538 -1.323445 0.124950 -0.108429 -0.929198 0.965824 0.353635 -3.657613
0.610612 0.859418 -0.089489 -1.709313 0.193867 0.353103 0.274524 -0.636591 0.641423
0.316979 -0.486176 0.360181 1.695192 1.084828 0.222218 -0.836575 1.441104 0.394015
0.391837 -0.358597 0.890482 -0.343651 -1.125477 0.061493 -0.378951 -0.092428 -0.60
8786 0.489472 1.049782 -0.665079 0.658174 -2.023572 0.470518 0.873427 -0.027145 -0.
288989 -0.996898 -0.132767 1.091299 1.850836 1.198227 -0.163551 -0.644186 1.550309
0.834730 0.862353 -0.926124 -1.422117 1.249107 0.879044 -0.475338 -1.326855 -0.1758
19 -0.074287 0.805599 1.251540 0.250244 -1.487520 -0.613375 -1.215273 1.193103 -0.3
26710 0.062066 1.182082 -0.004571 1.657484 -1.436651 1.125518 -0.059620 -1.569551 -
1.176232 1.130522 -0.562197 -0.909619 2.463722 -2.038161 -3.272364 -0.693172 -1.516
634 0.608599 1.697897 0.106166 -0.643266 -0.530735 -0.346173 1.263336 -1.940812 1.5
44174 -0.069551 0.484433 -0.454699 -0.073029 1.823525 -1.230293 0.114755 -0.973414
-1.436314 0.585872 1.635062 0.624976 0.628061 -0.338800 -1.370777 -0.922116 0.65065
1 0.827019 -0.499087 0.839170 1.022026 0.166019 1.313811 -0.831556 0.269677 -0.4105
20 -0.895042 0.469732 1.705045 -0.870327 0.075432 0.324212 0.469682 -0.083868 -1.63
4213 -1.372676 -0.187009 1.269333 -0.117587 0.571715 0.059540 0.052783 -0.717782 0.
249476 -0.277475 0.245191 -0.350057 -0.044107 -0.110979 -0.956787 1.531478 -0.01204
9 -0.903314 -0.201593 0.033855 0.943450 -0.243478 1.400463 -0.411754 -0.537283 0.95
7149 0.513336 0.387036 1.056089 0.888748 -0.087058 -3.347779 -0.804479 0.183698 0.3
10222 -0.880073 1.243621 -0.808660 -0.383105 -0.371600 0.523720 0.469980 -1.516599
0.081214 -0.748677 0.318449 1.498366 -1.279370 -0.268170 -0.681142 0.779319 -0.4720
45 -2.394954 1.669088 -0.006754 0.513816 -1.117913 0.268147 1.889279 -1.074362 -1.3
48440 -0.873692 -0.115132 0.065119 -1.131308 -1.598367 0.600826 1.590972 -0.176950
-0.550166 -1.123122 1.063906 -0.794156 0.377486 -0.363288 0.635216 0.675820 1.67062
6 0.234076 0.450927 0.525231 -0.469470 -0.145025 -0.808723 -0.741980 0.402442 -0.35
1515 1.845330 1.980758 -0.860577

----- PREGUNTA PARA INVESTIGAR -----
¿Fue divertido?

```



Respuesta de investigación: Sí, fue divertido. Fue muy entretenido resolver cada challenge e interactuar con el servidor.

```
Felicitaciones, finalizaron el juego. Ahora deberán implementar el servidor que se
comporte como el servidor provisto

andre@LAPTOP-7ISFNU3T:/mnt/d/ITBA/2022/1erQ/50/TP3$ nc 0.0.0.0 8080
entendido
itba
M4GFKZ289aku
fk3wflCm3QyS
too_easy
.RUN_ME
K5n2UfFpFMUN
BUmyYq5XxxGt
u^v
chín_chu_lan_cha
gdb_rules
normal
[]
```

## Easter Eggs

```
< ESTO ES UN EASTER_EGG >
-----
\      ^  ^
 \    (oo)\_____
  (__) \         )\ /\
       ||----w |
       ||     ||
```