

C Piscine
Shell 00

Resumen: Este documento corresponde al enunciado del módulo Shell 00 de la C Piscine de 42.

Versión: 3

Índice general

Τ.	Histi acciones	
II.	Introducción	4
III.	Ejercicio 00 : Z	6
IV.	Ejercicio 01 : testShell00	7
V.	Ejercicio 02 : Sí, quiero más	8
VI.	Ejercicio 03 : ¡Conéctame por SSH!	9
VII.	Ejercicio 04 : midLS	10
VIII.	Exercise 05 : GiT commit	11
IX.	Exercise 06 : gitignore	12
Χ.	Ejercicio 07 : diff	13
XI.	Ejercicio 08 : clean	14
XII.	Eiercicio 09 : Illusions, not tricks, Michael	15

Capítulo I

Instrucciones

- Esta página será la única referencia: no te fíes de los rumores.
- ¡Ten cuidado! Los enunciados pueden cambiar en cualquier momento.
- Asegúrate de que tus directorios y archivos tienen los permisos adecuados.
- Debes respetar el procedimiento de entrega para todos tus ejercicios.
- Tus compañeros de piscina se encargarán de corregir tus ejercicios.
- Además de por tus compañeros, también serán corregidos por un programa que se llama la Moulinette.
- La Moulinette es muy estricta a la hora de evaluar. Está completamente automatizada. Es imposible discutir con ella sobre tu nota. Por lo tanto, se extremadamente riguroso para evitar cualquier sorpresa.
- La Moulinette no tiene una mente muy abierta. No intenta comprender el código que no respeta la Norma. La Moulinette utiliza el programa norminette para comprobar La Norma en sus archivos. Entiende entonces que es estúpido entregar un código que no pase la norminette.
- Los ejercicios han sido ordenados con mucha precisión, del más sencillo al más complejo. En ningún caso se tendrá en cuenta un ejercicio complejo si no se ha conseguido realizar perfectamente un ejercicio más sencillo.
- El uso de una función prohibida se considera una trampa. Cualquier trampa será sancionada con la nota -42.
- Los ejercicios shell se deben ejecutar con /bin/sh.
- No puedes dejar en tu directorio <u>ningún</u> archivo que no se haya indicado de forma explícita en los enunciados de los ejercicios.
- ¿Tienes alguna pregunta? Pregunta a tu compañero de la derecha. Si no, prueba con tu compañero de la izquierda.
- Tu manual de referencia se llama Google / man / Internet / ...
- ¡No olvides participar en el slack de tu Piscina!

C Piscine		Shell 00
• Lee detenidamente los ejemplos riamente en los enunciados	s. Podrían exigir cosas que no se e	
• Razona. ¡Te lo suplico, por Tho	or, por Odín! Maldita sea.	
	3	

Capítulo II

Introducción

Aquí tienes la letra de la canción "Moonlight Shadow", tema principal de City Hunter:

The last time ever she saw him
Carried away by a moonlight shadow
He passed on worried and warning
Carried away by a moonlight shadow.
Lost in a riddle that Saturday night
Far away on the other side.
He was caught in the middle of a desperate fight
And she couldn't find how to push through

The trees that whisper in the evening Carried away by a moonlight shadow Sing a song of sorrow and grieving Carried away by a moonlight shadow All she saw was a silhouette of a gun Far away on the other side.

He was shot six times by a man on the run And she couldn't find how to push through

[Chorus]
I stay, I pray
See you in Heaven far away...
I stay, I pray
See you in Heaven one day.

Four A.M. in the morning
Carried away by a moonlight shadow
I watched your vision forming
Carried away by a moonlight shadow
A star was glowing in the silvery night
Far away on the other side
Will you come to talk to me this night
But she couldn't find how to push through

C Piscine Shell 00

[Chorus]

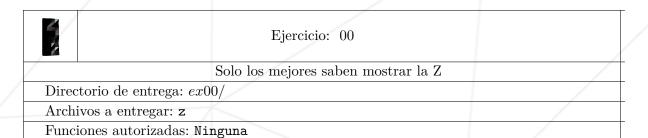
Far away on the other side.

Caught in the middle of a hundred and five
The night was heavy and the air was alive
But she couldn't find how to push through
Carried away by a moonlight shadow
Carried away by a moonlight shadow
Far away on the other side.

Desgraciadamente, estos enunciados no tienen nada que ver con \mathtt{City} $\mathtt{Hunter}.$

Capítulo III

Ejercicio 00 : Z

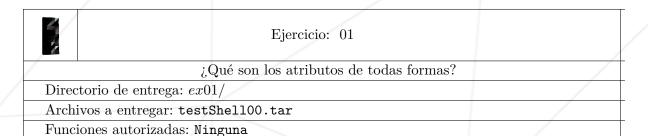


• Crea un archivo z que, cuando se ejecute un cat sobre él, muestre "Z" seguido de un salto de línea.

?>cat z
Z
?>

Capítulo IV

Ejercicio 01: testShell00



- Crea el archivo testShell00 en tu directorio de entrega.
- Encuentra la manera de que tu pantalla muestre lo siguiente (excepto por la línea de "total 1"):

```
%> ls -l
total XX
-r--r-xr-x 1 XX XX 40 Jun 1 23:42 testShell00
%>
```

• Una vez completados los pasos anteriores, ejecuta el siguiente comando para crear el archivo que debe ser entregado: tar -cf testShell00.tar testShell00.



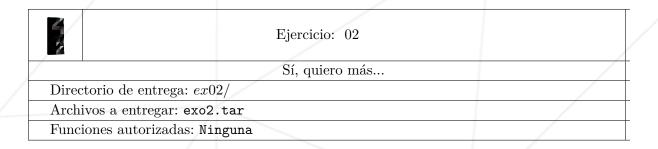
No te preocupes por lo que aparece en lugar de "XX"



Se acepta un año como respuesta en lugar de la hora.

Capítulo V

Ejercicio 02 : Sí, quiero más...



• Crea los siguientes archivos y directorios. Haz lo que sea necesario para que la ejecución de un ls -l en tu directorio muestre algo que se parezca a esto:

```
%> ls -1
total XX
drwx--xr-x 2 XX XX XX Jun 1 20:47 test0
-rwx--xr-- 1 XX XX 4 Jun 1 21:46 test1
dr-x--r-- 2 XX XX XX Jun 1 22:45 test2
-r---r-- 2 XX XX 1 Jun 1 23:44 test3
-rw-r---x 1 XX XX 2 Jun 1 23:43 test4
-r---r-- 2 XX XX 1 Jun 1 23:44 test5
lrwxr-xr-x 1 XX XX 5 Jun 1 22:20 test6 -> test0
%>
```

• Una vez resuelto el ejercicio, ejecuta el comando tar -cf exo2.tar * para crear el archivo que tendrás que entregar.



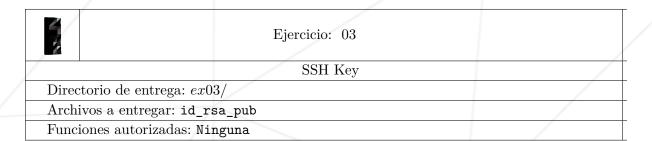
No te preocupes por lo que aparece en lugar de "XX"



Se acepta un año como respuesta en lugar de la hora.

Capítulo VI

Ejercicio 03: ¡Conéctame por SSH!



- Crea tu propia clave SSH. Cuando lo tengas:
 - o Añade tu clave pública a tu repositorio, en un archivo id_rsa_pub
 - o Actualiza tu clave ssh en la intranet. Esto te permitirá lanzar el repositorio a nuestro servidor git.



El nombre del archivo no ha sido elegido de manera aleatoria.



Asegúrate de entender la diferencia entre clave pública y clave privada.

Capítulo VII

Ejercicio 04 : midLS

4	Ejercicio: 04			
/	midLS	/		
Directorio de entrega: $ex04/$				
Archivos a entregar: midLS				
Funci	ones autorizadas: Ninguna	/		

• Crea en un archivo midLS que contenga el comando que hay que ejecutar para listar los archivos y los directorios del directorio actual, pero no los archivos ocultos, ni "."ni ".."(nada que empiece con un punto), separados por comas, ordenados por fecha de modificación y de tal modo que los directorios vayan seguidos por el carácter slash ("/").



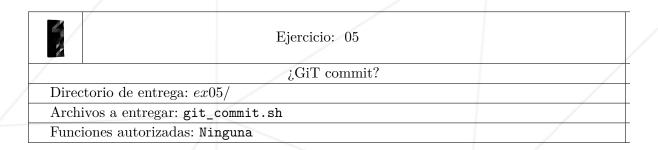
¡No hay que hacer lo que no se ha pedido!



RTFM!

Capítulo VIII

Exercise 05: GiT commit



• Crea un script de shell que muestre en pantalla los id de los últimos 5 commits de tu repositorio git.

```
%> bash git_commit.sh | cat -e
baa23b54f0adb7bf42623d6d0a6ed4587e11412a$
2f52d74b1387fa80eea844969e8dc5483b531ac1$
905f53d98656771334f53f59bb984fc29774701f$
5ddc8474f4f15b3fcb72d08fcb333e19c3a27078$
e94d0b448c03ec633f16d84d63beaef9ae7e7be8$
%>
```

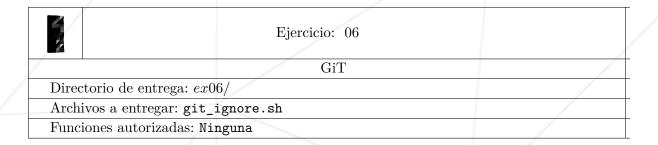
Para probar tu script, utilizaremos nuestro propio entorno.



RTFM!

Capítulo IX

Exercise 06: gitignore



• Crea un script de shell que muestre por pantalla la lista de archivos existentes ignorados por tu repositorio git actualmente presentes en tu repositorio local. Ejemplo:

```
%> bash git_ignore.sh | cat -e
.DS_Store$
mywork.c~$
%>
```

Para probar tu script, utilizaremos nuestro propio entorno.



RTFM!

Capítulo X

Ejercicio 07: diff



Ejercicio: 07

Directorio de entrega: ex07/

Archivos a entregar: b

Funciones autorizadas: Ninguna

• Crea el archivo b de tal modo que:

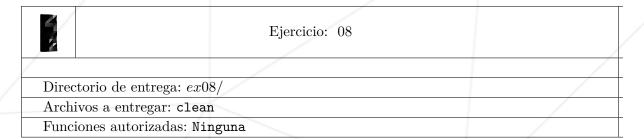
%>diff a b > sw.diff



man patch

Capítulo XI

Ejercicio 08 : clean



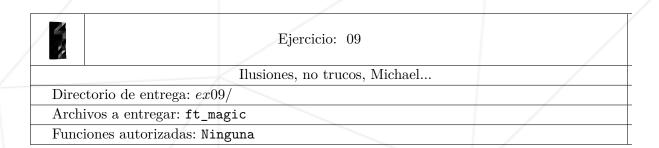
- Escriba en un archivo clean una línea de comandos que busque, a partir del directorio actual y en todos sus subdirectorios, los archivos cuyo nombre se termine por ~, o empiece y se termine por #.
- La línea de comandos mostrará y borrará los archivos encontrados.
- Solo se autoriza un comando: nada de ';' , '&&' ni otros trucos.



man find

Capítulo XII

Ejercicio 09 : Illusions, not tricks, Michael...



• Crea un archivo mágico que se llame ft_magic que tendrá que estar formateado correctamente para detectar los archivos de tipo 42 file y constituidos por la cadena "42" en el byte 42.



man file