







LABORATORIO DE SISTEMAS (GIRS)

## EXAMEN PRÁCTICO 1 - SHELL Y PYTHON SCRIPTING

- 
-  **Profesores:** Elena García-Morato y Felipe Ortega.
  -  **Lenguaje de programación:** Bash, sed, awk, Python.
  -  **Hora tope de entrega:** Miércoles, 22 de marzo de 2023 a las 14:40.
  -  **Envío:** Código fuente **comentado**, a través del correo de Aula Virtual.
- 

### INSTRUCCIONES

- Lee con detenimiento el enunciado de cada problema y plantea la solución antes de lanzarte a programar para resolverla.
- Puedes consultar el documento anexo con información de utilidad para resolver los ejercicios. No es posible consultar ninguna página de Internet ni acceder a los contenidos de Aula Virtual durante el examen.
- **Cada *script* debe estar almacenado en un fichero diferente. El nombre de cada fichero debe coincidir exactamente con el indicado en el enunciado del problema.**
- Una vez que termines, envía todos los *scripts* a través del espacio de entrega que se ha habilitado para el examen, en la sección Evaluación. No es necesario comprimir los archivos antes de enviarlos. Sube cada *script* al espacio de envío y luego confirma el envío para concluir el examen.
- **IMPORTANTE:** El espacio de envío se cierra automáticamente una vez superada la hora límite de finalización de la prueba. Por favor, no apures hasta el último momento para enviar tus respuestas.

## Ejercicio 1

Escribe un script de shell llamado `permissions.sh` que se comporte de la siguiente manera:

- ▶ Si se ejecuta el *script* sin ninguna opción, imprime por pantalla una lista de los **directorios** contenidos en el directorio actual o en alguno de sus subdirectorios (búsqueda recursiva), **incluyendo los ocultos**, así como **información completa** sobre los mismos (propietario, permisos, tamaño, última modificación, etc.).
- ▶ Si se ejecuta con la opción `+r` seguida de una letra, concede **permisos de lectura a todos los usuarios del sistema** para todos los **archivos regulares** contenidos en el directorio actual o en alguno de sus subdirectorios (búsqueda recursiva) cuyo nombre empiece por la letra indicada y cuya extensión sea `.txt`. En este caso no se debe aplicar dicha modificación a los ficheros ocultos.
- ▶ Si recibe cualquier otra opción como parámetro o un número incorrecto de parámetros debe imprimir un mensaje de error alertando sobre ello y **finalizar devolviendo el código de status** apropiado.

## Ejercicio 2

Dada la siguiente información sobre los usuarios registrados en un determinado servicio web (nombre, apellido, edad, lugar de residencia, id) que debes almacenar en un fichero denominado `ids.txt`:

John	Smith	32	New York	10001
Mary	Jones	45	Chicago	90001
David	Miller	27	Chicago	60601
Emily	Williams	31	Houston	77001
Daniel	Davis	38	Miami	33101
Julia	Martin	29	Miami	75201
Robert	Taylor	52	Philadelphia	19019
Sarah	Anderson	23	Chicago	85001
Michael	Brown	41	Atlanta	30301
Samantha	Lee	36	New York	94101

Escribe un *script* de shell llamado `ids.sh` que muestre por pantalla la información relativa a los 5 usuarios más jóvenes. A continuación, utiliza las **órdenes necesarias** para averiguar la ciudad con mayor número de usuarios registrados, así como el número de registros de la misma, almacenando estos datos en sendas variables.

Por último, imprime por pantalla dicha información siguiendo el siguiente formato:

[nombre-ciudad] es la ciudad con más personas registradas en este sitio web, con un total de [N] usuarios.

## Ejercicio 3

Escribe un *script* de shell con ayuda de los programas `sed` y `awk`, con el nombre `procesa_ls.sh`, que procese la salida del comando `ls -l` sobre un directorio cualquiera, produciendo el siguiente resultado:

- ▶ Solo se mostrarán las líneas correspondientes a archivos regulares. No se mostrarán directorios ni otros elementos.
- ▶ Se descarta la segunda columna (que muestra en la salida de `ls -l` el número de ítems contenidos en cada elemento). Esa columna **no debe aparecer** en la salida del *script*.
- ▶ La primera línea de la salida debe ser un título para el contenido de cada columna, por ejemplo:

PERMISOS	USUARIO	GRUPO	TAMAÑO	FECHA Y HORA	NOMBRE
----------	---------	-------	--------	--------------	--------

A continuación, se debe imprimir una línea en blanco para separar el título del resto del contenido.

- ▶ Se imprime el contenido de `ls -l`, quitando las líneas que no corresponden a ficheros regulares y la segunda columna de cada línea.
- ▶ Por último se debe imprimir una línea en blanco y, para acabar la salida del *script*, otra línea que indique el tamaño total de todos los archivos del directorio,
- ▶ Por ejemplo, si la salida del comando `ls -l` devuelve:

```
-rw-rw-r-- 1 jfelipe jfelipe 422 feb 20 16:25 ejemplo_parser.py
drwxrwxr-x 3 jfelipe jfelipe 4096 mar  9 00:09 gitrepo
-rwxrwxr-x 1 jfelipe jfelipe 42 feb 20 09:46 hello2.py
-rwxrwxr-x 1 jfelipe jfelipe 79 feb 20 09:50 hello3.py
-rwxrwxr-x 1 jfelipe jfelipe 46 feb 20 03:17 hello.py
```

La salida completa que produce el *script* debe ser:

PERMISOS	USUARIO	GRUPO	TAMAÑO	FECHA Y HORA	NOMBRE
-rw-rw-r--	jfelipe	jfelipe	422	feb 20 16:25	file1
-rwxrwxr--	jfelipe	jfelipe	42	feb 20 09:46	file2
-rwxrwxr--	jfelipe	jfelipe	79	feb 20 09:50	file3
-rwxrwxr--	jfelipe	jfelipe	46	feb 20 03:17	file4
TAMAÑO TOTAL: 589					

## Ejercicio 4

Crea un *script* en Python llamado `numeros.py` que se ajuste a las siguientes especificaciones:

- ▶ Como argumentos obligatorios debe recibir dos números enteros, separados entre sí por un espacio en blanco.
- ▶ Si recibe el parámetro opcional `-h` o bien `--help` imprime información de ayuda sobre la sintaxis para ejecutar el *script* y las opciones disponibles.
- ▶ Si recibe el parámetro opcional `-m` o bien `--mult` imprime el resultado obtenido al multiplicar ambos números. Ejemplo de ejecución:

```
$ python3 numeros.py --mult 3 2
El resultado de la multiplicación es: 6
```

- ▶ Si recibe el parámetro opcional `-p` o bien `--por` imprime el valor que resulta al elevar el primer número (base) al segundo (exponente). Ejemplo de ejecución:

```
$ python3 numeros.py --pow 3 2
El resultado de la potencia es: 9
```

- ▶ Si no se recibe ningún parámetro imprime una serie numérica que empiece en 1 y esté formada por tantos términos como indique el primer argumento, siendo cada término el anterior más el segundo número introducido como argumento.

```
$ python3 numeros.py 3 2
La serie es: 1 3 5
```