

<b>Actividad</b>	<b>Sistemas informáticos</b>
<b>Práctica (AP)</b>	U5.AP1. Shell scripts para desoxidar cerebro y manos

**Objetivos de la actividad:**

Los objetivos de la actividad son:

- Conocer los fundamentos sintácticos de Shell Scripts
- Utilizar las estructuras de entrada/salida en el desarrollo de Shell Scripts
- Utilizar las estructuras de comparación/repetición en el desarrollo de Shell Scripts
- Desarrollar Shell Scripts básicos para la automatización de tareas

**Temporalización:**

La duración prevista para dicha actividad es de 5 sesiones de 55 minutos y/o trabajo en casa

**Enunciado de la actividad:**

A continuación, se presentan **3 grupos de 6 ejercicios** (grupo I, grupo II y grupo III) **de los que solo tendremos que entregar uno de ellos**, mediante la siguiente regla:  
*“Si el resto de la división entera entre el número de letras de mi primer apellido y el número 3 es igual a:*

- 1, entonces haré los ejercicios del grupo I*
- 2, entonces haré los ejercicios del grupo II*
- 0, entonces haré los ejercicios del grupo III”*

Por ejemplo, si mi primer apellido es Barceló:

Barceló → 7 letras

$$7 \% 3 = 1$$

→ Haré los ejercicios del grupo I

Por ejemplo, si mi primer apellido es González:

González → 8 letras

$$8 \% 3 = 2$$

→ Haré los ejercicios del grupo II

Por ejemplo, si mi primer apellido es García:

García → 6 letras

$$6 \% 3 = 0$$

→ Haré los ejercicios del grupo III

**Desarrolla los 6 ejercicios (programas en Shell Script) del grupo de ejercicios asignado de forma autónoma. Deberás entregar los archivos .sh resultantes dentro de un archivo .zip.**

### **GRUPO I**

1. Escribe un script que pida al usuario su nombre, después sus dos apellidos, permanezca esperando durante 5 segundos y después muestre por pantalla un mensaje "Tu nombre completo es ...."
2. Escribe un script que pida al usuario un valor mayor que 0 y muestre un mensaje indicando si es par o impar. El script debe asegurar que el valor introducido es correcto; si no, deberá mostrar un mensaje y volver a pedirlo.
3. Escribe un script al que se le pase como argumento un valor mayor que 0 y muestre el listado de números que van desde 0 hasta este valor. El script debe asegurar que efectivamente se ha pasado un valor como argumento y que el valor introducido es correcto; si no, deberá mostrar un mensaje a modo de instrucciones de uso y salir.
4. Escribe un script que pida al usuario un valor del día numérico del mes (de 1 a 30) y muestre por pantalla qué día de la semana es, asumiendo que el día 1 es lunes. Para cualquier valor introducido fuera del rango, se mostrará un mensaje y se volverá a pedir.
5. Escribe un script llamado libros-anyadir.sh que pida al usuario el título de un libro, su año, su editorial y su género (entre varios géneros a elegir de los que se informará al usuario). El script insertará los datos en una nueva línea del archivo bdlibros.txt
6. Escribe un script libros.sh que ofrezca un menú con las opciones siguientes y las implemente: 1) Buscar por título. 2) Buscar por año. 3) Buscar por editorial. 4) Buscar por género. 5) Insertar libro. 6) Salir. NOTA: El script busca en el archivo bdlibros.txt del ejercicio anterior, que ya implementa el punto 5).

### **GRUPO II**

1. Escribe un script que pida dos cadenas de caracteres y las guarde en dos variables. Después, realizará la comparación de las cadenas e indicará por pantalla si las cadenas son iguales o diferentes.

2. Escribe un script al que se le pase como argumento un valor mayor que 0 y muestre un mensaje indicando si es par o impar. El script debe asegurar que efectivamente se ha pasado un valor como argumento y que el valor introducido es correcto; si no, deberá mostrar un mensaje a modo de instrucciones de uso y salir.
3. Escribe un script que pida al usuario un valor hasta que introduzca el 0. Cuando el usuario introduzca el 0, se mostrará un mensaje que muestre la suma total de valores introducidos así como el valor medio.
4. Escribe un script que pida al usuario la cantidad de litros de agua consumidos y muestre el coste total, atendiendo a las siguientes indicaciones:
  - a. Los primeros 50 litros cuestan 20 euros.
  - b. Los restantes hasta 200 litros cuestan a 20 céntimos el litro.
  - c. Los restantes cuestan a 10 céntimos el litro.
  - d. El script debe funcionar bien con cualquier valor mayor que 0.
5. Escribe un script llamado `micat` que recibe una ruta de fichero como parámetro e implementa la siguiente funcionalidad:
  - a. Si el fichero existe, lo mostrará de principio a fin.
  - b. Si no existe, informará al usuario de este hecho.
  - c. Adicionalmente se le puede pasar el parámetro `-r` que mostrará el fichero de fin a principio.
  - d. Estará disponible en el terminal como si de un comando más se tratara, i.e. sin necesidad de ejecutarlo con `./` delante.

NOTA: El comando `test` te será útil.
6. Escribe un script para generar números aleatorios de forma que pida al usuario cuántos números aleatorios quiere y entre qué valores mínimo y máximo han de estar. Tras ello, mostrará por pantalla los números aleatorios generados. NOTA: Infórmate sobre la variable `$RANDOM`.

### GRUPO III

1. Escribe un script que pida al usuario dos números, mostrando un mensaje de cuál es el mayor o de si son iguales.
2. Escribe un script que pida al usuario una nota y muestre un mensaje de la calificación que tiene (suspendido, aprobado, bien, notable o sobresaliente).

El script debe asegurar que el valor introducido es correcto; si no, deberá mostrar un mensaje y volver a pedirlo.

3. Escribe un script que le pida palabras al usuario, una a una, hasta que el usuario escriba “fin”. Tras ello, el programa indicará cuántas palabras se han introducido y las mostrará ordenadas alfabéticamente.
4. Escribe un script a modo de calculadora simple, al que se le pasan 3 argumentos: el primer valor, el operador (+, -, \*, /) y el segundo valor. El script deberá informar ante fallos habituales del usuario (p. ej. división entre 0 o no haber pasado los argumentos necesarios o en el orden adecuado). NOTA: Ayúdate del comando `expr`.
5. Escribe un script llamado `informa` que recibe una ruta de fichero como parámetro e implementa la siguiente funcionalidad:
  - a. Indica si el fichero existe o no.
  - b. Si existe, ofrece la siguiente información de forma amigable para el usuario:
    - i. Qué tipo de fichero es (directorio, fichero, enlace simbólico...)
    - ii. Qué permisos tiene el usuario sobre él
    - iii. Si el usuario es el propietario del archivo o no
    - iv. Si el fichero está vacío

NOTA: El comando `test` al rescate.

6. Escribe un script al que se le pase como parámetro un número `n` y muestre los `n` primeros elementos de la sucesión de Fibonacci.

#### **Evaluación de la actividad:**

Para la evaluación de la actividad se tendrán en cuenta el grado de resolución de los ejercicios, la presencia o no de fallos y la calidad del código.

#### **Recursos adicionales:**

Diapositivas y recursos de la Unidad en Florida Oberta

Robbins, Beebe. Classic Shell Scripting. O'Reilly

Búsqueda autónoma en Internet