

Examen 1 de la 3ª evaluación

La entrega de este examen se realizará utilizando [Github](#), por lo que tendrás que hacer lo siguiente:

- En tu cuenta de Github, crea un nuevo repositorio que llamarás `examen_iso`.
- Realiza los pasos necesarios para sincronizar ese directorio con una carpeta local que contendrá este fichero.
- Responde a las preguntas en este mismo documento en el espacio dedicado a tal efecto.
- Asegúrate de que este documento se publica mediante **Github Pages**
- La página principal del repositorio o índice debe incluir tu nombre y apellidos, así como un enlace a este fichero (con las respuestas) y otro a este mismo fichero guardado en PDF.

1.- Desde tu directorio personal, crea un fichero en `/tmp` llamado `ev3ex1.txt` que contenga tu nombre utilizando una única orden (por supuesto sin utilizar ningún editor).

```
javier@ASUS-TUF:~$ echo "Javier Calleja" > /tmp/ev3ex1.txt
```

2.- Muestra por pantalla el número de ficheros que tiene el directorio `/bin` (es decir, la salida tiene que ser únicamente un número).

```
javier@ASUS-TUF:~$ ls /bin | wc -l
```

3.- Muestra por pantalla todos los archivos del directorio `/bin` que tengan por lo menos dos vocales en su nombre, suponiendo que te encuentras en tu directorio personal.

```
javier@ASUS-TUF:~$ ls /bin | grep -E "^[aeiou]{2}*"
```

4.- Estando en tu directorio personal, muestra por pantalla la quinta línea del fichero `/etc/passwd`.

5.- Sabemos que el fichero `/usr/share/dict/spanish` contiene un listado de palabras en castellano. Calcula el número de palabras de este fichero que tienen un número impar de letras. Por tanto, la salida del comando deberá ser un único número.

6.- Indica qué expresiones regulares utilizarías para identificar los siguientes patrones, utiliza la sintaxis del motor ERE y suponiendo que la cadena buscada es el único contenido de cada línea. Se valora la precisión en

la expresión regular para que detecte únicamente el elemento que se pide.

a) Un NIF. Ejemplos: 2345678f, 71.555.333N, 10.333.222-T, ...

b) Una matrícula de coche (formato nuevo y antiguo). Ejemplos: 2288FSF, 4441KSD, LE3308G, A1234AJ, ...

c) Una fecha de la forma 04/05/2021

d) Una dirección MAC. Ejemplos: 00-09-0F-FE-00-01, 3C:A0:67:43:D3:92, ...

e) Una dirección IP. Ejemplos: 192.168.1.31, 10.0.0.1, 172.255.255.1, ...

7.- En clase hicimos una práctica en la que utilizábamos el editor **sed** para extraer datos de un fichero **csv** y convertirlos a formato **json** aunque es algo que se puede extrapolar a cualquier formato.

Supón que tienes un fichero CSV con la siguiente estructura:

```
nombre, apellidos, email, telefono, localidad
Victor, González Rodríguez, vgonzalez@mail.com, 666444555, León
Luis, Fernández Fernández, luis@mail.com, 888777666, Villabalter
...
```

Indica qué órdenes deberías introducir para automatizar la extracción de datos a un fichero XML de la forma:

```
<usuario>
  <nombre>Victor González Rodríguez</nombre>
  <telefono>666444555</telefono>
</usuario>
```

Observa que hay información del fichero CSV que se descarta (mail y localidad). Por otro lado, obviaremos las etiquetas de apertura y cierre del documento XML por claridad, limitándote a generar el código XML indicado.

```
cat archivo.csv | sed -e 's/,//' -e 's/, .*@.*.com//' -E -e 's/^(.*),(.*),
(.*)/<usua
rio>\n\t<nombre> \1</nombre>\n\t<telefono>\2</telefono>\n</usuario>/'
```

Entrega y criterios de calificación

El examen debe entregarse como se indica al principio de este documento. Si no pudieras o supieras entregarlo de esta manera podrás entregarlo alternativamente en formato PDF en el aula virtual.

	Puntuación
Se ha subido el examen a Github	0.5 punto
El examen es accesible mediante Github Pages	0.5 punto
Se ha creado una página de índice con los datos indicados	0.5 punto
Está disponible el examen en formato HTML y PDF	0.5 punto
Preguntas 1 a 5 (total 2 puntos)	0.4 puntos
Preguntas 6a a 6e (total 2 puntos)	0.4 puntos
7. Se genera el código pedido	0.75 punto
7. La salida incluye saltos de línea y tabulados	0.25 puntos