

1. Contar palabras en un archivo:

Desarrolla un script que reciba un archivo de texto como entrada y cuente el número de palabras que contiene. Puedes usar `grep -o` para buscar cada palabra individual y luego contarlas.

```
#!/bin/bash

# Solicitar al usuario que ingrese el nombre del archivo
read -p "Introduce el nombre del archivo de texto: " archivo

# Verificar si el archivo existe y es legible
if [ ! -r "$archivo" ]; then
    echo "Error: El archivo '$archivo' no existe o no es legible."
    exit 1
fi

# Contar el número de palabras en el archivo
num_palabras=$(grep -o '\w\+' "$archivo" | wc -w)

# Mostrar el resultado al usuario
echo "El archivo '$archivo' contiene $num_palabras palabras."
```

2. Extraer nombres de usuario de un archivo de registro:

Escribe un script que analice un archivo de registro y extraiga los nombres de usuario presentes en él. Puedes usar `grep` para buscar líneas que contengan nombres de usuario y luego `cut` para extraer solo los nombres de usuario.

```
#!/bin/bash

# Solicitar al usuario que ingrese el nombre del archivo de registro

read -p "Introduce el nombre del archivo de registro: " logfile

# Usar grep para buscar líneas que contienen nombres de usuario y luego cut para
extraer solo los nombres de usuario

usernames=$(grep -o 'user [^ ]*' "$logfile" | cut -d ' ' -f 2 | sort -u)

# Mostrar los nombres de usuario extraídos

echo "Nombres de usuario encontrados en el archivo de registro:"

echo "$usernames"
```

3. Sustituir caracteres en un archivo:

Crea un script que tome un archivo de texto como entrada y sustituya todos los espacios en blanco por guiones bajos. Puedes usar `tr` para realizar esta operación de sustitución.

```
#!/bin/bash
```

```
# Solicitar al usuario que ingrese el nombre del archivo de texto

read -p "Introduce el nombre del archivo de texto: " input_file

# Verificar si el archivo existe

if [ ! -f "$input_file" ]; then

    echo "El archivo $input_file no existe."

    exit 1

fi

# Crear un nuevo archivo de salida con el nombre modificado

output_file="{input_file%.txt}_modificado.txt"

# Sustituir los espacios en blanco por guiones bajos usando tr

tr ' ' '_' < "$input_file" > "$output_file"

# Mostrar un mensaje de éxito

echo "Se ha creado el archivo $output_file con los espacios sustituidos por guiones bajos."
```

4. Buscar coincidencias en archivos de registro:

Desarrolla un script que busque coincidencias de patrones específicos en varios archivos de registro y muestre las líneas que contienen esas coincidencias. Puedes usar `grep` para buscar las líneas que contienen los patrones especificados.

```
#!/bin/bash

# Patrones a buscar

pattern1="error"

pattern2="warning"


# Directorio donde se encuentran los archivos de registro

log_directory="/ruta/a/tu/directorio/de/registros"


# Verificar si el directorio existe

if [ ! -d "$log_directory" ]; then

    echo "El directorio $log_directory no existe."

    exit 1

fi


# Buscar coincidencias en los archivos de registro

echo "Coincidencias para el patrón '$pattern1':"

grep -r "$pattern1" "$log_directory"/*.log
```

```
echo

echo "Coincidencias para el patrón '$pattern2':"

grep -r "$pattern2" "$log_directory"/*.log
```

5. Extraer información del archivo de configuración:

Escribe un script que analice un archivo de configuración y extraiga cierta información, como direcciones IP o nombres de host. Puedes usar `grep` para buscar las líneas relevantes y luego `cut` para extraer la información específica que necesitas.

```
#!/bin/bash

# Archivo de configuración
config_file="/ruta/al/archivo/de/configuración.conf"

# Verificar si el archivo de configuración existe
if [ ! -f "$config_file" ]; then
    echo "El archivo de configuración $config_file no existe."
    exit 1
fi

# Extraer direcciones IP del archivo de configuración
echo "Direcciones IP encontradas en $config_file:"
grep -oE "\b([0-9]{1,3}\.){3}[0-9]{1,3}\b" "$config_file"

echo

# Extraer nombres de host del archivo de configuración
echo "Nombres de host encontrados en $config_file:"
grep -oE "\b[a-zA-Z0-9]+\.[a-zA-Z]+\b" "$config_file"
```