



UNIVERSIDAD NACIONAL
AUTÓNOMA DE MÉXICO



FACULTAD DE INGENIERÍA

BASES DE DATOS AVANZADAS

Ejercicio práctico 02

Programación en Shell y variables de entorno

Alumno(s):

Flores Martínez Emanuel

Grupo: 1

Profesor:

Ing. Jorge Alberto Rodríguez Campos

26 de febrero de 2021

Semestre 2021-2

1. Objetivos

Practicar los conceptos básicos y fundamentales de la programación Shell en específico GNU Bash (Bourne-again shell). Estos conceptos serán empleados durante el curso como parte de las actividades de administración de bases de datos.

2. Actividades

C1. Código fuente del programa

```
# @Autor Emanuel Flores Martínez
# @Fecha 26/02/2021
# @Descripcion Script para descargar y comprimir imágenes

# parámetros
# proteger nuestro programas (shell injection)
archivoImágenes="${1}"
numImágenes="${2}"
archivoZip="${3}"

#
# Función que se encarga de mostrar el contenido de el archivo
# s-02-ayuda.sh y terminar la ejecución en caso de encontrar un error
#
function ayuda(){
    status="${1}"
    cat s-02-ayuda.sh
    exit "${status}"
}

# Validar parámetros de entrada
if [ -z "${archivoImágenes}" ]; then
    echo "[-] ERROR: El nombre del archivo de imágenes es obligatorio"
    ayuda 100
elif ! [ -f "${archivoImágenes}" ]; then
    echo "[-] ERROR: El archivo ${archivoImágenes} no existe"
    ayuda 101
fi;

# Validar el número de imagenes
# =~ -> Expresión regular
if ! [[ "${numImágenes}" =~ [0-9]+ && "${numImágenes}" -gt 0 &&
    "${numImágenes}" -le 90 ]]; then
    echo "[-] ERROR: El número de imagenes ${numImágenes} es inválido"
    ayuda 102
fi;

# Validar el parámetro 3. Si se especifica, verificar que exista el
# directorio
if [ -n "${archivoZip}" ]; then
    # extraer la ruta del archivo zip
    dirSalida=$(dirname "${archivoZip}")
```

```
nombreZip=$(basename "$archivoZip")
if ! [ -d "${dirSalida}" ]; then
    echo "[-] ERROR: El directorio ${dirSalida} no existe"
    ayuda 103
fi;
else
    dirSalida="/tmp/${USER}/imagenes"
    mkdir -p "${dirSalida}"
    nombreZip="imagenes-$(date '+%Y-%m-%H-%M-%S').zip"
fi;

echo "Parámetros correctos, obteniendo imágenes..."
# limpiar el directorio en caso de no estar vacío
rm -rf "${dirSalida}/${nombreZip}"

# Leer el contenido del archivo que contiene la lista de imágenes
# que se van a descargar
count=1
while read -r linea
do
    wget -q -P "${dirSalida}" "${linea}"
    # valida el código de salida de wget
    status=$?
    if ! [ ${status} -eq 0 ]; then
        echo "[-] ERROR: No se pudo obtener el archivo de ${linea}"
        ayuda 104
    fi;

    if [ ${count} -ge ${numImagenes} ]; then
        break;
    fi;

    # Evalua la expresión aritmética y lo asigna a count
    count=$((count+1))

done < "${archivoImagenes}"

# Generando archivo zip
export IMG_ZIP_FILE="${dirSalida}/${nombreZip}"

zip -j ${IMG_ZIP_FILE} "${dirSalida}/*.jpg"

# eliminando imagenes descargadas
rm -rf "${dirSalida}/*.jpg"

# cambiando permisos
chmod 700 "${IMG_ZIP_FILE}"

# Generando archivo s-00-lista-archivos.txt
# -Z nombre de los archivos sin ruta
# -l listar sin descomprimir
unzip -Z1 "${IMG_ZIP_FILE}" > "${dirSalida}/s-00-lista-archivos.txt
```

C2. Salida de ejecución del validador

```
[emanuel@pc-efm ejercicio02]$ ./s-03-validador-main-enc.sh
Validador Tema 01 - Ejercicio práctico 02
===== Validacion de resultados =====
1. Fecha.....2021-02-26 09:25:49
2. Nombre del Host ..... pc-efm.fi.unam
3. Nombre del usuario ..... emanuel
4. fv_hsh_user ..... 21

Para mayores detalles de posibles errores revisar los siguientes archivos:
/tmp/s-03-validador.log y /tmp/s-03-validador.err
001021-E2021-02-26P09:25:49 - OK - Invocando sin argumentos
002021-M2021-02-26C09:25:49 - OK - Código correcto 100 al invocar sin parámetros
003021-A2021-02-26-09:25:49 - OK - Código correcto 101 al invocar con archivo de imágenes inválido
004021-N2021-02-26E09:25:49 - OK - Código correcto 102 al invocar con número de imágenes -5
005021-U2021-02-26F09:25:49 - OK - Código correcto 102 al invocar con número de imágenes 150
006021-E2021-02-26M09:25:49 - OK - Código correcto 103 con ruta de salida inexistente
007021-L2021-02-26-09:25:49 - OK - Invocando con 89 imágenes
008021-E2021-02-26F09:26:38 - OK - Directorio de salida válido
009021-M2021-02-26I09:26:38 - OK - Variable IMG_ZIP_FILE definida: /tmp/emanuel/imagenes/imagenes-2021-02-09-25-49.zip
010021-A2021-02-26-09:26:38 - OK - Archivo encontrado: /tmp/emanuel/imagenes/imagenes-2021-02-09-25-49.zip
011021-N2021-02-26U09:26:38 - OK - Permisos del archivo zip válidos
012021-U2021-02-26N09:26:38 - OK - Número correcto de imágenes en zip: 89
013021-E2021-02-26A09:26:38 - OK - Archivo s-02-ayuda.sh correcto
014021-L2021-02-26M09:26:38 - OK - Archivo s-00-lista-archivos.txt correcto
015021-E2021-02-26P09:26:38 - OK - Validación concluida.
```

3. Conclusiones

Este ejercicio práctico en mi opinión es uno de los más útiles ya que se ve lo que es Shell Scripting, que aunque parezca un poco complicado e incluso algo tardado, puede ahorrar una cantidad considerable de tiempo a largo plazo, ya que permite automatizar tareas sencillas o complicadas que pueden ser repetitivas.

Además se revisaron conceptos del mismo sistema operativo como pueden ser las redirecciones, variables de entorno y algunos comandos básicos, por lo que este ejercicio práctico además es muy completo y nos permite ver la utilidad de temas vistos en materias anteriores.

Referencias

- [1] Oracle. *Oracle Database Documentation* en <https://docs.oracle.com/en/database/oracle/oracle-database/index.html>
- [2] Oracle Database. *Database PL/SQL Language Reference* en <https://docs.oracle.com/en/database/oracle/oracle-database/12.2/lnpls/database-pl-sql-language-reference.pdf>