TEMA 01
Ejercicio práctico 02
Programación en Shell y variables de entorno.

NOMBRE: **GRUPO:**

FECHA DE ENTREGA: **CALIFICACION:**

1.1. OBJETIVO

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.

1.2. LINEAMIENTOS DE FORMATO DE CÓDIGO Y CONSULTAS.

1.2.1. Formato de código

En algunos ejercicios se solicita incluir el código fuente de los programas en cual deberá ser incluido en el documento de entrega debidamente formateado. La indentación deberá ser a 2 espacios sin utilizar tabulaciones. El tipo de letra deberá ser Courier new tamaño 8 o equivalente.

Ejemplo:

```
set serveroutput on
declare
  cursor cur_reprobados is
    select rownum, e.nombre nombre_estudiante,e.apellido_paterno, e.apellido_materno,
           a.nombre nombre_asignatura, ei.calificacion
    from estudiante e, asignatura a, curso c, estudiante_inscrito ei
    where a.asignatura_id = c.asignatura_id
    and c.curso_id = ei.curso_id
    and e.estudiante_id = ei.estudiante_id
    and ei.calificacion = 5;
begin
  -- código
  -- códifgo
end;
/
```

Este lineamiento aplica para cualquier lenguaje de programación.

1.2.2. Obtención de consultas.

Otro aspecto importante son las consultas. En diversos ejercicios se solicita la inclusión de datos como resultado de realizar consultas SQL. En algunos casos las consultas pueden realizarse en alguna herramienta gráfica, por ejemplo SQL Developer. De ser el caso, se puede copiar el resultado de la consula en el formato de la herramienta.

Ejemplo:

	ORA_ROWSCN	ORA_ROWSCN_TS	ID	NOMBRE
1	2064191	05-MAY-20 04.53.47.000000000 PM	1	Juan
2	2064191	05-MAY-20 04.53.47.000000000 PM	2	Euduardo
3	2064191	05-MAY-20 04.53.47.000000000 PM	3	Maribel

En algunas ocasiones la consulta puede realizarse el SQL*Plus. En este caso se deberán realizar configuraciones adicionales de tal forma que la consulta se visualice de forma correcta. Los comandos principales que permiten configurar el formato son los comandos `col, linesize y pagesize`.

Ejemplo:

Suponer que se realiza la siguiente consulta en SQL*Plus

```
select username, to char(created,'DD/MM/YYYY HH:MI:SS') fecha creacion,
  to char(expiry date,'DD/MM/YYYY HH:MI:SS') fecha expiracion
from user_users;
```

La salida de la consulta se ve así:

```
USERNAME
--------------------------------------------------------------------------------
FECHA_CREACION     FECHA_EXPIRACION
------------------ ------------------
EJEMPLOS
13/10/2015 08:40:05 10/04/2016 08:40:05
```

- Lo anterior se debe a que SQL *Plus reserva la longitud máxima de una columna para mostrarla en pantalla. En este caso el campo `username` está definido como un `varchar2(128)`. Por esta razón aparecen demasiados espacios y guiones para el campo `username`.
- Para corregir lo anterior, es posible limitar el número de caracteres máximo empleados para mostrar una columna empleando el comando `column`

```
col[umn] <column_name> format A<size>
```

Ejemplo:

```
sql> column username format A30
sql> set linesize window
sql> set pagesize 100
```

Al ejecutar nuevamente la sentencia, se obtiene una salida mucho mejor:

```
USUARIO                        FECHA_CREACION      FECHA_EXPIRACION
------------------------------ ------------------- -------------------
EJEMPLOS                       13/10/2015 08:40:05 10/04/2016 08:40:05
```

- Adicional a lo anterior, se recomienda agrandar la terminal para que las columnas puedan ser mostradas de forma correcta.

1.3. EJERCICIOS A RESOLVER.

Generar un Shell script llamado `s-01-procesa-imagenes.sh` empleando el intérprete Bash. A nivel generar este script deberá descargar una lista de imágenes de Internet y deberá copiarlas a una determinada ruta dentro del sistema de archivos para finalmente construir archivos zip que contienen las imágenes descargadas.

Para todos los Shell scripts que se realicen en el curso, se deberá agregar el siguiente encanezado.

```
# @Autor         <Nombre del autor>
# @Fecha         dd/mm/yyyy
# @Descripcion   <Breve descripción del programa>
```

El programa deberá recibir 3 parámetros.

- Primer parámetro: Ruta donde se encuentra el archivo con la lista de imágenes que serán descargadas. El archivo que contiene esta lista se encuentra en la carpeta compartida de la asignatura.
- Segundo parámetro: El número de imágenes a descargar.
- Tercer parámetro: nombre del archivo zip a generar. Este parámetro es opcional. En caso de no ser especificado, su valor será `imagenes-yyyy-mm-dd-hh:mi:ss.zip` Los valores `yyyy-mm-dd-hh:mi:ss` corresponden a la fecha en la que se generó el archivo.

El programa deberá validar los parámetros de entrada. En caso de ser incorrectos deberá mostrar un mensaje de ERROR y un texto de ayuda que explique la forma en la que debe ejecutarse. Este texto deberá estar contenido en un archivo llamado `s-02-ayuda.sh` El programa deberá crear una función llamada `ayuda` que se encargue de leer el contenido el archivo de ayuda y mostrarlo en pantalla. El texto de ayuda es el siguiente:

```
Para ejecutar el programa utilizar la siguiente sintaxis:

./s-01-procesa-imagenes.sh  <path_archivo_imagen> <num_imagenes> [<nombre_archivo_zip>]

<path_archivo_imagen>    Ruta y nombre donde se encuentra el archivo de texto que contiene un URL
                         por cada renglón de la imagen que será descargada de internet.
<num_imagenes>           Total de imágenes a descargar, entre 1 y hasta 90 imágenes
[<nombre_archivo_zip>]   Parámetro opcional que indica la ruta y nombre del archivo zip a generar.
                         En caso de no especificarse se empleará el nombre
                         imagenes-yyyy-mm-dd-hh:mi:ss.zip y estará ubicado en
                         ${TMP}/imagenes/${USER}/imagenes. Si se especifica este parámetro, su valor
                         deberá incluir la ruta y el nombre del archivo zip a generar. La ruta
                         absoluta y el nombre del archivo zip generado deberá ser exportado empleando
                         una variable de entorno IMG_ZIP_FILE
```

Validación de parámetros

- En caso de no especificar el archivo de imágenes, el programa deberá detenerse y generar el código de salida 100
- Si el archivo de imágenes no existe, el programa deberá detenerse y generar el código 101.
- Si se especifica un número incorrecto de imágenes, el programa deberá detenerse y generar el código 102
- Si se especifica un directorio de salida para guardar el archivo zip y este no existe, el programa deberá detenerse y generar el código 103
- En caso contrario, el programa deberá leer el contenido del archivo de valores. Con base al número de imágenes proporcionado, el programa deberá descargar las imágenes en la ruta destino. Si alguna de las imágenes no puede ser descargada por algún error, el programa deberá detenerse y generar el código de salida 104.
- Posteriormente, el programa deberá armar el archivo zip incluyendo las imágenes descargadas.
- Una vez que el archivo zip haya sido generado, el programa deberá eliminar las imágenes descargadas.
- El archivo zip deberá contar únicamente con permisos de lectura y escritura para el usuario que invocó el programa. Para los demás usuarios no se deberán tener permisos de ningún tipo.
- El programa deberá exportar una variable de entorno llamada `IMG_ZIP_FILE` cuyo valor contendrá la ruta absoluta y el nombre del archivo zip generado.
- El programa deberá generar un archivo llamado `s-00-lista-archivos.txt` Dicho archivo deberá contener una lista con los nombres de los archivos incluidos en el archivo zip. Los nombres no deben incluir rutas.

1.4. VALIDADOR

- Obtener todos los archivos de la carpeta correspondiente a este ejercicio práctico. Copiarlos a la misma carpeta donde se encuentra el programa.
- Ejecutar el validador: `./s-03-validador-main-enc.sh`
- Verificar que no existan errores de validación.

1.5. CONTENIDO DE LA ENTREGA.

- C1. Código fuente del programa
- C2. Salida de ejecución del validador.
- Elementos generales indicados en la rúbrica general de ejercicios prácticos (datos generales, conclusiones y comentarios).
- Entrega individual