




| | | |
|---|---|---|
|  | UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA |  |
| Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación | | |
| Aprobación: 2022/03/01 | Código: GUIA-PRLD-001 | Página: 1 |

GUÍA DE LABORATORIO

(formato docente)

| INFORMACIÓN BÁSICA | | | | | |
|---|--|-----------------------|------------|-----------------------|----|
| ASIGNATURA: | PROGRAMACIÓN DE SISTEMAS | | | | |
| TÍTULO DE LA PRÁCTICA: | Introducción a la programación en Bash | | | | |
| NÚMERO DE PRÁCTICA: | 01 | AÑO LECTIVO: | 2023- A | NRO. SEMESTRE: | V |
| TIPO DE PRÁCTICA: | INDIVIDUAL | | | | |
| | GRUPAL | MÍNIMO DE ESTUDIANTES | 03 | MÁXIMO DE ESTUDIANTES | 04 |
| FECHA INICIO: | 04/05/2023 | FECHA FIN: | 13/05/2023 | DURACIÓN: | 04 |
| RECURSOS Y EQUIPOS A UTILIZAR: | | | | | |
| Linux | | | | | |
| DOCENTE(s): Magister Edith Giovanna Cano Mamani Magister Jorge Manuel Polanco Argüelles | | | | | |

| OBJETIVOS/TEMAS Y COMPETENCIAS | |
|--|--|
| OBJETIVOS: La presente práctica de laboratorio tiene como objetivo principal de que el estudiante se familiarice con la programación Shell – Scripts en Linux | |
| TEMAS: ✓ Shell – Scripts en Linux | |
| COMPETENCIAS A ALCANZAR | C. Diseña responsablemente sistemas, componentes o procesos para satisfacer necesidades dentro de restricciones realistas: económicas, medio ambientales, sociales, políticas, éticas, de salud, de seguridad, manufacturación y sostenibilidad. |
| | M. Construye responsablemente soluciones siguiendo un proceso adecuado llevando a cabo las pruebas ajustada a los recursos disponibles del cliente. |
| | |
| | |

| | | |
|---|---|---|
|  | <p style="text-align: center;">UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p> |  |
| <p style="text-align: center;">Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p> | | |
| <p>Aprobación: 2022/03/01</p> | <p>Código: GUIA-PRLD-001</p> | <p>Página: 2</p> |

CONTENIDO DE LA GUÍA

I. MARCO CONCEPTUAL

El intérprete de comandos o Shell es un programa que permite a los usuarios interactuar con el sistema, procesando las órdenes que se le indican. Los comandos invocables desde el Shell pueden clasificarse en internos (corresponden en realidad a órdenes interpretadas por el propio Shell) y externos (corresponden a ficheros ejecutables externos al Shell). Además de comandos, los Shells ofrecen otros elementos para mejorar su funcionalidad, tales como variables, funciones o estructuras de control. El conjunto de comandos internos y elementos disponibles, así como su sintaxis, dependerá del Shell concreto empleado.

Además de utilizar el Shell desde la línea de comandos (basada en el Prompt como la indicación del Shell para anunciar que espera una orden del usuario), puede emplearse para la interpretación de Shell-scripts. Un Shell-script o “guión de órdenes” es un fichero de texto que contiene un conjunto de comandos y órdenes interpretables por el Shell.

La duración de laboratorio (actividades y ejercicios resueltos y propuestos) será de 04 horas académicas

II. EJERCICIO RESUELTO

1. Crear archivos usando el comando cat en Linux


```
$ cat > ejemplo.txt
```

Presionar CTRL + D para terminar de escribir el archivo
 Para ver el contenido del archivo usamos

```
$cat ejemplo.txt
```

 Excepcionalmente se puede crear archivos usando el comando touch
2. Uso de \$@ para mostrar lista de parámetros en bash

La variable \$ * contiene todos los parámetros como un solo valor.
 La variable \$ @ contiene todos los parámetros como valores separados, por lo que puede iterar sobre ellos.

```
#!/bin/bash total=1
for var in "$*"; do
echo "\$* =====> #$total = $var"
total=$((total + 1))
done
total=1
for var in "$@"; do
echo "\$@ =====> #$total = $var"
total=$((total + 1))
done
```

Para comprobar el resultado:

```
$bash myscript.sh 1 2 3 4 5
```

3. Uso de \$# para saber la cantidad de parámetros en bash

Para obtener la cantidad de parámetros pasados, puede utilizar esta variable (\$#).

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

```
echo hay $# parametros pasados
```

Para comprobar el resultado:

```
bash myscript.sh 1 2 3 4 5
```

4. Uso del operador de comparación -le para saber si un número es menor o igual al segundo número

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

```
if [ 2 -le 3 ]
```

```
then
```

```
echo "2 es el menor o igual"
```

```
else
```

```
echo "2 no es el menor"
```

```
fi;
```

Para comprobar el resultado:

```
bash myscript.sh
```

5. Creación de una función sin parámetros en bash

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

```
Hola_mundo () {
```

```
echo 'hola, mundo'
```

```
}
```

```
Hola_mundo
```

Para comprobar el resultado:

```
bash myscript.sh
```

6. Uso del comando who

Cuando ejecutamos who sin ningún argumento, será desplegada la información de la cuenta con detalles como nombre de usuario, terminal del usuario, hora de inicio de sesión y del host en el cual se ha iniciado sesión de la siguiente forma:

```
$who
```

Si deseamos imprimir el encabezado de las columnas generadas, debemos usar el parámetro -H de la siguiente forma:

```
$who -H
```



Para imprimir en pantalla los nombres de inicio de sesión y la cantidad total de usuarios conectados, basta con usar el parámetro -q así:

```
$who -q
```

Si el objetivo es desplegar solo el nombre de host y el usuario asociado con stdin, podremos hacer uso del parámetro -m:

```
$who -m
```

7. Uso del comando grep para buscar una palabra en una lista

| | | |
|---|--|---|
|  | <p style="text-align: center;">UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p> |  |
| <p style="text-align: center;">Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p> | | |
| <p>Aprobación: 2022/03/01</p> | <p>Código: GUIA-PRLD-001</p> | <p>Página: 4</p> |

Comprobamos que Documents esta en la lista generada por "ls"
\$ ls | grep Documents

8. Uso de comandos cp para crear copias de seguridad de todo el contenido de una carpeta

Cuando queremos copiar un directorio podemos usar -r o -R pero también podemos usar la opción -a para archivar los elementos incluidos en el directorio (archivos y carpetas) y mediante ese archivado crear una copia exacta de esos archivos y directorios, incluyendo enlaces simbólicos.

cp -a directorio respaldo

III. EJERCICIOS PROPUESTOS

1. Realizar un **script llamado '01-hola-mundo.sh'** que muestre por pantalla "Hola mundo!".
2. Ídem pero que en vez de "mundo" muestre los parámetros introducidos (**'02-hola-parametros.sh'**).
3. Ídem y que además verifique que al menos hayamos introducido un parámetro (**'03-hola-al-menos-1-parametro.sh'**).
4. Ídem y que además separe cada argumento por ", " (**'04-hola-parametros-separados.sh'**).
5. Ídem y que además en caso de error muestra una ayuda (**'05-hola-con-ayuda.sh'**).
6. Ídem y que además verifique que sean usuarios conectados al sistema (**'06-hola-usuario.sh'**).
7. Realizar un script llamado **'usuarioconectado'** que retorna un SI si el primer parámetro coincide con algún usuario conectado o NO en caso contrario.
8. Modificar el fichero '.bashrc' para modificar el PATH y añadir la carpeta de estos ejercicios. Para ello añade la siguiente línea: **export PATH=\$PATH:~/ruta_carpeta_ejercicios**
9. Modificar el script '06holausuario.sh' para que llame a 'usuarioconectado' (**'09-hola-usuario.sh'**).
10. Realizar un script llamado **'usuariosistema'** que retorna un SI si el primer parámetro coincide con algún usuario del sistema o NO en caso contrario.

IV. CUESTIONARIO

1. ¿Por qué el Bash script es importante dentro de Linux?
2. ¿Existe otra forma de crear un archivo ejecutable en Linux sin usar bash?

V. REFERENCIAS Y BIBLIOGRAFÍA RECOMENDADAS:



- [1] LinuxCabal, (2012, Diciembre, 9), Programación en Bash - Capítulo 1, [Online], Available: <https://www.youtube.com/watch?v=m9Ve556jMb4>
- [2] Duhnnae, (2017, Julio, 15), Tutorial para aprender bash - Linux Terminal – Scripts [Online], Available: <https://www.youtube.com/watch?v=g05j4Dvlqfw>
- [3] Ivam3byCinderella, (2018, Junio, 24) Curso de Programacion en Bash Script desde Termux (1ra parte-uso de Termux), [Online], Available: https://www.youtube.com/watch?v=jWvLi6qy1_A

TÉCNICAS E INSTRUMENTOS DE EVALUACIÓN

| | |
|------------------|----------------------|
| TÉCNICAS: | INSTRUMENTOS: |
| <i>Trabajo</i> | <i>Rubricas</i> |

CRITERIOS DE EVALUACIÓN Y LOGROS ALCANZADOS

| Criterios | Aprendizaje Alto | Aprendizaje Bueno | Aprendizaje Medio | Aprendizaje Bajo |
|---|---|---|--|---|
| Planificar el tratamiento del problema | Se planifica completa y adecuadamente el tratamiento del problema | Se planifica completamente el tratamiento del problema con observaciones en acápitos | Se planifica parcialmente el tratamiento del problema | No planifica el tratamiento del problema |
| Organizar el trabajo del equipo/grupo | Se organiza el trabajo del equipo completa y adecuadamente | Se organiza el trabajo del equipo completamente | Se organiza el trabajo del equipo parcialmente | No se organiza el trabajo del equipo |
| Problema | Se describe, identifica, enuncia, determinando las causas y efectos de manera muy claro y puntual del problema a investigar | Se describe, identifica, enuncia, determinando las causas y efectos de manera bien claro y puntual el problema a investigar | Se describe, y/o identifica, y/o enuncia, y/o determina las causas y efectos de manera parcial el problema a | No se trata el problema que se va a investigar |
| Marco Teórico | Los estudiantes saben cómo buscar, organizar y compartir la información nueva, muy bien sobre el problema | Los estudiantes saben cómo buscar, organizar y compartir la información nueva, bien sobre el | Los estudiantes saben cómo buscar, organizar y compartir la información nueva, regular sobre el problema | Los estudiantes no saben cómo buscar, organizar y compartir la información nueva, sobre el problema |
| Comparativa de la selección de aspectos | Se compara muy bien los aspectos de productos u otros elementos | Se compara bien los aspectos de productos u otros elementos | Se compara parcialmente los aspectos de productos u otros elementos | No se compara los aspectos de productos u otros elementos |

| | | |
|--|--|---|
|  | <p align="center">UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p> |  |
| <p align="center">Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p> | | |
| <p>Aprobación: 2022/03/01</p> | <p align="center">Código: GUIA-PRLD-001</p> | <p align="right">Página: 6</p> |

| | | | | | |
|---|--|--|---|--|--|
| <p>Trabajar en grupo, colaborativamente con compañeros evitando trabajar solo</p> | <p>Los estudiantes trabajan muy bien colaborativamente con sus compañeros reflejándose en la lista de actividades realizadas</p> | <p>Los estudiantes trabajan bien colaborativamente con sus compañeros reflejándose en la lista de actividades realizadas</p> | <p>Los estudiantes trabajan regular colaborativamente con sus compañeros reflejándose en la lista de actividades realizadas</p> | <p>Los estudiantes No trabajan colaborativamente con sus compañeros reflejándose en la pobre lista de actividades realizadas</p> | |
| <p>Generación de posibles soluciones (Alternativas)</p> | <p>Se genera muy bien propuestas de posibles soluciones</p> | <p>Se genera bien propuestas de posibles soluciones</p> | <p>Se genera regular propuestas de soluciones</p> | <p>No se plantea propuestas de soluciones</p> | |
| <p>Presentación de la Solución</p> | <p>La presentación de la solución está muy bien elaborada con calidad en su</p> | <p>La presentación de la solución está bien elaborada con calidad en su</p> | <p>La presentación de la solución está regularmente elaborada con</p> | <p>La presentación de la solución No está elaborada de acuerdo a lo</p> | |