



INFORME SISTEMAS OPERATIVOS

**PRÁCTICA DE CONTROL
SCRIPT PARA EL MANEJO DE TRADUCCIONES**

JUNIO DE 2024
AUTOR: ALBERTO CUERVO PRIETO
Grupo: 201

Contenido

INTRODUCCIÓN	2
FUNCIONAMIENTO	3
ADVERTENCIAS:.....	3
1. Menú de inicio	3
2. Escribir referencias en archivos no referenciados	4
3. Cambiar lenguaje de scripts ya referenciados	6
4. Re-referenciar scripts	7
5. Añadir y eliminar lenguajes	9
6. Salida.....	10
ESTRUCTURA DEL CÓDIGO	11
Comentarios del código	11
Inicio	11
Bucle principal	12
CONCLUSIÓN	17
REFERENCIAS	18

INTRODUCCIÓN

Bash, abreviatura de "Bourne Again SHell", es un intérprete de comandos y lenguaje de scripting ampliamente utilizado en sistemas Unix y similares, como Linux y macOS. Nacido como una mejora del shell original de Unix, el Bourne Shell (sh), Bash ofrece una interfaz potente y flexible para interactuar con el sistema operativo. Esto permite a los usuarios ejecutar comandos, crear scripts y automatizar tareas de diversa complejidad. Su compatibilidad con múltiples sistemas operativos y su capacidad para manejar scripts avanzados han hecho de Bash una herramienta indispensable tanto para administradores de sistemas como para desarrolladores. Una de las características más notables de Bash es su eficiencia en la manipulación de archivos y procesos.

En este trabajo estaremos utilizando este lenguaje para realizar un script que realice la gestión de comentarios de los scripts de los subdirectorios, haciendo posible realizar traducciones y modificaciones en los mismos, siendo guardados e identificados mediante las siglas mas significativas del idioma y el número de comentario.

La numeración se realizará de 10 en 10 y podremos añadir comentarios entre medias indicando la correspondiente referencia, podremos crear nuevos idiomas, eliminarlos, regenerar referencias, referenciar scripts sin referencias y cambiar los lenguajes de los comentarios de los scripts, siendo guardados en los respectivos ficheros de idioma.

FUNCIONAMIENTO

ADVERTENCIAS:

Tendremos que iniciar el código en una carpeta libre de archivos basura de otros procesos como, por ejemplo, archivos de idioma de otros códigos.

El usuario debe de modificar los comentarios dentro del script y **NO** directamente en los **ARCHIVOS .txt**

1. Menú de inicio

Al iniciar el script se nos presentará la información de nuestro programa en consola, entre ello su función, la versión, el autor, tutor, universidad y año de realización.

```
PROGRAMA PARA REALIZAR TRADUCCIONES EN LOS CÓDIGOS DE GESTIÓN DE LOS SUBDIRECTORIOS
VERSIÓN 3.1
Autor: ALBERTO CUERVO PRIETO
Tutor: JOSÉ MANUEL SÁIZ DIEZ
Grado Ing. Informática (2023-2024)
Universidad de Burgos
```

Más abajo se nos mostrarán los idiomas encontrados en el script (guardados al inicio del mismo en comentarios iniciados por #?) que al inicio siempre tenemos español inglés y francés por defecto:

```
Idiomas encontrados:
Espanol Inglés Francés
```

Además, se buscarán archivos de los diferentes idiomas, en caso de no encontrarlos se crearán automáticamente esos archivos **VACÍOS** para cada uno de los códigos .sh de los subdirectorios a la espera de que se realice una referenciación:

```
No existen archivos .txt que comiencen con ES
Procesando archivo: 1-GestionDeProcesosYMemoria-FI-FNI-PrimerMejorPeor.sh
Procesando archivo: 2-GestionDeProcesosYMemoria-SN.sh
Creados los archivos de lenguaje vacíos en el idioma ES
No existen archivos .txt que comiencen con EN
Procesando archivo: 1-GestionDeProcesosYMemoria-FI-FNI-PrimerMejorPeor.sh
Procesando archivo: 2-GestionDeProcesosYMemoria-SN.sh
Creados los archivos de lenguaje vacíos en el idioma EN
No existen archivos .txt que comiencen con FR
Procesando archivo: 1-GestionDeProcesosYMemoria-FI-FNI-PrimerMejorPeor.sh
Procesando archivo: 2-GestionDeProcesosYMemoria-SN.sh
Creados los archivos de lenguaje vacíos en el idioma FR
```

Después de ello se nos mostrará el menú principal de opciones:

- ```
=====
1. Crear referencias en scripts ya comentados(Sin referencias) y sincronizar referencias de todos los idiomas
2. Cambiar lenguaje de scripts ya referenciados
3. Re-referenciar scripts (Reescribir números de las referencias)
4. Añadir lenguaje
5. Eliminar lenguaje
6. Salir
=====
```

En este podremos elegir entre todas las opciones del script, para elegir cualquiera de ellos tendremos que introducir el índice de la opción deseada.

## 2. Escribir referencias en archivos no referenciados

Esta opción nos permite referenciar todos los comentarios de los códigos de los subdirectorios con la marca de su respectivo idioma y número de comentario.

El inicio de todos los comentarios estará marcado primero por la marca de comentario, en nuestro caso será “#-“, por el código del lenguaje indicado por dos letras mayúsculas, por ejemplo, español lo indicamos con “ES”, el número de comentario (10,20,30...) y el fin de referencia marcado por “-”.

Antes de la ejecución de esta opción, nuestros scripts tendrán los comentarios con la siguiente estructura:

```
#!/bin/bash
#####
#
AUTORES: Los alumnos de la asignatura Sistemas Operativos del Grado en Ingeniería :
Universidad de Burgos
TUTOR: José Manuel Saiz Diez
#
#####
Script realizado para la simulación de un SO que utilice diferentes modelos de gestio
total de 32 opciones diferentes.
El algoritmo de gestión de procesos se basará en FCFS, SJF, SRPT, Prioridad Mayor y l
Apropiativo) y Round-Robin.
La gestión de memoria se implementa según las necesidades de los procesos y podrá ser
Reubicable y no resubicable.
```

Para referenciarlos elegiremos la opción 1 en el menú de inicio:

```
=====
1. Escribir referencias en scripts ya comentados (Sin referencias)
2. Cambiar lenguaje de scripts ya referenciados
3. Re-referenciar scripts (Reescribir números de las referencias)
4. Añadir lenguaje
5. Eliminar lenguaje
6. Salir
=====
Selecione una opción: 1
```

Se nos pedirá indicar el idioma en que se encuentran los comentarios actualmente para guardarlos en su archivo de traducción correspondiente:

```
Elige un idioma con su índice
1: Espanol
2: Inglés
3: Francés
¿Cuál es el lenguaje actual? 1
```

Una vez elegido el idioma, comenzarán a generarse las referencias y más tarde copiarlas a los archivos de idioma, tanto a los del mismo idioma como a los de los idiomas restantes.

```
¿Cuál es el lenguaje actual? 1
Creando referencias en 1-GestionDeProcesosYMemoria-FI-FNI-PrimerMejorPeor
Referencias creadas y comentarios copiados al lenguaje ES
Creando referencias en 2-GestionDeProcesosYMemoria-SN
Referencias creadas y comentarios copiados al lenguaje ES

Sincronizando referencias (Esto puede tardar algún minuto)
1-GestionDeProcesosYMemoria-FI-FNI-PrimerMejorPeor en ES
1-GestionDeProcesosYMemoria-FI-FNI-PrimerMejorPeor en EN
1-GestionDeProcesosYMemoria-FI-FNI-PrimerMejorPeor en FR
2-GestionDeProcesosYMemoria-SN en ES
2-GestionDeProcesosYMemoria-SN en EN
```

Nuestro código final quedará de la siguiente forma, con todos los comentarios referenciados correctamente con la numeración de 10 en 10:

```
#!/bin/bash
#####
#-ES10-#
#-ES20-# AUTORES: Los alumnos de la asignatura Sistemas Operativos del Grado en Ingeniería
Universidad de Burgos
#-ES30-# TUTOR: José Manuel Saiz Diez
#-ES40-#
#####
#-ES50-# Script realizado para la simulación de un SO que utilice diferentes modelos de gestión
con un total de 32 opciones diferentes.
#-ES60-# El algoritmo de gestión de procesos se basará en FCFS, SJF, SRPT, Prioridad Máxima
Apropiativo) y Round-Robin.
#-ES70-# La gestión de memoria se implementa según las necesidades de los procesos y memoria
y Reubicable y no reubicable.
#####
#-ES80-# VARIABLES DE EJECUCIÓN
#####
#-ES90-# Configuración de los parámetros de simulación
```

Todos estos comentarios se copiarán en los archivos de idioma correspondientes a cada archivo:

### Archivo de idioma español:

```
#-ES10-#
#-ES20-# AUTORES: Los alumnos de la asignatura Sistemas Operativos del Grado en Ingeniería Informática (2014-2024) -
Universidad de Burgos
#-ES30-# TUTOR: José Manuel Saiz Diez
#-ES40-#
#-ES50-# Script realizado para la simulación de un SO que utilice diferentes modelos de gestión de procesos y memoria
con un total de 32 opciones diferentes.
```

### Archivo de idioma francés:

```
#-FR10-#
#-FR20-#
#-FR30-#
#-FR40-#
#-FR50-#
#-FR60-#
```

## 3. Cambiar lenguaje de scripts ya referenciados

Una vez tenemos escritas las referencias en los archivos podemos proceder a realizar cambios de idioma.

Mediante esta opción realizaremos varias acciones, primeramente guardaremos los comentarios referenciados del script en su archivo de idioma, más tarde comprobaremos que en los demás archivos de idioma tenemos las mismas referencias y por último introduciremos los comentarios en el idioma escogido, lo que significa

que si añadimos un comentario extra, por ejemplo el “#-ES45-“, este también se escribirá en los demás idiomas:

```
#-ES20-# AUTORES: Los alumnos de la asignatura Sistemas Operativos del Grado en Ingeni
Universidad de Burgos
#-ES30-# TUTOR: José Manuel Saiz Diez
#-ES40-#
#-ES45-# Este comentario no estaba antes|
#-ES50-# Script realizado para la simulación de un SO que utilice diferentes modelos de
```

Elegimos la opción 2 en el menú de inicio indicando el idioma actual y el idioma al que queremos cambiar el idioma:

Elige un idioma con su índice

1: Español

2: Inglés

3: Francés

¿Cuál es el lenguaje actual? 1

¿Cuál es el lenguaje al que quieres cambiar? 2

Creando referencias en ES1-GestionDeProcesosYMemoria-FI-FNI-PrimerMejorPeor.txt  
Creando referencias en ES2-GestionDeProcesosYMemoria-SN.txt

Sincronizando referencias (Esto puede tardar algún minuto)

1-GestionDeProcesosYMemoria-FI-FNI-PrimerMejorPeor en ES

1-GestionDeProcesosYMemoria-FI-FNI-PrimerMejorPeor en EN

1-GestionDeProcesosYMemoria-FI-FNI-PrimerMejorPeor en FR

2-GestionDeProcesosYMemoria-SN en ES

2-GestionDeProcesosYMemoria-SN en EN

Una vez terminado el proceso, se habrán reemplazado los comentarios:

```
#!/bin/bash
#####
#-EN10-
#-EN20-
#-EN30-
#-EN40-
#-EN45-
#####
#-EN50-
#-EN60-
#-EN70-
#####
```

Se encuentran vacíos ya que no hemos escrito ninguna traducción por ahora.

#### 4. Re-referenciar scripts



Esta opción nos permite reescribir los números de las referencias, haciendo que vuelvan a separarse de 10 en 10, es decir, que si elegimos esta acción en este momento, teniendo el comentario “#-EN45-” se nos renumerará como “#-EN50-”, así como en todos los archivos de idioma:

Elige un idioma con su índice

1: Español

2: Inglés

3: Francés

¿Cuál es el lenguaje actual? 2

Sincronizando referencias (Esto puede tardar algún minuto)

1-GestionDeProcesosYMemoria-FI-FNI-PrimerMejorPeor en ES

1-GestionDeProcesosYMemoria-FI-FNI-PrimerMejorPeor en EN

1-GestionDeProcesosYMemoria-FI-FNI-PrimerMejorPeor en FR

2-GestionDeProcesosYMemoria-SN en ES

2-GestionDeProcesosYMemoria-SN en EN

2-GestionDeProcesosYMemoria-SN en FR

Reescribiendo referencias en 1-GestionDeProcesosYMemoria-FI-FNI-PrimerMejorPeor

Reescribiendo referencias en 2-GestionDeProcesosYMemoria-SN



### Comentarios en el script

```
#!/bin/bash
#####
#-EN10-
#-EN20-
#-EN30-
#-EN40-
#-EN50-|
#####
#-EN60-
```

### Comentarios en el archivo de idioma español

```
#-ES10-#
#-ES20-# AUTORES: Los alumnos de la asignatura Sistemas Operati
Universidad de Burgos
#-ES30-# TUTOR: José Manuel Saiz Diez
#-ES40-#
#-ES50-# Script realizado para la simulación de un SO que utilice
con un total de 32 opciones diferentes.
#-ES60-# El algoritmo de gestión de procesos se basará en FCFS, (
Apropiativo) y Round-Robin.
#-ES70-# La gestión de memoria se implementa según las necesidades
v Reubicable v no reubicable.
```

## 5. Añadir y eliminar lenguajes

Esto nos permitirá generar traducciones en otros idiomas a los iniciales (Español, Inglés y Francés). Para añadir un nuevo lenguaje tendremos que elegir la opción número 4, indicar sus dos letras mayúsculas que lo identifiquen y su nombre completo de la siguiente forma:

```
3. Re-referenciar scripts (Reescribir números de las referencias)
4. Añadir lenguaje
5. Eliminar lenguaje
6. Salir
=====
Seleccione una opción: 4
Ingrese el nuevo idioma en mayúsculas (2 caracteres): CH
Ingrese el nombre completo del idioma: Chino
Procesando archivo: 1-GestionDeProcesosYMemoria-FI-FNI-PrimerMejorPeor.sh
Procesando archivo: 2-GestionDeProcesosYMemoria-SN.sh
█
```

Una vez realizado se nos habrá guardado el idioma de la siguiente forma en el código:

```
#!/bin/bash
#?ES-Espanol
#?EN-Inglés
#?FR-Francés
#?CH-Chino
```

Y se nos mostrará en el menú junto con los demás idiomas:

```
Lenguajes en el sistema:
Espanol Inglés Francés Chino
```

Para eliminar un idioma tendremos que usar la opción 5 del menú y posteriormente elegiremos el idioma a eliminar mediante el número de su índice:

Seleccione una opción: 5

Elige un idioma con su índice

1: Espanol

2: Inglés

3: Francés

4: Chino

Ingrese el idioma a eliminar 4

Automáticamente se nos eliminará el idioma tanto del código como del menú.

Lenguajes en el sistema:

Espanol Inglés Francés

```
#!/bin/bash
#?ES-Espanol
#?EN-Inglés
#?FR-Francés
```

## 6. Salida

Para salir del programa utilizaremos la opción número 6 del menú que finalizará la ejecución el script:

```
=====
1. Crear referencias en scripts ya comentados(Sin referencias) y sincronizar referencias de todos los idiomas
2. Cambiar lenguaje de scripts ya referenciados
3. Re-referenciar scripts (Reescribir números de las referencias)
4. Añadir lenguaje
5. Eliminar lenguaje
6. Salir
=====
```

Seleccione una opción: 6

Saliendo...

alberto@alberto-VirtualBox:~/Versions/SIN PARÁMETRO -I\$

## ESTRUCTURA DEL CÓDIGO

En este apartado explicaremos el funcionamiento general del código, el manejo de idiomas, creación, sincronización y cambio de referencias.

### Comentarios del código

El funcionamiento de cada una de las funciones está explicado mediante comentarios dentro del script, marcando con líneas de almohadillas los inicios de funciones junto a los parámetros utilizados por ellas:

#### Comentarios de inicio

```
5
6
7 #####
8 # AUTOR: Alberto Cuervo Prieto
9 # GRADO INGENIERÍA INFORMÁTICA AÑO 2023-24 || Asignatura Sistemas Operativos
10 #TUTOR: José Manuel Saiz Diez
11 #####
12 # Este script nos permite manejar las traducciones de los comentarios de los archivos .sh
13 # de los subdirectorios, teniendo las opciones de:
14 # REFERENCIAR, CAMBIAR REFERENCIAS, RE-REFERENCIAR, AÑADIR Y ELIMINAR LENGUAJE
15 #####
16
17
18
```

#### Comentarios de función

```
#####
Añadimos referencias a los scripts y metemos los comentarios a su archivo de lenguaje correspondiente
Parámetro1: directorio sobre el que están los archivos a re-referenciar
Parámetro2: lenguaje actual
#####

change_comment_start() {
 local directory="$1"
 local language="$2"
 local num=10

 # Encuentra todos los archivos .sh dentro del directorio dado y sus subdirectorios
}
```

### Inicio

Al inicializar el programa este realiza unos procedimientos previos a la entrada en el bucle principal:

```
find_languages
welcome_screen
```

Las primeras funciones a las que llamamos son **find\_languages** y **welcome\_screen**.

“find\_languages” es la función que encuentra los lenguajes guardados en el mismo código, es decir, lee todos los comentarios comenzados por “#?” y guarda las siglas de los lenguajes y sus nombres completos en arrays separados

```
#!/bin/bash
#?ES-Espanol
#?EN-Inglés
#?FR-Francés
```

La función “welcome\_sscreen” nos muestra la pantalla de bienvenida con la versión año, autor y tutor.

Posteriormente, en el bucle for comprobamos si existen los archivos txt de los idiomas y en caso de no existir los crea **vacíos**, si todavía no se ha referenciado, o con el número de referencias que tengamos en los archivos .sh .

```
echo -e "\nCOMPROBAMOS SI EXISTEN ARCHIVOS DE IDIOMA, EN CASO NEGATIVO CREAMOS LOS ARCHIVOS -->>>VACÍOS<<<--\n"
for lang in "${languages[@]}; do
 # Comprobar si existe al menos un fichero .txt que inicie con el idioma actual
 if ls ./"/${lang})*.txt 1> /dev/null 2>&1; then
 echo "Existen archivos .txt que comienzan con $lang"
 else
 echo "No existen archivos .txt que comiencen con $lang"
 create_reference_files "$lang"
 fi
done
```

## Bucle principal

Este bucle while se repetirá hasta que elijamos la opción de salir en el menú.

```
Bucle principal del menú
while true; do
 print_languages
 show_menu
 read -p "Seleccione una opción: " option
```

Primeramente, se ejecutarán las funciones **print\_languages** (que mostrará todos los lenguajes dentro del array de lenguajes) y **show\_menu** que como su nombre indica mostrará las opciones que tiene el programa:

```
Función para mostrar el menú
show_menu() {
 echo ""
 echo
 "=====
 echo "1. Crear referencias en scripts ya comentados(Sin referencias) y sincronizar referencias de todos los idiomas"
 echo "2. Cambiar lenguaje de scripts ya referenciados"
 echo "3. Re-referenciar scripts (Reescribir números de las referencias)"
 echo "4. Añadir lenguaje"
 echo "5. Eliminar lenguaje"
 echo "6. Salir"
 echo
 "=====
}
```

Posteriormente usaremos el comando **read** para leer la opción escogida por el usuario

```
read -p "Seleccione una opción: " option
```

Tras ello usaremos un **case** con la variable **option** en el que dirigimos la ejecución en función de la elección del usuario:

```
case $option in
 1)
 show_languages
 read -p "¿Cuál es el lenguaje actual? " j
 k=1
 if check_valid_data "$j" "$k" ; then
 actual_language=${languages[$((j-1))]}
 change_comment_start "./" "$actual_language"
 sync_references "$actual_language"
 else
 echo "Introduce datos válidos"
 ..
 ..
```

En los diferentes casos posibles se usan las funciones del código, las más utilizadas son:

- “**show\_languages**” y “**check\_valid\_data**” que como su nombre indica, muestran por pantalla los idiomas que se encuentran en el sistema junto a su índice y posteriormente se detecta si el valor introducido es válido:

```
show_languages
read -p "¿Cuál es el lenguaje actual? " j
k=1
if check_valid_data "$j" "$k" ; then
 actual_language=${languages[$((j-1))]}
```

- “**change\_comment\_start**” es la función principal de la opción número 1, esta referencia los archivos .sh y guarda los comentarios en su txt de traducción correspondiente. Recibe como parámetros el directorio de uso y el idioma actual de los scripts:

```
change_comment_start "./" "$actual_language"
```

```
Añadimos referencias a los scripts y metemos los comentarios a su archivo de lenguaje correspondiente
change_comment_start() {
 local directory="$1"
 local language="$2"
 local num=10

 # Encuentra todos los archivos .sh dentro del directorio dado y sus subdirectorios
 find "$directory" -mindepth 2 -type f -name "*.sh" -print0 | while IFS= read -r -d '' file; do
 # Obtener el nombre base del archivo sin la extensión
 base_name=$(basename "$file" .sh)
 # Definir el archivo de salida
 output_file="./$(dirname "$file")/${language}${base_name}.txt"
 # Definir el archivo temporal
 temp_file=$(mktemp)

 echo "Creando referencias en $base_name"

 # Procesar el archivo, modificarlo y redirigir los comentarios modificados al archivo de salida
 awk -v num=10 -v language="$language" -v output_file="$output_file" '
 BEGIN {
 OFS = " "
 # Abrir el archivo de salida para escribir los comentarios modificados
 out = output_file
 print "" > out # Limpiar el archivo de salida
```

- “**sync\_references**” es una función que nos permite sincronizar las referencias de los archivos de traducciones, es decir, que si existe una referencia que no está recogida en los demás archivos esta se añadirá en todos los txt con la ejecución de esta función. Recibe como parámetro el lenguaje actual:

```
sync_references "$actual_language"
```

```
Función para comparar y sincronizar referencias en archivos de diferentes idiomas con el introducido por parámetro a la función
sync_references() {
 local lang="$1"
 echo ""
 echo "Sincronizando referencias (Esto puede tardar algún minuto)"
 find "/" -mindepth 2 -type f -name "*.sh" -print0 | while IFS= read -r -d '' file; do
 for language in "${languages[@]};do
 local contador=0
 echo "${basename "$file" .sh} en $language"
 # Leer cada línea del archivo de entrada y compararlo
 while IFS= read -r linea; do

 local numero=$(echo "$linea" | grep -oP "#-${lang}\K\d+")
 if ["$lang" != "$language"] && ! grep -q "#-${language}$numero-" "$file" && [-n "$numero"]; then

 sed -i "${contador}a\##-${language}$numero-" "$file" && [-n "$numero"]; then
 #echo "Actualizando referencia #-${language}$numero- -> #-${language}$numero-"
 echo "Actualizando referencia #-${language}$numero- -> #-${language}$numero-"
 fi
 ((contador++))
 done < "$file" && [-n "$numero"]; then
 done
 done
 done
}
```

- “**copy\_comments**” y “**replace\_comments**” son las funciones principales de la opción número 2, la primera de ellas se encarga de copiar los comentarios referenciados del archivo .sh a su archivo txt de idioma correspondiente mientras que la segunda reemplaza los comentarios del script por los del lenguaje a cambiar. Ambas reciben como parámetro el lenguaje sobre el que tienen que realizar la operación:

```
copy_comments "/" "$actual_language"
```

```
replace_comments "/" "$change_language"
```

- “**re-reference**” realiza la función re-referenciar los comentarios, buscando referencias que no se encuentren en los demás archivos y añadiéndolas en ese caso, es la función principal de la opción número 3. Recibe como parámetro el lenguaje actual del script:

```
re_reference "$actual_language"
```

- “**add\_language**”, “**delete\_language**”, “**save\_languages**” y “**validate\_language**” como sus nombres indican, estas funciones nos permiten manejar los idiomas guardados en el código, podremos guardar, añadir y eliminar los idiomas del código, guardados como comentarios iniciados por “#?” y separadas las siglas



de los nombres completos por "-". También nos permiten verificar si los lenguajes introducidos por el usuario son válidos, es decir, si las siglas del idioma son dos letras mayúsculas.

```
4) add_language "ES";;
```

```
5) delete_language;;
```

```
validate_language "$new_lang"
```

```
save_languages
```

Por último tenemos la opción de salida que nos permitirá salir del bucle principal y, por tanto, salir del programa:

```
6) echo -e "\nSaliendo..."; break ;;
```

## **CONCLUSIÓN**

A lo largo de este trabajo, hemos explorado en profundidad el uso de Bash, una herramienta fundamental para la automatización y gestión de tareas en sistemas Unix y similares. Nos hemos centrado en la creación de un script capaz de gestionar los comentarios en scripts de subdirectorios, permitiendo su traducción y modificación. Este proceso nos ha permitido aplicar tanto los conceptos básicos como avanzados de Bash, demostrando su eficacia en la manipulación de archivos y procesos.

Ha sido todo un reto aprender Bash desde cero, pero a lo largo del proyecto he adquirido las habilidades necesarias para implementarlo con éxito. He aprendido a numerar sistemáticamente los comentarios, gestionar múltiples idiomas y asegurar la correcta referencia y actualización de comentarios en diversos scripts. También he aprendido a crear, eliminar y regenerar referencias, así como a traducir comentarios de manera efectiva. Este trabajo no solo ha consolidado mi conocimiento de Bash, sino que también ha demostrado su versatilidad y eficiencia en la administración de tareas complejas en entornos de desarrollo y producción. A pesar de las dificultades iniciales, he logrado sacar adelante el proyecto con éxito.

## **REFERENCIAS**

1. TLDP. (2001). **Advanced Bash-Scripting Guide**. The Linux Documentation Project. Recuperado de <http://www.tldp.org/LDP/abs/html/>
2. GNU. (2024). **Bash Reference Manual**. GNU Project. Recuperado de <https://www.gnu.org/software/bash/manual/bash.html>
3. Stack Overflow. (n.d.). **Bash - Stack Overflow**. Recuperado de <https://stackoverflow.com/questions/tagged/bash>
4. Linuxize. (2019). **Bash Scripting Tutorial for Beginners**. Linuxize. Recuperado de <https://linuxize.com/post/bash-scripting-tutorial-for-beginners/>
5. DigitalOcean. (2020). **How To Write Bash Scripts**. DigitalOcean. Recuperado de [https://www.digitalocean.com/community/tutorial\\_series/how-to-write-bash-scriptsC](https://www.digitalocean.com/community/tutorial_series/how-to-write-bash-scriptsC)
6. GeeksforGeeks. (2021). **Introduction to Bash Script**. GeeksforGeeks. Recuperado de <https://www.geeksforgeeks.org/introduction-to-bash-script/>