



# Sistema de Guardado de Notas

Samuel Vila Camino  
Juan Cabaleiro Feijóo  
Laura González Lemos  
Gabriel Orejas Guerra



# Introducción

Nuestro proyecto consiste en la creación de un sistema de ficheros que servirá a la hora de tomar notas, para una mejor organización de las mismas.

## OBJETIVO:

El objetivo principal de nuestro sistema de ficheros es ayudar a alumnos familiarizados con el entorno linux a tomar apuntes. Para ello se almacenarán todas las notas recogidas entre unos tags y se plasmarán en un archivo de texto con un nombre representativo elegido por el usuario.



# Funcionalidades Adicionales

Además de crear el sistema de ficheros en el que se recogerán los apuntes, hemos añadido las siguientes funcionalidades:

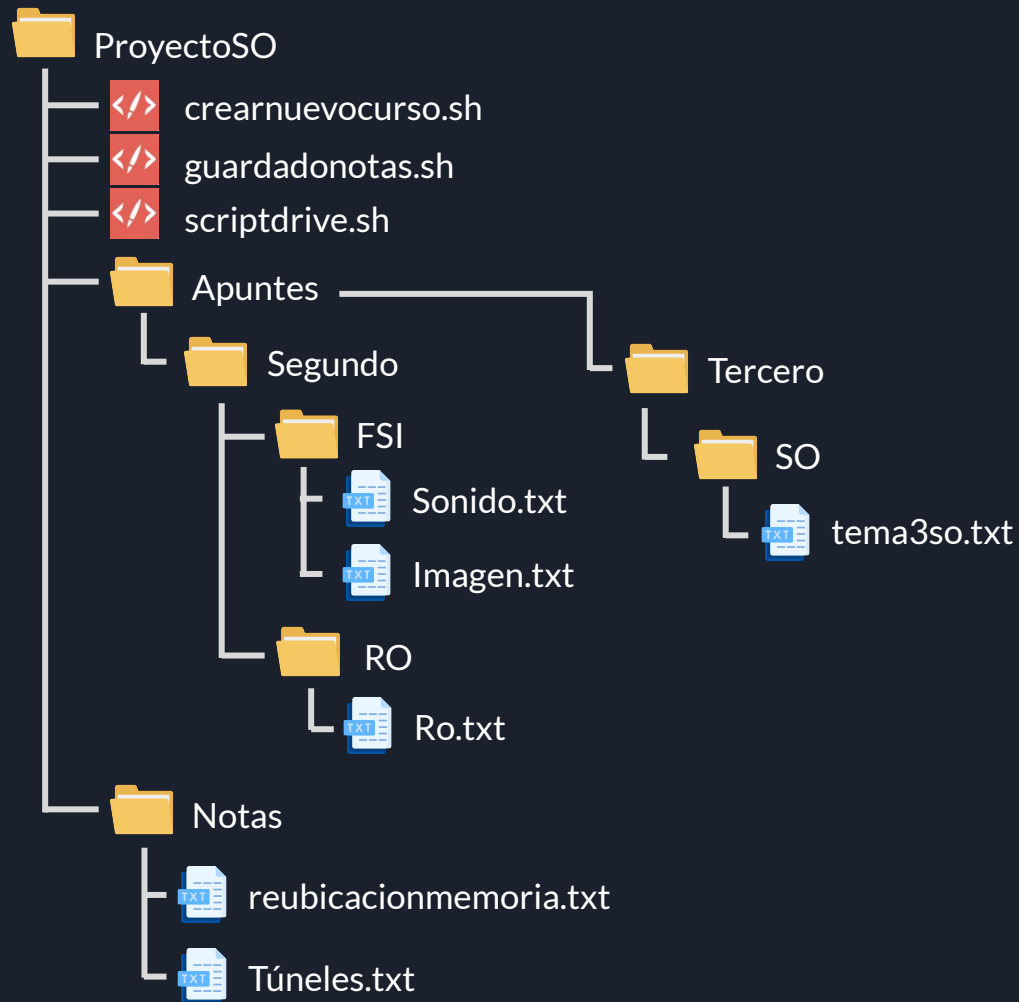
-Creación de un nuevo curso:

Consiste en la posibilidad mediante un script de crear un nuevo curso (directorio) con las asignaturas correspondientes elegidas por el alumno.

-Copia de seguridad:

Consiste en la creación de una carpeta en la nube (en nuestro caso Google Drive) mediante un script, en la que se hará una copia de seguridad de la carpetas Notas y Apuntes.

# Estructura



# Estructura: Archivos de texto

Archivos .txt que pueden contener tags:

<<ejemplotag>>Texto<<ejemplotag>>

En cuanto a la reubicación de la memoria, hay dos formas:

-Estática: en tiempo de compilación

<<reubicacionmemoria>>En estática, dirección logica=física.<<reubicacionmemoria>>  
El compilador genera código máquina con direcciones absolutas.

-Dinámica: en tiempo de compilación

<<reubicacionmemoria>>En dinámica, dirección lógica ≠ física.<<reubicacionmemoria>>  
Código reubicable: el código máquina contiene direcciones relativas al comienzo del programa.  
Las direcciones físicas se calculan en tiempo de ejecución...

(Apuntes/tercero/so/tema3so.txt)

<<reubicacionmemoria>>En estática,

=física.<<reubicacionmemoria>>



# guardadonotas.sh

Empleamos **scripts de shell** en Linux:

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

```
echo "Dentro de este directorio, las partes de los ficheros con notas son: "  
grep -rn "<<" --exclude="*.sh*" Apuntes/  
echo "-----"
```

```
for archivo in $(find Apuntes -type f -name "*.txt")  
do
```

```
    echo "Procesando el archivo: $archivo"  
    patron="<<\K[^<]+(?:=>)"  
    nombres_totales_tag=$(grep -oP "$patron" "$archivo")
```

```
    for nombre_tag in $nombres_totales_tag
```

```
    do  
        archivo_salida="Notas/${nombre_tag}.txt"  
  
        sed -n "/<<${nombre_tag}>>.*<<${nombre_tag}>>/p" "$archivo" > "$archivo_salida"
```

```
    done  
    echo "Se ha guardado el contenido entre los tags '$nombre_tag' en el archivo '$archivo_salida'"
```

```
    echo ""
```

```
done  
echo "-----"  
echo "Proceso de copia finalizado"
```

```
exit 0
```

# guardadonotas.sh

<<

\K : reinicia la coincidencia

[^<]+ : excluimos un posible < del tag de cierre

(?=>>) : solo considera coincidencia si termina en ">>"

```
patron="<<\K[^<]+(?=>>)"
```

```
nombres_totales_tag=$(grep -oP "$patron" "$archivo")
```

o : solo imprime lo que coincide con el patrón  
P : activa el soporte para regex



## guardadonotas.sh

Comando sed:

```
sed 's/cadenaactual/cadenacambio/' archivo.txt
```

```
sed '/patron/d' archivo.txt
```



# guardadonotas.sh

\n  
/: inicio patrón  
.\* : cualquier cantidad de caracteres  
/: fin patrón  
p : imprime la línea que coincide

```
sed -n "/<<${nombre_tag}>>.*<<${nombre_tag}>>/p" "$archivo" > "$archivo_salida"
```



# guardadonotas.sh

```
-----  
ubuntu@ubuntu:~/Desktop/ProyectoS0$ ./guardadonotas.sh
```

Dentro de este directorio, las partes de los ficheros con notas son:

Apuntes/segundo/ro/RO.txt:3:<<redesdeordenadores>> Las redes son... <<<redesdeordenadores>>

Apuntes/segundo/ro/RO.txt:7:<<redesdeordenadores>> Lass 2 redes son... <<<redesdeordenadores>>

Apuntes/tercero/S0/gestionmemoria.txt:9:<<reubicacionmemoria>>En estática, dirección logica=fisica.<<<reubicacionmemoria>>

Apuntes/tercero/S0/gestionmemoria.txt:13:<<reubicacionmemoria>>En dinámica, dirección lógica ≠ fisica.<<<reubicacionmemoria>>

-----

Procesando el archivo: Apuntes/segundo/ro/RO.txt

Se ha guardado el contenido entre los tags 'redesdeordenadores' en el archivo 'Notas/redesdeordenadores.txt'

Procesando el archivo: Apuntes/tercero/S0/gestionmemoria.txt

Se ha guardado el contenido entre los tags 'reubicacionmemoria' en el archivo 'Notas/reubicacionmemoria.txt'

-----

Proceso de copia finalizado

```
ubuntu@ubuntu:~/Desktop/ProyectoS0$ █
```



# guardadonotas.sh

Resultado:

```
<<reubicacionmemoria>>En estática, dirección logica=fisica.<<<reubicacionmemoria>>  
<<reubicacionmemoria>>En dinámica, dirección lógica ≠ física.<<<reubicacionmemoria>>
```

(Notas/reubicacionmemoria.txt)



# crearnuevocurso.sh

Comando read:

```
read -p "¿Cuántas asignaturas tiene? " numeroasignaturas
```

```
#!/bin/bash

read -p "¿Qué curso es el que quieres crear? " directorio_nuevo

if [ -d "Apuntes/$directorio_nuevo" ]
then
    echo "Ya hay una carpeta creada para $directorio_nuevo. Continúa si quieres añadir asignaturas en él"

    read -p "¿Cuántas asignaturas tiene? " numeroasignaturas

    for ((i=1; i<=$numeroasignaturas; i++))
    do
        read -p "Nombre de la asignatura número $i: " nombre_subdirectorio
        mkdir "Apuntes/$directorio_nuevo/$nombre_subdirectorio"
    done
    echo "Creado nuevo directorio y respectivos subdirectorios"
else
    mkdir "Apuntes/$directorio_nuevo"

    read -p "¿Cuántas asignaturas tiene? " numeroasignaturas

    for ((i=1; i<=$numeroasignaturas; i++))
    do
        read -p "Nombre de la asignatura número $i: " nombre_subdirectorio
        mkdir "Apuntes/$directorio_nuevo/$nombre_subdirectorio"
    done
    echo "Creado nuevo directorio y respectivos subdirectorios"
fi

exit 0
```



crearnuevocurso.sh

Proceso de copia finalizado

```
ubuntu@ubuntu:~/Desktop/ProyectoSO$ ./crearnuevocurso.sh
```

```
¿Qué curso es el que quieres crear? cuarto
```

```
¿Cuántas asignaturas tiene? 1
```

```
Nombre de la asignatura número 1: asignatura1
```

```
Creado nuevo directorio y respectivos subdirectorios
```

```
ubuntu@ubuntu:~/Desktop/ProyectoSO$ 
```



# Copia de seguridad en Google Drive

- Bash no tiene una forma directa de guardar archivos en Google Drive.
- Dos opciones:
  - Programar utilizando la API de Google con cURL en el estándar OAuth 2.0 .
  - Utilizar programas o utilidades que facilitan y automatizan estas operaciones para scripts de bash.





# Rclone

- Rclone es un programa de línea de comandos que busca proporcionar comandos equivalentes al rsync, cp, mv, mount, ls, ncd, tree, rm, y cat de unix para almacenamiento en la nube.




# Instalación y configuración de Rclone

- En Linux es posible la instalación mediante la ejecución de : “sudo -v ; curl <https://rclone.org/install.sh> | sudo bash”.
  - Debido a que se encuentra incluido en los repositorios de muchas ramas de Linux, también existen otras comandos para poder instalarlo.
- Se encuentra en su página web una lista de servicios de almacenamiento en la nube para los cuales se puede configurar un “remote” que permite realizar operaciones de transferencia de datos con estos servicios.







# Script utilizado para subir la copia de seguridad a Google Drive

```
#!/bin/bash

# Ruta de la carpeta que deseas subir
folder_path_a="/home/gabriel/SO/Notas"
folder_path_b="/home/gabriel/SO/Apuntes"

# Nombre de la configuración remota en rclone
remote_name="drive"

# Subir la carpeta y sus archivos a Google Drive
rclone sync "$folder_path_a" "$remote_name:/Notas" --progress --create-empty-src-dirs
rclone sync "$folder_path_b" "$remote_name:/Apuntes" --progress --create-empty-src-dirs

echo "Carpeta y archivos subidos exitosamente a Google Drive."
```



# Copia de seguridad en Google Drive

Aunque nosotros hemos optado por esta opción hay distintas opciones para poder realizar la subida de archivos desde terminal hacia Google Drive:

- Con **gdrive**, que es un script de línea de comandos escrito en Go que te permite interactuar con Google Drive desde la terminal. Puede utilizarse para realizar diversas operaciones, como subir y descargar archivos, crear carpetas, y listar el contenido de tu cuenta de Google Drive.
- Con **google-drive-upload**, que es una herramienta de línea de comandos que se especializa en la carga de archivos y directorios a Google Drive. Para usarla hay que configurar las credenciales de la API de Google Drive a través de la Consola de Desarrolladores de Google y descargarlas. La herramienta simplifica el proceso de carga, proporcionando una solución específica para esta tarea sin funcionalidades adicionales.

# Adaptabilidad del proyecto

Las funcionalidades del proyecto pueden utilizarse en distintos y se pueden adaptar para distintos entornos:

- Se podría cambiar el uso de ficheros .txt por otros como ficheros de código como .java, .py...
- En nuestro caso nuestros tags se forman utilizando << >>, pero podríamos cambiar eso por !!, o por //, por ejemplo para poder guardar los comentarios de un archivo con código.
- Se podrían crear directorios nuevos.
- Podría utilizarse en otros ámbitos además del académico, como en el administrativo.



**FIN**

