

Lenguaje utilizado:

Python 3

Entorno de desarrollo:

Este programa fue desarrollado en Linux.

Instrucciones de ejecución:

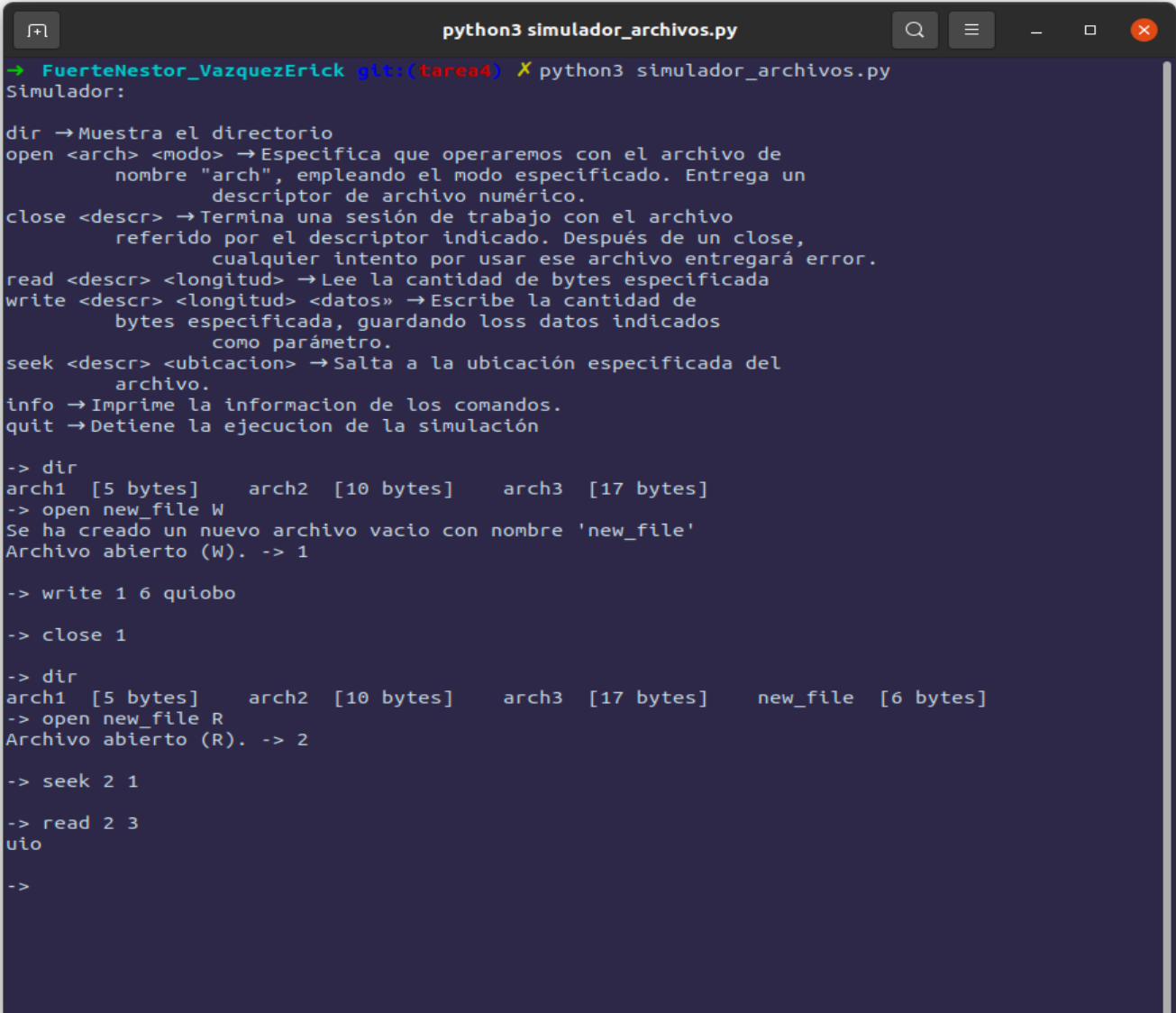
```
python3 <nombre_del_archivo.py>
```

Acerca del programa

Para esta tarea se solicitó implementar un sistema de pseudo archivos que consta de 5 operaciones básicas (open, close, read, write y seek).

Nosotros utilizamos un switch en donde íbamos comprobando que operación el usuario quería usar. Y mediante el uso de excepciones se iba comprobando que lo que ingresará el usuario cumpliera con las sintaxis de las operaciones.

Ejemplo de una ejecución exitosa:



```
python3 simulador_archivos.py
→ FuerteNestor_VazquezErick git:(tarea4) X python3 simulador_archivos.py
Simulador:

dir → Muestra el directorio
open <arch> <modo> → Especifica que operaremos con el archivo de
    nombre "arch", empleando el modo especificado. Entrega un
    descriptor de archivo numérico.
close <descr> → Termina una sesión de trabajo con el archivo
    referido por el descriptor indicado. Después de un close,
    cualquier intento por usar ese archivo entregará error.
read <descr> <longitud> → Lee la cantidad de bytes especificada
write <descr> <longitud> <datos> → Escribe la cantidad de
    bytes especificada, guardando loss datos indicados
    como parámetro.
seek <descr> <ubicacion> → Salta a la ubicación especificada del
    archivo.
info → Imprime la informacion de los comandos.
quit → Detiene la ejecucion de la simulación

-> dir
arch1 [5 bytes]   arch2 [10 bytes]   arch3 [17 bytes]
-> open new_file W
Se ha creado un nuevo archivo vacio con nombre 'new_file'
Archivo abierto (W). -> 1

-> write 1 6 quiobo

-> close 1

-> dir
arch1 [5 bytes]   arch2 [10 bytes]   arch3 [17 bytes]   new_file [6 bytes]
-> open new_file R
Archivo abierto (R). -> 2

-> seek 2 1

-> read 2 3
uio

->
```