**Simulación de una terminal de Linux que contenga los siguientes requerimientos:**

1. Administración de directorios.
2. Administración de archivos.
3. Memoria del sistema.

**Observaciones:**

1. El proyecto debe ser funcional en Python 2.7 en Linux.
2. Conceptualmente el árbol comienza vacío donde el primer nodo representa el directorio raíz inicial "/".
3. Un directorio puede contener ilimitados directorios e ilimitados archivos.
4. El directorio raíz inicial de todo el árbol no se puede eliminar.
5. Al entrar en su proyecto usted listará un pequeño texto mostrando:
6. El nombre, descripción y versión de su programa.
7. Sus autores y año de “publicación”.

**Comandos a simular:**

1. pwd: mostrar directorio actual.
2. cd: cambiar directorio. Parámetro: nueva ruta absoluta o relativa.
3. ls: listar los archivos y directorios de cualquier ruta, en formato horizontal. Parámetro: ruta relativa o absoluta. No se necesitan comodines.
4. ls -l: listar los archivos y directorios de cualquier ruta, en formato vertical. Parámetro: ruta relativa o absoluta. No se necesitan comodines.
5. touch: crea un archivo en blanco. Parámetro: ruta y nombre relativa o absoluta del archivo a crear. Solo se puede crear un directorio en una ruta que existe.
6. mkdir: crea un directorio. Parámetro: ruta y nombre relativa o absoluta del directorio a crear. Solo se puede crear un directorio en una ruta que existe.
7. mv: mueve un archivo o un directorio desde un Path origen hacia un Path destino. Parámetro ruta origen y ruta destino.
8. rm: elimina un archivo. Elimina un directorio vacío.
9. rm -r: elimina un directorio no vacío.
10. exit: debe finalizar el programa y salir del simulador.
11. find: debe buscar y listar las rutas que contiene un archivo con dicho nombre exacto.

**Requisitos del código:**

1. El proyecto debe resolverse mediante TDAs y haciendo uso del concepto de árboles.
2. El proyecto debe manejar el código fuente usando directorios (paquetes de python), haciendo uso de un único [main.py](http://main.py/) como cara del programa.
3. Todo el árbol debe ir creándose en un archivo de texto plano en un directorio del proyecto llamado "ROM". Cada vez que se inicie el programa se debe leer el árbol desde dicho directorio. El archivo de memoria debe ser en formato JSON.

**Requisitos de Entrega y Presentación:**

1. Fecha: Debe entregar el proyecto el día 3 de Marzo, a las 23:55 horas, 2019.
2. Integrantes: 4 integrantes por equipo.
3. Debe comprimir el proyecto en formato 7zip exclusivamente.
4. Nombre del archivo: 2019 PAC I - AED\*\*\*\* - Proyecto 1 - [APELLIDO] - [APELLIDO] - [APELLIDO] - [APELLIDO].7z
5. Cada [APELLIDO] corresponde con el primer apellido de cada miembro del equipo.
6. Nombre del Asunto del correo: 2019 PAC I - AED\*\*\*\* - Proyecto 1
7. En lugar de \*\*\*\* escribirá 0800 o 1200 dependiendo de su sección de clase.
8. En el cuerpo del proyecto deberá escribir el nombre completo y número de cuenta de los 4 integrantes del proyecto.
9. Cada equipo deberá grabar y enviar junto al proyecto un video mínimo de 2 minutos, máximo 3 minutos, explicando la funcionalidad completa de su proyecto y mostrando al final el archivo de "memoria".
10. Sobre el video: debe hablar una única persona en todo el video, o en caso de no querer hablar, puede hacer un video guiado con "textos" claramente explicados sobre lo que muestre en pantalla.