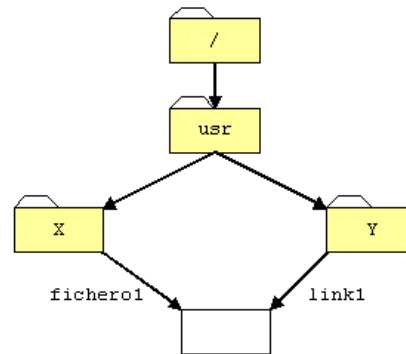


A continuación, se aportan una serie de actividades sobre sistemas de ficheros, estructuras de directorios y gestión de ficheros a realizar como práctica. Para la entrega de las mismas debe ser en un archivo .pdf a través del campus de EducamosCLM, realizando capturas de pantalla y documentando todos los pasos necesarios para llevarlas a cabo. En las capturas de pantalla debe de aparecer el *prompt* de la consola para asegurar la originalidad de las mismas.

Actividad 1. Creación de enlaces duros.

“Como ya se indicó con anterioridad los enlaces duros corresponden a asociaciones de nombres de ficheros con i-nodos. El objetivo de estos enlaces es realizar un i-nodo asignando nombres distintos y localizándose en directorios distintos o en el mismo.



A raíz de esto no es difícil discernir que tanto los archivos originales como los enlaces duros de dichos archivos harán uso del mismo i-nodo, de forma que apuntarán hacia el mismo contenido en el disco duro.

El enlace duro es importante en un sistema de ficheros como es ext4 extendido ampliamente en la mayoría de distribuciones basadas en el kernel de Linux. El enlace duro **permite identificar un contenido almacenado en el disco duro con un nombre distinto al del archivo original.**”

Como actividad práctica se pide:

1. Generar un enlace duro y un enlace simbólico explicando las diferencias.
2. Generar enlaces duros recursivos de todo un directorio.
3. Mostrar el resultado obtenido.

Actividad 2. Creación y rutas de un árbol de directorios.

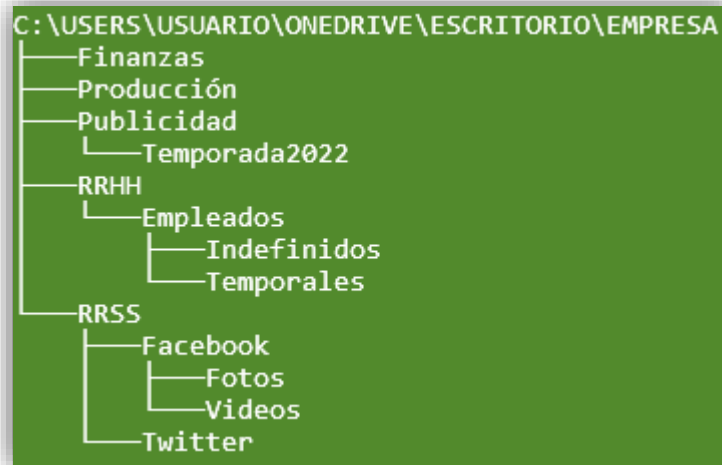
“A lo largo de la unidad hemos ido diferenciando entre el uso de rutas absolutas y rutas relativas tanto en sistemas operativos Linux como en sistemas operativos Microsoft Windows. Sabemos que el manejo ágil del comando `cd` o `CD` (cambiar de directorio) y el conocer en todo momento nuestra localización dentro de la estructura de directorios es fundamental para trabajar de forma eficiente”.

Dependiendo de la localización y de la necesidad en cada momento debemos hacer uso de un tipo de ruta u otra.

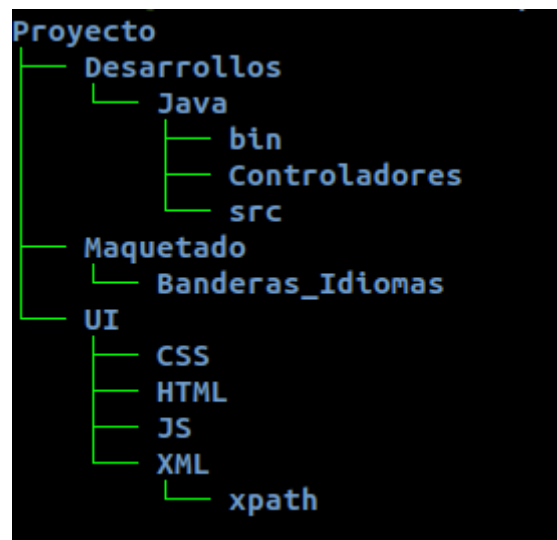
También hemos abordado la creación y visualización de árboles de directorios mediante relaciones jerárquicas, así como el redireccionamiento de esta información a fichero de texto con información.”

Como actividad práctica se pide:

1. Generar el siguiente árbol de directorios (*mediante interfaz textual*) en Microsoft Windows, y mostrar el resultado por pantalla.



2. Redireccionar la información del árbol de directorios en formato ASCII a un fichero de texto.
3. Generar el siguiente árbol de directorios (*mediante interfaz textual*) en Linux, y mostrar el resultado por pantalla.



4. Redireccionar la información del árbol de directorios a un fichero de texto.
5. Desde la consola de Linux, situarnos en el directorio **/src**, y desde ese directorio crear un fichero llamado panel.xml dentro del directorio **/XML** mediante el uso de una *ruta relativa* en un solo comando. Mostrar el resultado obtenido desde ese mismo directorio haciendo uso de una *ruta relativa*.
6. Realizar el procedimiento anterior, pero en este caso haciendo uso de una *ruta absoluta*.

Actividad 3. Uso de la expansión de comodines para copiar y eliminar ficheros en Linux.

“Sabemos que sabiendo la ruta origen y la ruta destino de un determinado fichero podemos copiarlo sin ningún problema de forma textual. También sabemos que a los comandos usados para realizar este tipo de acciones se les pueden añadir opciones como copiar con recursividad o forzar el copiado. También sabemos que podemos eliminar ficheros indicando la ruta origen.

Existe lo que se denomina **expansión de comodines** mediante la cual podemos copiar ficheros a través de patrones de identificación de ficheros haciendo uso del asterisco (*), que hace referencia a cualquier cadena de caracteres, y haciendo uso del (?), que solo hace referencia a un único carácter.”

Como actividad práctica se pide:

1. Investigar sobre el uso de esta opción.
2. A partir del árbol de directorios anterior crear los siguientes ficheros:
3. Una vez creado el árbol debemos copiar todos aquellos ficheros que contengan un 3 dentro del directorio /Controladores haciendo uso de la expansión de comodines.
4. Eliminar todos aquellos ficheros que contengan un 1 o un 2 en su nombre usando la expansión de comodines. Hacer uso del *verbose* para indicar los ficheros eliminados.
5. Por último, vamos a borrar todos los ficheros que empiecen por 3. Volver a hacer uso del *verbose* para indicar los ficheros eliminados.

```
usuario@usuario-VirtualBox:~/Escritorio/Proyecto$ tree
.
├── Desarrollos
│   ├── 01des.java
│   ├── 02des.java
│   ├── 03des.java
│   ├── 04des.java
│   ├── 05des.java
│   ├── 06des.java
│   ├── 11des.java
│   ├── 12des.java
│   ├── 13des.java
│   ├── 22des.java
│   ├── 23des.java
│   ├── 28des.java
│   ├── 33des.java
│   ├── 39des.java
│   └── 99des.java
├── Java
│   ├── bin
│   ├── Controladores
│   └── src
├── Maquetado
│   └── Banderas_Idiomas
├── UI
│   ├── CSS
│   ├── HTML
│   ├── JS
│   ├── XML
│   ├── panel.xml
│   └── xpath
```

Actividad 4. Introducción al uso de tuberías en Linux.

“En sistemas operativos con *kernel* de Linux sabemos que se puede hacer uso del operador tubería para concatenar comandos, de forma que la salida estándar del primer comando se comportará como entrada estándar del segundo comando. También se puede usar el operador tubería para dirigir la información estándar de la salida de un primer comando a un archivo mediante el comando *tee*.



Por otro lado, como bien se ha visto con anterioridad en el Sistema de Archivos Jerárquico (HFS 'Hierarchical File System') que utilizan la mayoría de distribuciones de Linux, el directorio `/etc` contiene los ficheros de configuración globales que afectan a todos los usuarios del sistema.

Como actividad práctica se pide:

1. Hacer uso del operador tubería para **contabilizar el número total** de ficheros de configuración (**wc**) que tenemos en dicho directorio, **almacenando (tee) un listado con los nombres** de dichos ficheros (**ls**) en un documento de texto externo que debe situarse en el escritorio de tu usuario personal, todo ello en **una sola línea de comando por terminal.**"

Actividad 5. Procesamiento de textos.

"En Linux existen dos comandos muy utilizados para extraer información sobre textos generados por otros comandos, ficheros o cadenas de caracteres en general.

Estos comandos son:

- A. *grep*: permite localizar un patrón en uno o varios ficheros, mostrando las líneas donde se encuentra. Es como una especie de *filtro*.
- B. *awk*: permite reconocer y buscar texto que siga un determinado patrón. Además, una vez identificado dicho patrón permite llevar a cabo distintas acciones sobre esa línea de texto en concreto.

Como actividad práctica se pide:

1. Buscar y comprender el uso, así como la sintaxis requerida para el mismo, de ambos operadores haciendo uso de la ayuda por terminal que te aporta el sistema operativo Linux (*hacer uso del [COMANDO] --help*) o de la información disponible en la documentación oficial del sistema operativo/foros especializados.
2. Crear un fichero de texto mediante el uso del editor de textos nativo *nano* que contenga el primer párrafo del Quijote de Cervantes:

```
usuario@usuario-VirtualBox:~/Escritorio$ cat parrafo
En un lugar de La Mancha, de cuyo nombre no quiero acordarme, no ha mucho tiempo que vivía un hidalgo de los de lanza en astillero, adarga antigua, rocín flaco y galgo corredor. Una de algo más vaca que carnero, salpicón las más noches, duelos y quebrantos los sábados, lentejas los viernes, algún palomino de añadidura los domingos consumían las tres partes de su hacienda.
```

3. Una vez entendido el uso de los comandos anteriores, y escrito el párrafo en un fichero de texto se pide encontrar la palabra galgo dentro de la línea del texto que corresponda haciendo uso del *grep*.
4. Busca información sobre el comando *awk*, y buscando un texto de tu elección, pon un ejemplo utilizando tuberías y los comandos que necesites, para buscar en dicho texto e imprimir en un archivo .txt aquellas líneas o frases que contengan la palabra "un", y que tengan una longitud mayor a 15 pero menor de 100 palabras.