

# Manual de usuario

## File System en c++

V. 1.0

Sofía Casandra Arancibia

**FICH**

**UNL**

**Facultad de Ingeniería  
y Ciencias Hídricas**  
UNIVERSIDAD NACIONAL  
DEL LITORAL

# Tabla de contenidos

Requerimientos.....3

Instalación de la aplicación.....3

Modo de uso.....4

Comandos.....4

Errores.....5

## Requerimientos

- Sistema operativo: Linux, UNIX

## Instalación de la aplicación

La aplicación no requiere de instalación, sólo se ejecuta. La ejecución se realiza corriendo directamente el código fuente del proyecto desde alguna IDE c/c++ (se recomienda Zinjai, disponible en <http://zinjai.sourceforge.net/>)

Dicho procedimiento se muestra en las figuras 1 y 2.

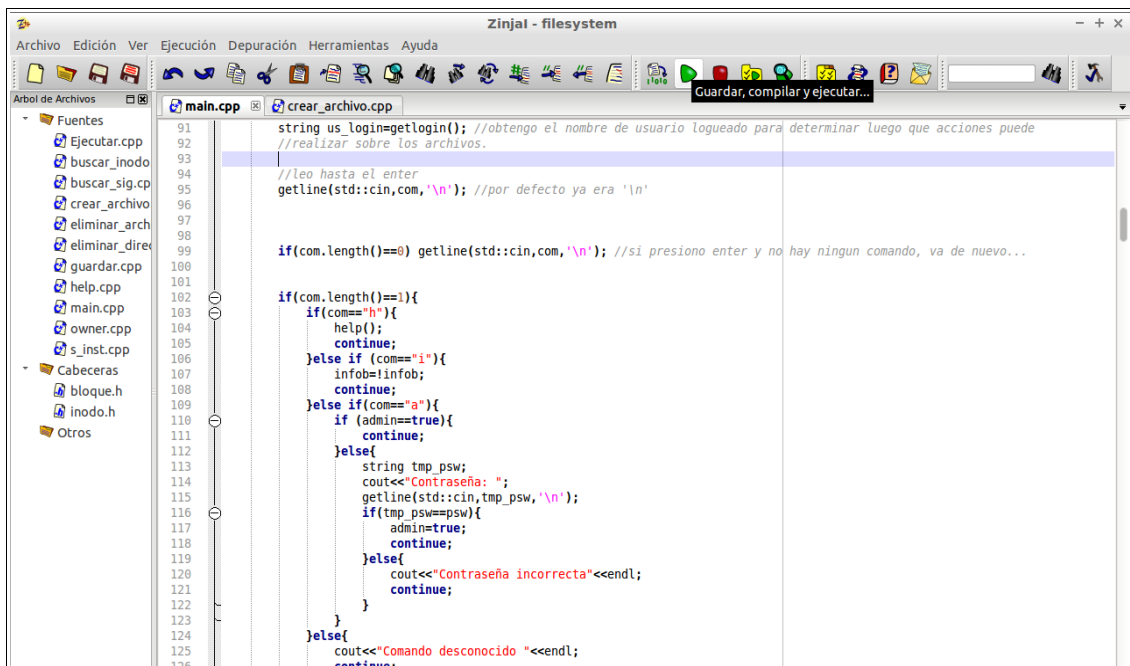


Fig. 1: Ejecutando código fuente desde IDE Zinjai

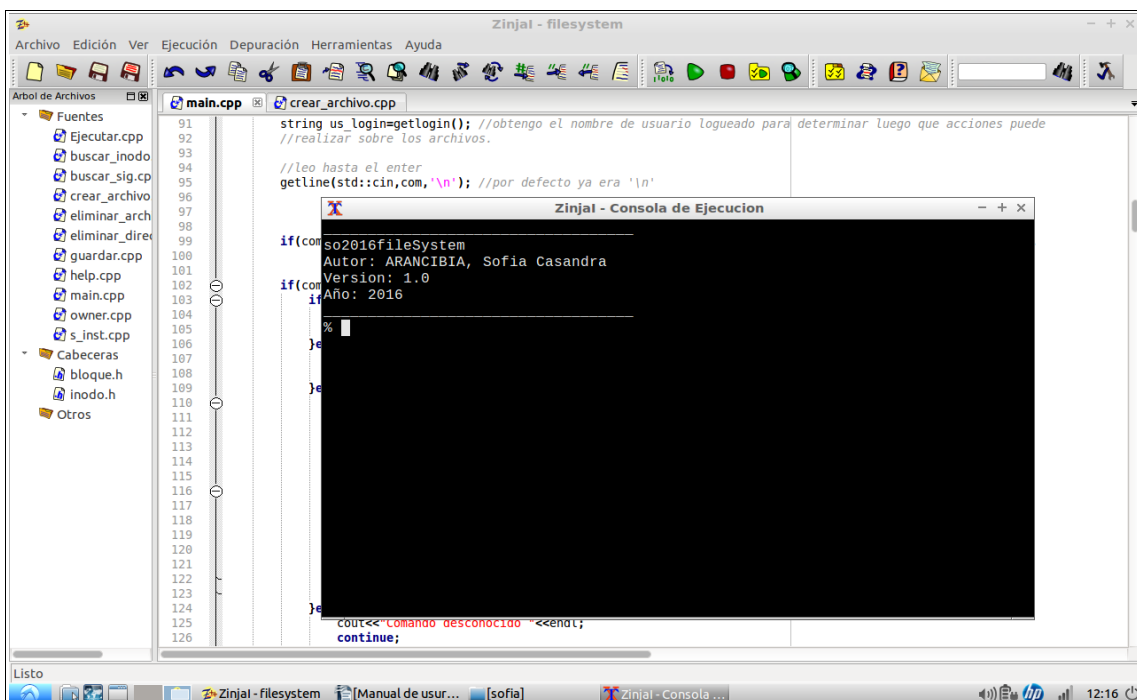


Fig. 2: Ejecutando código fuente desde IDE Zinjai

## Modo de uso

Al iniciar, la aplicación proveerá de un prompt (%) como el que se observa en la Fig.2, que indicará que la misma está a la espera de una instrucción. Dicha instrucción se ingresará por teclado e indicando como primera expresión el comando a ejecutar y seguido, los argumentos del mismo. Para salir de la aplicación, se deberá ingresar el comando **exit**.

Los comandos disponibles se listan a continuación.

## Comandos

**n** : crear nuevo archivo. Este comando requiere ingresar primero el nombre del archivo a crear y posteriormente el destino (ruta completa del directorio donde quiere ubicarse el archivo). Ejemplo de uso se observa en la Fig.3. La ruta ingresada debe corresponderse con un directorio existente en el sistema de archivos, de otro modo la aplicación informará error. Luego de ingresar el destino, se tipea el contenido del nuevo archivo. La aplicación leerá el teclado hasta detectar que se ha pulsado [enter], en ese momento se detiene la carga de texto y el archivo se guarda.

**v** : ver archivo. Seguido al comando se espera el nombre del archivo a leer, el cual se visualizará por pantalla.

**d** : eliminar archivo. Seguido al comando se espera la ruta completa del archivo a eliminar. Si el archivo no se encuentra, la aplicación informará error. Para efectuar la eliminación de un archivo, el usuario logueado deberá ser el propietario del archivo o ser "root" o administrador.

**m** : crear directorio. Seguido al comando se espera la ruta completa del directorio a crear. Por defecto, archivos y directorios se crean en el directorio raíz (existente, se crea al iniciar el file system). Si se pretende crear un directorio dentro de otro, este antecesor deberá haberse creado previamente. De otro modo, la aplicación informará error.

**r** : remover directorio. Seguido al comando se espera la ruta completa del directorio a eliminar. El comando se comporta recursivamente, esto significa que al invocarlo junto con una ruta de directorio, se eliminará también todo su contenido, con lo cual no será necesario que el directorio este vacío para realizar esta acción. Si el directorio a eliminar no existe, la aplicación informará error.

**a** : iniciar sesión como administrador. El usuario deberá ingresar una contraseña para loguearse como administrador. Un vez logueado, podrá efectuar todas las acciones mencionadas anteriormente sin necesariamente ser el propietario de los archivos.

**l** : listar contenido. Así como 'dir' en Windows o 'ls' en sistemas Linux, este comando lista el contenido de un directorio cuya ruta completa se ingresa a continuación del comando. El contenido se visualiza de diferentes colores según sean archivos(azul) o directorios(rojo). Si el directorio indicado no existe la aplicación informará error.

**h** : mostrar ayuda. Se visualiza un listado de los comandos disponibles.

**exit** : Salir de la aplicación.

## Errores

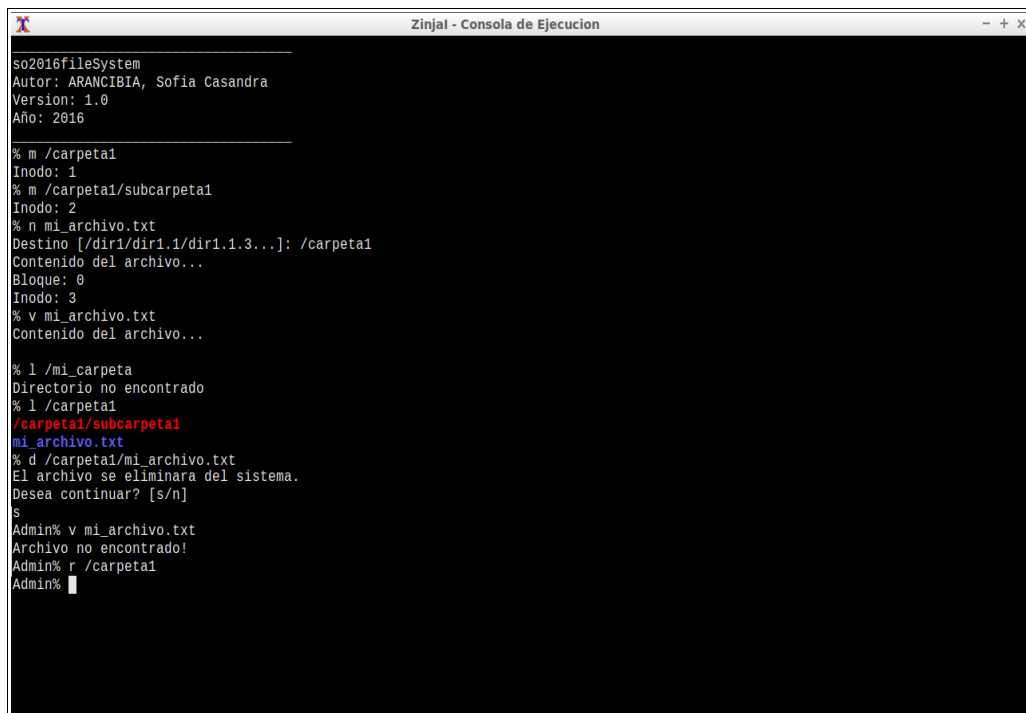
La aplicación notificará un error si el comando ingresado es incorrecto o inexistente.

Se notificarán además errores ajenos al usuario:

- Error en fork(): El proceso responsable de la ejecución del comando ha fallado. Entre los

posibles detonantes se encuentra la falta de memoria. Intente reiniciar la aplicación.

En la imagen siguiente se presenta una demostración de los comandos y posibles errores:



```
so2016fileSystem
Autor: ARANCIBIA, Sofia Casandra
Version: 1.0
Año: 2016

% m /carpeta1
Inodo: 1
% m /carpeta1/subcarpeta1
Inodo: 2
% n mi_archivo.txt
Destino [/dir1/dir1.1/dir1.1.3...]: /carpeta1
Contenido del archivo...
Bloque: 0
Inodo: 3
% v mi_archivo.txt
Contenido del archivo...

% l /mi carpeta
Directorio no encontrado
% l /carpeta1
/carpeta1/subcarpeta1
mi_archivo.txt
% d /carpeta1/mi_archivo.txt
El archivo se eliminara del sistema.
Desea continuar? [s/n]
s
Admin% v mi_archivo.txt
Archivo no encontrado!
Admin% r /carpeta1
Admin%
```

Fig. 3: Comandos y errores posibles