



Se desea desarrollar una aplicación que nos permita gestionar documentos de texto. Los usuarios de dicha aplicación podrán crear documentos nuevos que serán guardados en un fichero o mostrar el contenido de un documento desde un fichero.

Para mantener la integridad de los ficheros los usuarios deberán ser dados de alta en el sistema de forma que de cada usuario se deberá almacenar su nombre de usuario y su contraseña, y sus ficheros serán almacenados dentro de un directorio cuyo nombre coincide con el nombre de usuario. Lógicamente no puede existir dos usuarios con el mismo nombre de usuario.

Tendremos tres tipos de usuarios:

#### ADMINISTRADOR

Podrá crear usuarios, de cualquier tipo. Para crear un usuario se pedirá su usuario y su contraseña y se dará de alta en el sistema. Cuando un usuario es creado se creará también un directorio con su nombre. Será necesario comprobar previamente que dicho usuario no existe.

Podrá crear un nuevo fichero de texto: se le pedirá el nombre del fichero (sólo el nombre sin ruta). Se comprobará si el fichero existe y, si es así, se le preguntará al usuario si quiere sobrescribirlo. A continuación se le pedirá que escriba por teclado, línea a línea el contenido del fichero. Cuando el usuario escriba una línea con el texto “exit”, que podrá estar en mayúsculas, minúsculas o una combinación de ambas, terminará, obviando esa línea. El contenido se guardará en el fichero indicado.

El administrador no tiene límite en cuanto al número de ficheros o tamaño de los mismos.

También podrá mostrar un documento siempre que éste haya sido creado por él. Para ello pedirá el nombre del documento a mostrar y si está en su carpeta, muestra su contenido por pantalla.

#### USUARIO COMÚN

Puede realizar las mismas acciones que el administrador salvo crear usuarios. Además tendrá establecido un límite en cuanto al número de ficheros que estará definido en una constante de la clase y será de 10 y el tamaño máximo de todos los ficheros será de 10 MBytes, que estará definido en otra constante. Cuando el usuario cree un fichero, si ya ha llegado al número máximo de ficheros permitidos, se le indicará y no se podrá crear. Si aún no ha alcanzado la cuota y no ha llegado al número máximo de ficheros, se le permitirá crearlo aunque con este fichero se sobrepase la cuota, pero si es sobrepasada, ya no podrá crear más. Para conocer el tamaño en bytes de un fichero podemos usar el método `length()` de la clase `File`.

#### USUARIO INVITADO

Se caracteriza porque no tiene que ser dado de alta y no crearemos un objeto para cada uno y, por lo tanto, no podrá ser añadido en la lista de usuarios. Un usuario invitado podrá realizar las mismas acciones que un usuario común pero no tendrá directorio propio ni límite de archivos o cuota. Todos los ficheros del usuario invitado se almacenarán en la carpeta `DEFAULT`, por lo que no se podrá utilizar esa palabra como nombre de usuario.

Cuando se arranque el programa, si no existe ningún usuario, el sistema mostrará un mensaje indicando que se va a iniciar el programa por primera vez y pedirá un usuario y una contraseña para crear un primer usuario que será administrador.

Si al arrancar ya existe algún usuario, se dará la opción de entrar como invitado o se pedirá el usuario y la contraseña y, si coincide con alguno de los registrados en el sistema, mostrará un menú que dependerá del tipo de usuario.

Para un usuario administrador las opciones del menú son:

- Crear usuario
- Listar usuarios: mostrará un listado de todos los usuarios en el que aparecerá su nombre, el número de ficheros, la cuota de disco utilizada en bytes y el porcentaje de su cuota que tiene utilizado.
- Cambiar de usuario: se pedirá nuevamente el usuario y la contraseña, pasando a ser el usuario actual
- Crear documento
- Mostrar documento
- Listar documentos: muestra un listado con los documentos del usuario y el tamaño de cada uno
- Salir

Para un usuario común o usuario invitado tendremos las mismas opciones salvo crear usuario o listar usuarios.

Los datos de los usuarios serán almacenados antes de cerrar el programa en un fichero binario.

Para aprobar el examen será imprescindible que éste no tenga errores de compilación.