

## Trabajo Práctico N° 7: Archivos

*En cada ejercicio, el archivo debe abrirse y cerrarse dentro de cada función. La variable de tipo FILE\* debe ser local a la función. Se puede pasar por parámetro una variable de tipo char [30] con el nombre del archivo, ejemplos: "archivo.bin" o "alumnos.dat", o trabajar con una constante declarada.*

Utilizando datos primitivos (int, char, float, etc), elija un tipo de dato y realice las siguientes funciones:

1. Hacer una función que agregue un elemento al final de un archivo.
2. Hacer una función que muestre por pantalla el contenido de un archivo.
3. Hacer una función que retorne la cantidad de registros que contiene un archivo.

Utilizando la estructura:

```
typedef struct {  
    int legajo;  
    char nombreYapellido [30];  
    int edad;  
    int anio;  
    //año que cursa, recordar que no podemos utilizar la ñ para definir variables  
} stAlumno;
```

4. Crear una función que cargue un archivo de alumnos. Abrirlo de manera tal de verificar si el archivo ya está creado previamente. Cargar el archivo con 5 datos. Cerrarlo dentro de la función
5. Crear una función que muestre por pantalla los registros de un archivo de alumnos. Modularizar.
6. Crear una función que permita agregar de a un elemento al final del archivo. O sea, se debe abrir el archivo, se piden los datos (se llena una variable de tipo struct alumno), se escribe en el archivo y se cierra.
7. Crear una función que pase a una pila los números de legajo de los alumnos mayores de edad.
8. Dado un archivo de alumnos, hacer una función que cuente la cantidad de alumnos mayores a edad específica que se envía por parámetro.
9. Dado un archivo de alumnos, mostrar por pantalla el nombre de todos los alumnos entre un rango de edades específico (por ejemplo, entre 17 y 25 años). Dicho rango debe recibirse por parámetro. Modularizar

# UTN – FR Mar del Plata - Técnico Universitario en Programación

## Programación I

10. Dado un archivo de alumnos, mostrar los datos del alumno de mayor edad. Modularizar.
11. Crear una función que retorne la cantidad de alumnos que cursan un determinado año. El año buscado se pasa por parámetro.
12. Crear una función que reciba como parámetro un arreglo de tipo alumno y lo copie en el archivo. Asimismo, realice otra función que pase los elementos del archivo a un arreglo de alumnos, filtrando los estudiantes de un año en particular.
13. Crear una función que retorne la cantidad de registros que tiene el archivo. **Usar fseek y ftell.** Puede pensar la función para uso genérico, que sirva para averiguar la cantidad de registros de cualquier archivo.
14. Dado un archivo de alumnos, que tenga al menos 10 registros, hacer una función que muestre el contenido de un registro, cuyo número (entre 0 y 9) se pase por parámetro. Controlar no sobrepasar los límites del archivo.
15. Realice una (o varias) funciones que permitan modificar un registro existente en el archivo de alumnos. La misma debe permitir modificar uno o todos los campos de la estructura y sobrescribir el registro existente en el archivo.
16. Dado un archivo de alumnos, hacer una función que invierta los elementos del mismo. No se puede usar otro archivo auxiliar ni un arreglo auxiliar. Debe trabajar sobre el archivo. Puede utilizar variables de tipo alumno auxiliares.
17. Hacer una función principal que pruebe el funcionamiento de todos los incisos anteriores, con un menú de opciones para poder ejecutar todas las funciones requeridas. Tengan presente la correcta declaración y el ámbito de variables.